

# Teradata Vantage™ - Database Utilities

---

Release 17.10

July 2021

# Copyright and Trademarks

Copyright © 2000 - 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
	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.
	Indicates a hazardous situation which, if not avoided, could result in minor or moderate personal injury.
	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](mailto: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

<b>Chapter 1: Introduction to Utilities</b> .....	<b>8</b>
Changes and Additions .....	8
<b>Chapter 2: Database Utilities</b> .....	<b>10</b>
Alphabetical Listing of Utilities .....	10
For More Information .....	11
<b>Chapter 3: Abort Host (aborthost)</b> .....	<b>12</b>
Prerequisites .....	12
Runs From .....	12
Aborting Vantage Transactions .....	12
<b>Chapter 4: CheckTable (checktable)</b> .....	<b>13</b>
Runs From .....	13
Prerequisites .....	13
CheckTable Commands .....	13
<b>Chapter 5: CNS Run (cnsrun)</b> .....	<b>71</b>
Runs From .....	71
Syntax .....	71
Usage Notes .....	73
Example: Using cnsrun to run a utility .....	75
<b>Chapter 6: Control GDO Editor (ctl)</b> .....	<b>76</b>
Runs From .....	76
Syntax .....	76
Ctl Commands .....	77
<b>Chapter 7: Cufconfig Utility (cufconfig)</b> .....	<b>98</b>
Runs From .....	98
Syntax .....	98
Cufconfig Fields .....	99
Examples .....	147
<b>Chapter 8: Database Initialization Program (dip)</b> .....	<b>154</b>
Runs From .....	154
Running DIP Interactively .....	154
Running DIP Noninteractively .....	155
DIP Scripts .....	156

Usage Notes . . . . .	159
Example: DIP menu screen . . . . .	159
<b>Chapter 9: Database Window (xdbw) . . . . .</b>	<b>162</b>
Runs From . . . . .	162
Running DBW . . . . .	162
DBW Main Window . . . . .	162
Granting CNS Permissions . . . . .	163
Repeating Commands . . . . .	164
Saving Output . . . . .	164
DBW Scripts . . . . .	165
Accessing Help on DBW . . . . .	165
Commands . . . . .	166
<b>Chapter 10: DBS Control (dbscontrol) . . . . .</b>	<b>214</b>
Runs From . . . . .	214
DBS Control Field Groups . . . . .	214
DBS Control Commands . . . . .	215
DBS Control Fields . . . . .	227
Storage Group . . . . .	530
Checksum Fields . . . . .	532
<b>Chapter 11: Dump Unload/Load Utility (dul) . . . . .</b>	<b>540</b>
Runs From . . . . .	540
Starting and Running Dul . . . . .	541
Saving Linux Dumps to Removable Media . . . . .	542
Mailing Crash Dumps to the Teradata Support Center . . . . .	543
Restarting DUL . . . . .	543
Return Codes . . . . .	544
DUL Commands . . . . .	547
<b>Chapter 12: Ferret Utility (ferret) . . . . .</b>	<b>578</b>
Runs From . . . . .	578
Redirecting Input and Output . . . . .	578
The Teradata File System . . . . .	578
Write Ahead Logging (WAL) . . . . .	579
Ferret Command Syntax . . . . .	580
Using Ferret Parameters . . . . .	582
Ferret Commands . . . . .	589
<b>Chapter 13: Gateway Control (gtwcontrol) . . . . .</b>	<b>688</b>
Runs From . . . . .	688
Syntax . . . . .	688
Syntax Elements . . . . .	689
Usage Notes . . . . .	702

Gateway Host Groups .....	702
Gateway Log Files .....	703
Changing Maximum Sessions Per Gateway .....	704
<b>Chapter 14: Gateway Global (gtwglobal) .....</b>	<b>706</b>
Runs From .....	706
Syntax .....	707
Usage Notes .....	707
Gateway Global Commands .....	709
<b>Chapter 15: Lock Display (lokdisp) .....</b>	<b>744</b>
Runs From .....	744
Lock Modes .....	745
Locking and Row Partitions .....	746
Lock Display Utility Commands .....	746
<b>Chapter 16: Query Configuration (qryconfig) .....</b>	<b>775</b>
Runs From .....	775
About Query Configuration .....	775
Query Configuration Options .....	776
<b>Chapter 17: Query Session (qrysessn) .....</b>	<b>780</b>
Runs From .....	780
Query Session States .....	780
Parent and Child Sessions .....	781
Query Session Displays .....	781
Session State Display .....	783
State Information Displays .....	784
FastLoad Sessions State Displays .....	790
MultiLoad Sessions State Displays .....	794
FastExport Sessions State Displays .....	798
<b>Chapter 18: Recovery Manager (rcvmanager) .....</b>	<b>804</b>
Runs From .....	804
Prerequisites .....	804
Assigning Priority Levels .....	805
Online Transaction Recovery .....	805
Transaction Recovery Sequence .....	806
Multiple Recovery Sessions .....	806
Deferred Transaction Recovery .....	807
Down AMP Recovery .....	807
Startup/Restart Messages .....	812
Restarts .....	814
Canceling Rollback on Tables .....	814
Recovery Manager Commands .....	817

<b>Chapter 19: Show Locks (showlocks)</b>	<b>841</b>
Runs From	841
Host Utility Locks	841
Interpreting the Show Locks Display	842
Conflicts	843
<b>Chapter 20: Table Rebuild (rebuild)</b>	<b>844</b>
Runs From	844
Rebuilding Tables	844
Table Rebuild Utility Commands	845
Table Rebuild Usage Notes	859
<b>Chapter 21: Teradata Locale Definition Utility (tdlocaledef)</b>	<b>862</b>
Runs From	862
Syntax	862
SDF File	864
Example SDF Files	873
<b>Chapter 22: TPA Reset (tpareset)</b>	<b>878</b>
Runs From	878
Syntax	878
Usage Notes	881
Example: Resetting (restarting) the Database	881
<b>Chapter 23: Update DBC (updatedbc)</b>	<b>882</b>
Runs From	883
Usage Notes	883
<b>Chapter 24: Update Space (updatespace)</b>	<b>884</b>
Runs From	884
Syntax	884
Usage Notes	885
Example: Updating space for all databases	886
Example: Updating temporary space for PUBLIC database	887
<b>Chapter 25: Vproc Manager (vprocmanager)</b>	<b>888</b>
Runs From	888
Definitions of Terms Used in Vproc Manager	889
Vproc Manager Command Syntax	895
Vproc Manager Commands	896
<b>Appendix A: How to Read Syntax</b>	<b>933</b>
<b>Appendix B: Starting the Utilities</b>	<b>935</b>

**Appendix C: Session States, Events, and Actions . . . . . 939**

**Appendix D: PDE Tools . . . . . 960**

**Appendix E: Additional Information . . . . . 963**

# Introduction to Utilities

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 Vantage™ - Database Utilities*, describes the utility programs that support Vantage. These utilities are used to configure, maintain, and troubleshoot Vantage. They display control fields, find and correct problems within the file system, rebuild tables that may have been corrupted, and manage the virtual processors (vprocs). These utilities are also used to abort transactions and processes; monitor system performance, resources, and status, perform internal system checking, and perform system configuration, initialization, recovery, and tuning.

## Note:

The utilities described are not available for as-a-service customers.

Main titles 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 to start the utility from the command line or Database Window.

Note that these utilities are distinct from the “load utilities”, which are used to load data into Vantage. For detailed information about load utilities, see the following documents:

- *Teradata® FastExport Reference*, B035-2410
- *Teradata® FastLoad Reference*, B035-2411
- *Teradata® MultiLoad Reference*, B035-2409
- *Teradata® Parallel Data Pump Reference*, B035-3021

## Changes and Additions

Date	Description
July 2021	<ul style="list-style-type: none"> <li>• New DBS Control General fields <a href="#">FastAlterDefault</a> and <a href="#">FastAlterEnable</a>.</li> <li>• New DBS Control Native Object Store fields <a href="#">ForeignTableErrorsData</a> and <a href="#">ForeignTableErrorsLimit</a>.</li> <li>• New MAXSESSIONS, SCOPE { DICTIONARY   USER   ALL } options for <a href="#">CheckTable (checktable)</a></li> <li>• New LOCAL option for <a href="#">SHOWCYLALLOC</a> and <a href="#">SHOWWHERE</a>.</li> <li>• Updated syntax description of the IN PARALLEL option for <a href="#">REBUILD AMP</a> and <a href="#">REBUILD AMP FALLBACK TABLES</a> to show the increased number of parallel streams and to say that multiple tables can be rebuilt in parallel.</li> <li>• These utilities can be run from the Linux command line:               <ul style="list-style-type: none"> <li>◦ <a href="#">Abort Host (aborthost)</a></li> </ul> </li> </ul>



Date	Description
	<ul style="list-style-type: none"> <li>◦ <a href="#">Query Configuration (qryconfig)</a></li> <li>◦ <a href="#">Query Session (qrysessn)</a></li> <li>◦ <a href="#">Recovery Manager (rcvmanager)</a></li> <li>◦ <a href="#">Show Locks (showlocks)</a></li> <li>◦ <a href="#">Update DBC (updatedbc)</a></li> <li>◦ <a href="#">Update Space (updatespace)</a></li> <li>• TLSv1.2 is now supported between clients and the database server.               <ul style="list-style-type: none"> <li>◦ New options are added to <code>gtwglobal</code> to update the TLS configuration from the gateway TLS configuration. See <a href="#">Gateway Global (gtwglobal)</a>.</li> <li>◦ New option (<code>--TLS</code>) is added to <code>gtwcontrol</code> to enable or disable TLSv1.2 and trace the protocol. See <a href="#">Gateway Control (gtwcontrol)</a>.</li> </ul> </li> <li>• Fallback subtables are no longer compressed by default.</li> </ul>
June 2020	<ul style="list-style-type: none"> <li>• DBS Control:               <ul style="list-style-type: none"> <li>◦ New Performance field, <a href="#">DBQL_AWTDPS_CacheMaximum</a>.</li> <li>◦ New Optimizer Stats field, <a href="#">BLCStatsForCDS</a></li> </ul> </li> <li>• Ferret:               <ul style="list-style-type: none"> <li>◦ New PERVPROC option for Ferret <a href="#">SHOWBLOCKS</a> command.</li> </ul> </li> </ul>

## Database Utilities

### Alphabetical Listing of Utilities

Utility	Purpose
Abort Host (aborthost)	Aborts all outstanding transactions running on a failed host, until the system restarts the host.
CheckTable (checktable)	Checks for inconsistencies between internal data structures such as table headers, row identifiers, and secondary indexes.
CNS Run (cnsrun)	Allows running of database utilities from scripts.
Control GDO Editor (ctl)	Displays the fields of the PDE Control Parameters GDO, and allows modification of the settings.
Cufconfig Utility (cufconfig)	Displays configuration settings for the user-defined function and external stored procedure subsystem, and allows these settings to be modified.
Database Initialization Program (DIP)	Runs one or more of the standard DIP scripts packaged with Advanced SQL Engine. These scripts create a variety of database objects that can extend the functionality of Advanced SQL Engine with additional, optional features.
DBS Control (dbscontrol)	Displays the DBS Control fields, and allows these settings to be modified.
Dump Unload/Load (DUL)	Saves and restores system dump tables to and from files.
Ferret Utility (ferret)	Defines the scope of an action, such as a range of tables or selected vprocs, displays the parameters and scope of the action, and performs the action, either moving data to reconfigure data blocks and cylinders, or displaying disk space and cylinder free space percent in use of the defined scope.
Gateway Control (gtwcontrol)	Modifies default values in the fields of the Gateway Control Globally Distributed Object (GDO).
Gateway Global (gtwglobal)	Monitors and controls the Advanced SQL Engine workstation-connected users and their sessions.
Lock Display (lokdisp)	Displays a snapshot capture of all real-time database locks and their associated currently-running sessions.
Query Configuration (qryconfig)	Reports the current Advanced SQL Engine configuration, including the Node, AMP, and PE identification and status.
Query Session (qrysessn)	Monitors the state of selected Advanced SQL Engine sessions on selected logical host IDs.
Recovery Manager (rcvmanager)	Displays information used to monitor progress of a Advanced SQL Engine recovery.
Show Locks (showlocks)	Displays locks placed by Table Rebuild operations on databases and tables.

Utility	Purpose
	For details on Table Rebuild, see <i>Teradata Vantage™ - Database Utilities</i> , B035-1102 .
Table Rebuild (rebuild)	Rebuilds tables that Advanced SQL Engine cannot automatically recover, including the primary or fallback portions of tables, entire tables, all tables in a database, or all tables in an Access Module Processor (AMP). Table Rebuild can be run interactively or as a background task.
Teradata Locale Definition Utility(tdlocaledef)	Converts a Specification for Data Formatting file (SDF) into an internal, binary format (a GDO) for use by Advanced SQL Engine. The SDF file is a text file that defines how Advanced SQL Engine formats numeric, date, time, and currency output.
Tpareset (tpareset)	Resets the PDE and database components of Advanced SQL Engine.
Update DBC (updatedbc)	Recalculates the PermSpace and SpoolSpace values in the DBASE table for the user DBC, and the MaxPermSpace and MaxSpoolSpace values of the DATABASESPACE table for all databases based on the values in the DBASE table.
Update Space (updatespace)	Recalculates the permanent, temporary, or spool space used by a single database or by all databases in a system.
Vproc Manager (vprocmanager)	Manages the virtual processors (vprocs). For example, obtains status of specified vprocs, initializes vprocs, forces a vproc to restart, and forces a Advanced SQL Engine restart.

## For More Information

For more information on...	See...
starting the utilities	<a href="#">Starting the Utilities.</a>
utilities related to the PDE (operating system) level of Vantage	<a href="#">PDE Tools</a>
utilities related to Vantage security	<i>Teradata Vantage™ - Advanced SQL Engine Security Administration</i> , B035-1100
FastExport, FastLoad, MultiLoad, and TPump	the following client utility books: <ul style="list-style-type: none"> <li>• <i>Teradata® FastExport Reference</i>, B035-2410</li> <li>• <i>Teradata® FastLoad Reference</i>, B035-2411</li> <li>• <i>Teradata® MultiLoad Reference</i>, B035-2409</li> <li>• <i>Teradata® Parallel Data Pump Reference</i>, B035-3021</li> </ul>

# Abort Host (aborthost)

---

**Note:**

The utility described is not available for as-a-service customers.

---

The Abort Host utility, `aborthost`, allows you to cancel all outstanding transactions running on a mainframe-attached host that is no longer operating.

## Prerequisites

You should be familiar with Teradata client (host software), particularly the Teradata Director Program (TDP).

## Runs From

- Database Window or comparable interface to the Teradata console subsystem, such as `cnstern`
- Teradata Viewpoint Remote Console portlet
- Linux command line

For general information on starting the utilities from different interfaces, see [Starting the Utilities](#). For information on Viewpoint, see *Teradata® Viewpoint User Guide*, B035-2206.

## Aborting Vantage Transactions

To abort all Vantage transactions for a particular host, after starting `aborthost`, type the following then press Enter:

```
ABORT HOST  nnn
```

where *nnn* is the identifier of the host that originated the database transactions you want to abort.

For logon and start requests to a host that has been aborted in this manner, the system displays this error message:

```
Host quiesced by operator
```

After running Abort Host, the only way to re-establish host communication with the database is to restart the appropriate TDP.

# CheckTable (checktable)

---

## Note:

The utility described is not available for as-a-service customers.

---

The CheckTable utility is a diagnostic tool that finds inconsistencies and data corruption in data structures, such as table headers, row identifiers, and indexes (including geospatial).

CheckTable offers several levels of data checking. Higher-level checks are generally more thorough than lower-level checks, but require more time and system resources. Certain general table checks are performed at all levels. After CheckTable runs, it reports the total number of tables checked and bypassed.

Teradata recommends running CheckTable on a regular schedule to detect problems that may occur. Although CheckTable identifies data inconsistencies and corruption, it cannot repair these problems. For help repairing problems identified by CheckTable, contact the Teradata Support Center.

## Runs From

- Database Window or comparable interface to the Teradata console subsystem, such as cnstern
- Teradata Viewpoint Remote Console portlet

For general information on starting the utilities from different interfaces, see [Starting the Utilities](#). For information on Viewpoint, see *Teradata® Viewpoint User Guide*, B035-2206.

## Prerequisites

In order to run CheckTable:

- Vantage must be online
- No more than one AMP per cluster can be down
- Logons can be enabled or disabled

## CheckTable Commands

CheckTable supports the following commands.

Command	Purpose
<a href="#">CHECK</a>	Checks all tables or specified databases and tables. This is the main CheckTable command.
<a href="#">OUTPUT</a>	Displays or changes the destination for CheckTable output.
<a href="#">ERRORFILEDIR</a>	Displays or changes the location of the error file for the current CheckTable session.

Command	Purpose
<a href="#">HELP</a>	Displays CheckTable online help.
<a href="#">QUIT</a>	Terminates the CheckTable utility.
<a href="#">STATUS</a>	Displays the status of the currently running CHECK command.
<a href="#">ABORT</a>	Aborts the currently running CHECK command.
<a href="#">ABORT TABLE</a>	Aborts the check on the table currently being checked.

## CHECK

After the CheckTable utility is started, use the CHECK command to:

- specify tables and databases to be included in or excluded from the check
- designate the level of checking
- specify other CHECK options, discussed below

## Syntax

```
CHECK check_spec
  [ BUT { ONLY | NOT } check_object [,...] ] AT LEVEL
  { PENDINGOP | ONE | TWO | THREE }
  [ WITH ERROR LIMIT = [ nnn ] | WITH NO ERROR LIMIT ]
  [ SKIPLOCKS ]

  [ IN SERIAL | IN PARALLEL [ TABLES = n ] [ MAXSESSIONS ] ]
  [ PRIORITY = { L | M | H | R | performance_group_name } ]
  [ CONCURRENT MODE [ RETRY LIMIT n ] ]
  [ ERROR DOWN | DOWN ONLY ]
  [ COMPRESSCHECK ]
  [ CHECKINVALIDS ]
  [ SCOPE { DICTIONARY | USER | ALL } ] ;
```

### *check\_spec*

```
{ ALL TABLES [ EXCLUDE [dbname.]table_name [,...] ] |
  dbname EXCLUDE [dbname.]table_name [,...] |
  [dbname.]table_name [,...] |
  AGAIN [ Error-filename ]
}
```

### *check\_object*

```
{ INDEX ID = nnn |
  UNIQUE INDEXES |
  NONUNIQUE INDEXES |
  REFERENCE ID = nnn |
  REFERENCE INDEXES |
  DATA |
  LARGE OBJECT ID = nnn |
```

```

LOB ID = nnn |
LARGE OBJECTS |
LOBS |
SJI ID = nnn |
SJIS
}

```

## Syntax Elements

### ALL TABLES

CheckTable checks all tables, join indexes, and hash indexes in all databases in the system.

---

#### Note:

CheckTable considers join and hash indexes as tables. References to tables in this discussion should be interpreted as encompassing tables, join indexes, and hash indexes.

---

### *dbname*

CheckTable checks all tables, join indexes, and hash indexes in the specified database.

You can use wildcard characters or wildcard syntax in specifying database names. For more information, see [Using Wildcard Characters in Names](#).

### AGAIN

Re-checks tables that failed the previous check.

### Error-filename

Error-filename is a Linux filename.

### EXCLUDE

CheckTable excludes the specified tables or databases from the check.

- CHECK ALL TABLES EXCLUDE excludes one or more databases or tables.
- CHECK *dbname* EXCLUDE excludes one or more tables in a particular database.

If a specified object does not exist in the system, the summary report lists the object in a message.

If you do not specify this option, CheckTable checks all data objects in the system.



**tablename or dbname.tablename**

CheckTable checks a specific table, join index, or hash index, including global temporary tables.

You can use wildcard characters or wildcard syntax in specifying database names. For more information, see [Using Wildcard Characters in Names](#).

**BUT ONLY  
BUT NOT**

Places constraints on what CheckTable checks:

- BUT ONLY causes CheckTable to check only the subsequently specified objects.
- BUT NOT causes CheckTable to skip checking of the subsequently specified objects.

If you do not specify any selection constraints, CheckTable checks all data objects in the system.

These options are ignored for level-pendingop checks.

Example: The following command checks all tables limiting the check to only INDEX ID=*nnn* at level three:

```
CHECK ALL TABLES BUT ONLY INDEX ID=nnn AT LEVEL THREE;
```

**INDEX ID = *nnn***

Specifies a specific secondary index (specified by its index ID) when using a constraint with CheckTable. In general, specify this option only when you want to check a single table.

**UNIQUE INDEXES**

Specifies all unique secondary indexes when using a constraint with CheckTable.

**NONUNIQUE INDEXES**

Specifies all Nonunique Secondary indexes when using a constraint with CheckTable.

**REFERENCE ID = *nnn***

Specifies a specific reference index (as specified by its index ID) when using a constraint with CheckTable.

**REFERENCE INDEXES**

Specifies all reference indexes when using a constraint with CheckTable.

**DATA**

Specifies the data subtable when using a constraint with CheckTable.

**LARGE OBJECT ID = *nnn***

Specifies a specific large object (as specified by its ID) when using a constraint with CheckTable.

**LOB ID = *nnn***

Specifies a specific large object (as specified by its ID) when using a constraint with CheckTable.

**LARGE OBJECTS**

Specifies all large objects when using a constraint with CheckTable.

**LOBS**

Specifies all large objects when using a constraint with CheckTable.

**SJI ID = *nnn***

Specifies a specific system defined join index (as specified by its index ID) when using a constraint with CheckTable.

**SJIS**

Specifies all system defined join indexes when using a constraint with CheckTable.

**AT LEVEL**

Can be one of **PENDINGOP**, **ONE**, **TWO**, or **THREE**:

- **PENDINGOP** provides a list of tables for which pending operations exist.
- **ONE** isolates specific tables with errors.
- **TWO** provides a detailed check of:
  - Consistency of row IDs
  - Checksum of primary and fallback rows
  - Hash codes
- **THREE** provides the most diagnostic information, but uses more system resources and requires more time. Use this level of check only when necessary.

For more information, see [Check Levels](#).

**WITH ERROR LIMIT= *nnn***

CheckTable stops checking a table if it finds *nnn* or more errors. If CheckTable was checking more than one table, it continues on to the next table.

The default is 20 errors for each table checked.

**WITH NO ERROR LIMIT**

CheckTable reports all errors for each table.

**SKIPLOCKS**

CheckTable skips all locked tables automatically.

If a table is locked, and if CheckTable cannot obtain a lock, then CheckTable indicates the table check is skipped. The summary at the end of CheckTable processing includes the total number of tables skipped.

Without this option in non-concurrent mode, CheckTable may wait indefinitely on a table lock, depending on the value of the ChecktableTableLockRetryLimit field in DBS Control. The ChecktableTableLockRetryLimit field specifies the duration, in minutes, that CheckTable, in non-concurrent mode, will retry a table check when the table is locked by another application.

The default is 0, which indicates that in non-concurrent mode, CheckTable will retry a table check until CheckTable can access the table.

If the ChecktableTableLockRetryLimit setting is greater than 0, then CheckTable will retry a table check within the specified limit.

For more information, see [ChecktableTableLockRetryLimit](#).

**IN SERIAL****IN PARALLEL**

Specifies the mode of checking CheckTable uses:

- IN SERIAL means CheckTable checks a single table at a time. This is the default.
- IN PARALLEL means CheckTable checks multiple tables simultaneously. Using PARALLEL mode saves time but is resource intensive. The number of tables that CheckTable can check simultaneously in parallel depends on resource availability. You can check the status of the number of parallel checks CheckTable performs at any time. For information on how to check the status, see [Determining the Status of a Table Check](#).

**MAXSESSIONS**

Runs user table check with the maximum number of parallel sessions.

**TABLES=*n***

Optionally used with IN PARALLEL to specify the upper limit on the number of tables that will be checked simultaneously. The value of *n* can be any integer from two through eight.

At level-one checking, the actual maximum number of tables that can be checked simultaneously is based on the maximum number of AMP work tasks (AWT) defined for the system. At all other levels, the maximum is based on the maximum number of AWT

and on the available spool space. The TABLES=*n* option is used to decrease the number of tables checked in parallel to something less than the maximum.

If there is insufficient spool space to check *n* tables in parallel, the number checked will be less than the number specified.

#### **PRIORITY =**

Specifies the priority at which CheckTable should run. This option can be used to control resource usage and improve performance. The available priority levels are:

- L - Low
- M - Medium

This is the default, if the PRIORITY= option is not specified.

- H - High
- R - Rush

These values are mapped to specific workloads. Default mappings exist for L, M, H, and R, but can be changed in the Workload Designer Viewpoint portlet. For more information on the Workload Designer portlet, see *Teradata® Viewpoint User Guide*, B035-2206.

- *workload name* - a specific workload in which CheckTable should run. If the specified workload does not exist, CheckTable runs in the workload mapped to CheckTable in Workload Designer.

#### **CONCURRENT MODE**

Use when running on a non-quiet system. CONCURRENT MODE reduces lock contention by optimizing the locking protocol and by automatically skipping the locked tables to retry them later. After CheckTable checks all tables, CheckTable automatically retries all tables that were skipped due to lock contention.

In a non-quiet system with logons enabled or disabled, or on a quiet system with logons enabled, CheckTable overrides a user-specified command and enforces CONCURRENT MODE with RETRY LIMIT=1. In a normal running environment, CONCURRENT MODE is used by default.

When you specify the CONCURRENT MODE option, you cannot specify the IN PARALLEL option. CheckTable always checks tables serially in concurrent mode to reduce lock contention.

CHECK ALL TABLES CONCURRENT MODE does not check data dictionaries and the DBC database in concurrent mode. To check the DBC database in concurrent mode, use CHECK DBC CONCURRENT MODE.

For more information on CONCURRENT MODE, see [Determining the Status of a Table Check](#).

**RETRY LIMIT *n***

The duration in minutes that CheckTable waits before attempting to re-check a table that was skipped during CONCURRENT MODE operation.

If *n* is zero, CheckTable will not attempt to check tables that were skipped.

If RETRY LIMIT is not specified, CheckTable retries the locked tables indefinitely, or until all tables have been checked successfully.

When trying to access a locked table, CheckTable waits a maximum of five minutes for the lock to be released. If the lock is not released in that time, CheckTable moves on to the next table. If the RETRY LIMIT has not been met, CheckTable will return to the locked table to try again.

**ERROR ONLY**

CheckTable displays only bypassed tables and tables that have errors or warnings. This option allows you to quickly identify and address problems that CheckTable finds.

This option is ignored during level-pendingop checks.

**DOWN ONLY**

Vantage can isolate some file system errors to a specific data or index subtable, or to a contiguous range of rows ("region") in a data or index subtable. In these cases, Vantage marks only the affected subtable or region down. This improves system performance and availability by allowing transactions that do not require access to the down subtable or rows to proceed, without causing a database crash that would require a system restart.

However, if several regions in a subtable are marked down, it could indicate a fundamental problem with the subtable itself. Therefore, when a threshold number of down regions is exceeded per AMP, the entire subtable is marked down on all AMPs, making it unavailable to most SQL queries. This threshold can be adjusted by means of the MaxDownRegions field in the General fields group of the DBS Control utility.

The DOWN ONLY option causes CheckTable to display status information for only those data and index subtables that have been marked down. Down status information for subtables is displayed at check levels one, two, and three. The specific rows included in down regions are listed only for check levels two and three. Down status information is not displayed for level-pendingop checks.

**COMPRESSCHECK**

CheckTable compares the compression information (compress multi-value and name of compression algorithm UDF if specified in table definition) from the table header to the corresponding information stored in the CompressValueList column of the DBC.TVFields table.

At Level Three, compressed data subtables are checked. For example, for multi-value compressed columns, CheckTable compares the compress list in the table header to the column values in each row to ensure the appropriate values were compressed.

This option is ignored during level-pendingop checks.

For more information on compression, see *Teradata Vantage™ - Database Design*, B035-1094 and *Teradata Vantage™ - Data Types and Literals*, B035-1143.

## CHECKINVALIDSI

Causes CheckTable to continue index checks, even if it encounters an invalid secondary index. If this option is not specified, CheckTable bypasses further index checks and logs a warning message when it encounters an invalid SI. By allowing CheckTable to continue index checking, this option may result in a more detailed error reported that can help determine why the index is invalid.

## SCOPE

Use with one of the following:

- **DICTIONARY:** Runs only a dictionary check with batch mode error logging for the given CheckTable command.
- **USER:** Runs only a user table check for the given CheckTable command.
- **ALL:** Runs a dictionary check in batch mode with error logging and user table check for the given CheckTable command.

## Usage Notes

### CHECK Command Restrictions

The CHECK command, including database or table names, cannot exceed 1300 characters. Wildcard characters are counted as single characters. For more information, see [Valid Characters in Object Names](#) and [Wildcard Characters in Names](#).

The CHECK command is not case sensitive. You can specify the syntax elements in uppercase or lowercase or a mixture of both. This also applies to names specified in wildcard syntax.

CheckTable can check an unlimited number of databases and tables, however the CHECK command accepts a maximum of 30 names on the command line. Names of the form *dbname*, *tablename*, and *dbname.tablename* are each considered to be a single name. However, wildcard expressions are also considered to be single names, and can resolve to any number of databases and tables. For more information on using wildcard expressions, see [Wildcard Characters in Names](#).

## CHECK AGAIN Command Restrictions

Each CHECK command generates an error file containing a list of the significant errors. Use the CHECK AGAIN command to check only the (failed) tables listed in an error file. You can also specify an input error file or let CheckTable default the input error file to the output error file generated by the previous CHECK command.

If *Error-filename* is not specified, the last error file generated by the previous CHECK command in the current CheckTable instance is used as input.

*Error-filename* may be a fully qualified filename (such as /var /tmp/CheckTableErrors20130503164640) or a relative path filename (for example, CheckTableErrors20130503164640).

For a relative path name, the root directory is the error file directory, which is managed with the ERRORFILEDIR command.

## Determining the Status of a Table Check

To determine the status of the current table check type status at the CheckTable command prompt. For more information, see [STATUS](#).

For example, assume that the table check in progress specifies database DB0 with tables t1, t10, t100, t1000, and t11. The following table shows the results for Parallel and Serial modes.

Mode	Result Example
PARALLEL	<pre> &gt;&gt;&gt; STATUS: CheckTable running in PARALLEL mode.            5 CheckTable tasks started.            4 CheckTable tasks ACTIVE.            1 CheckTable tasks IDLE.           1000005 bytes spool space in use.            120 bytes spool space available.  Task      Status ****      ***** 1         Checking data subtable ("DB0"."T1"). 2         Checking data subtable ("DB0"."T10"). 3         Checking data subtable ("DB0"."T100"). 4         Waiting for a read lock on table           ("DB0"."T1000"). 0 lock retry(s), total           wait time of 1.4 minute(s).  These results indicate that CheckTable started five parallel checks: • Three CheckTable tasks are checking the data subtable of tables DB0.T1, DB0.T10, and DB0.T100. • One CheckTable task is waiting for a read lock on table DB0.T1000. • One CheckTable task is paused because of insufficient resources. </pre>

Mode	Result Example
SERIAL	<pre>&gt;&gt;&gt;&gt; STATUS: Checking data subtable ("DB0"."T10").</pre> <p>This indicates that CheckTable is checking the data subtable of the table DB0.T10.</p>

## Stopping Table Checks

To stop the check of the current single table (in SERIAL mode) or current group of tables being checked (in PARALLEL mode), type `abort table` at the CheckTable command prompt. For more information, see [ABORT TABLE](#).

To stop the check of all tables, type `abort` at the CheckTable command prompt. For more information, see [ABORT](#).

## Logging CheckTable Runs

CheckTable logs the start and finish times for CHECK runs to the DBC. SW\_Event\_Log table and to the operating system event log. The logged start and finish times allows calculation of duration of CheckTable run, and prediction of duration of future CheckTable runs.

Log entries can be viewed by querying Vantage, or viewing the operating system event logs. The event log is the messages file located in the `/var/log` directory.

## CheckTable Help

To access the CheckTable menu-driven, hierarchical help system, press the **F7** key. Use the function keys to navigate within the help system.

To display the complete text of the help system, type `help;` at the CheckTable command prompt. For more information, see [HELP](#).

## CheckTable and System Activity

CheckTable can be run on both quiescent and non-quiescent systems. Before running CheckTable, consider the following:

- Table integrity validation requires that tables do not change during the CheckTable diagnostics. Consequently, CheckTable places a READ lock on each table as that table is being checked. You cannot perform an INSERT while CheckTable verifies consistency between the primary and fallback rows of a table.
- CheckTable places a READ lock on DBC.TVM momentarily to check for the existence of the table being checked. This lock might cause a problem when you create or modify tables, views, or macros, which require a WRITE lock on DBC.TVM.



- In some cases, CheckTable cannot perform a complete validation of table integrity if one or more AMPs are down.

To run CheckTable when users are logged on, and the system is non-quiescent, use the following CHECK options:

- **PRIORITY**

Sometimes CheckTable places a substantial resource demand (for example, CPU cycles, disk access, and spool space) on the system, degrading performance significantly for users accessing the system. By default, CheckTable performs table checking at MEDIUM priority.

To control the job scheduling based on the expected system workload and to improve performance, use the PRIORITY option.

- **CONCURRENT MODE**

This option reduces lock contention by optimizing the locking protocol and checking tables serially.

CONCURRENT MODE allows CheckTable to run to completion, and helps prevent deadlocks on non-quiescent systems. The locking protocol used by concurrent mode is optimized to minimize the number of locks required, and to use less restrictive lock types as much as possible. However, some locks are still required to avoid reporting false errors due to in-progress transactions.

While lock contention is minimized, some blocking is still expected on an active system. For example, a read lock is placed on a table while it is being checked. Therefore, update operations to the table will be blocked until the table check is complete.

The CONCURRENT MODE RETRY LIMIT option skips all locked tables and retries these locked tables after CheckTable finishes checking all specified tables not locked by other sessions.

IF RETRY LIMIT is ...	THEN CheckTable ...
not specified	keeps retrying the locked tables forever or until all tables are checked successfully. When trying to access a locked table, CheckTable waits a maximum of five minutes for the lock to be released. If the lock is not released in that time, CheckTable moves on to the next table. CheckTable will return to the locked table to try again.
greater than 0	will continue to retry until the RETRY LIMIT is reached.
equal to 0	will not retry skipped tables.
negative	displays an error message.

---

**Note:**

On non-quiescent systems and on quiescent systems with logons enabled, CheckTable defaults to CONCURRENT MODE with RETRY LIMIT set to one. Under these conditions, Data Dictionary checks are skipped.

---

## Checking Data Dictionary Tables

The Data Dictionary tables are checked only if the check specifies ALL TABLES (non-concurrent mode) or explicitly specifies to check database DBC. In this case, the Teradata system must be quiescent with logons disabled.

Data Dictionary processing defaults to batch scan mode for performance reasons. If a dictionary error occurs, processing automatically switches to serial scan mode for its more comprehensive error handling functionality. However, serial scan mode is slower, and on large systems can take significant time.

Use the STATUS command to check on the progress of the Data Dictionary scan.

Use the ABORT command to abort the Data Dictionary scan along with the CHECK command itself. To skip the Data Dictionary scan, issue the CHECK command while logons are enabled.

## File Corruption and CheckTable

CheckTable ensures that file system data structures are consistent. If CheckTable experiences any file corruption problem while checking the tables and databases, the table check in progress is aborted, and CheckTable displays an error message.

To find the error, run the SCANDISK command after CheckTable finishes checking the remaining tables. For information on SCANDISK, see [Ferret Utility \(ferret\)](#).

## Out-Of-Order Rows

CheckTable treats rows that are out-of-order as if they do not exist, and continues looking for the next row in the sequence. This can cause CheckTable to report out-of-order errors for subsequent rows, even if only a single row was out of order.

For example, given the sequence: 1, 2, 4, 6, 8, 20, 9, 10, 11, 12, 17, 18, 21, 25, CheckTable would see 9 as being out of ascending sequence. CheckTable would therefore ignore the 9, then continue looking for a row to follow 20. In this case, although 20 is likely the only row that is out of order, CheckTable would report errors for rows 9 through 18.

## Special Tables and Exceptions

CheckTable only checks the table headers of base global temporary tables and their instances; data rows of instances are not checked during level-one checking. For more information on level-one checking, see [Level One Checks](#).

Stored procedures, UDFs, and UDMs, store their text and object code internally as special tables that are not normally accessible to users. Join index rows, and hash index rows are also stored internally as tables. CheckTable checks all of these tables, in addition to checking ordinary data tables.

Some checks are not done for unhashed tables, and certain join indexes and hash indexes.

CheckTable does not include the DBC.ALL table when accounting for tables checked because it is not considered a conventional table.

CheckTable does not check tables whose rollback has been canceled.

For foreign tables (tables created to analyze data in external object storage, such as AWS S3 data), CheckTable checks that table headers exist and that the Data Dictionary information for such tables is consistent. The COMPRESSCHECK and CHECKINVALIDSI options do not apply to foreign tables. For foreign tables, the check level is irrelevant; all levels perform the same checks.

## Referential Integrity Terminology

The following table defines the Referential Integrity terminology that CheckTable uses.

Term	Definition
Primary Key	One or more columns that uniquely identify a table row.
Foreign Key	One or more columns in a table (the child table) that references the Primary Key in one or more tables (the parent tables).
Parent Table	Table having a Primary Key that is referenced by another table, the child table.
Child Table	Table having a Foreign Key that references the Primary Key in another table, the parent table.

For more information on referential integrity, see *Teradata Vantage™ - SQL Fundamentals*, B035-1141 and *Teradata Vantage™ - Database Design*, B035-1094.

## Check Levels

CheckTable offers several levels of data integrity checking. Each successive level performs most or all of the checks from the lower levels, and performs additional checks that are more thorough. For example, level-two checking performs checks similar to those from the pendingop and first level, and adds additional checks. Higher check levels are generally more resource intensive, and require more time.

At each level of checking, CheckTable inspects specific internal data structures. If CheckTable detects errors during these checks, it displays error messages that describe the nature of the errors. The message may be followed by additional information to show the location of the problem: AMP, subtable (primary data, fallback data, or index), and row or range of rows, if applicable.

### Note:

For more information about specific CheckTable error messages by number, see *Teradata Vantage™ - Database Messages*, B035-1096. For more information on the Hilbert R-tree structure used by geospatial indexes, see *Teradata Vantage™ - SQL Request and Transaction Processing*, B035-1142.

The following table indicates the specific internal data structures that are checked by each type of level check, and suggests when to use each level.

Check Level	Internal Data Structures Checked	When to use this level
Pendingop	<ul style="list-style-type: none"> <li>Data dictionary (if database DBC is checked)</li> <li>Table dictionary</li> <li>Table header</li> </ul>	<p>Use pendingop checking to check for tables with the following pending operations:</p> <ul style="list-style-type: none"> <li>FastLoad</li> <li>MultiLoad</li> <li>Restore</li> <li>Rebuild</li> <li>Replicate copy</li> <li>System reconfiguration</li> </ul>
One	<ul style="list-style-type: none"> <li>Data dictionary (if database DBC is checked)</li> <li>Table dictionary</li> <li>Table header</li> <li>Obsolete subtables</li> <li>Unique secondary indexes</li> <li>Nonunique secondary indexes</li> <li>ParentCount</li> <li>ChildCount</li> <li>Subtable of a given table</li> <li>Base global temporary tables</li> <li>Data subtables</li> <li>Large object subtables</li> <li>Reference indexes</li> <li>System-defined join indexes (SJIs) for temporal tables with temporal unique constraints</li> </ul>	<p>Use level-one checking only to isolate specific tables with errors. Then perform level-two or level-three checks on those specific tables.</p> <p><b>Note:</b></p> <p>When level-one checks use the DOWN ONLY option, the CheckTable results show only the subtables that have been marked as down.</p>
Two	<ul style="list-style-type: none"> <li>Data subtables</li> <li>Large object subtables</li> <li>Unique secondary indexes</li> <li>Nonunique secondary indexes</li> <li>Reference indexes</li> <li>System-defined join indexes (SJIs) for temporal tables with temporal unique constraints</li> </ul>	<p>Use level-two checking when checks by level one fail, and you require a detailed checking of consistency of row IDs, the checksum of primary and fallback rows, and hash codes.</p> <p><b>Note:</b></p> <p>When level-two checks use the DOWN ONLY option, the CheckTable results show only the subtables and regions (ranges of rows in subtables) that have been marked down.</p>
Three	<ul style="list-style-type: none"> <li>Data subtables</li> <li>Large object subtables (same checking as for Level Two)</li> <li>Unique secondary indexes</li> </ul>	<p>Use level-three checking rarely and only for specific diagnostic purposes, such as when an AMP is down.</p>

Check Level	Internal Data Structures Checked	When to use this level
	<ul style="list-style-type: none"> <li>• Nonunique secondary indexes</li> <li>• Reference indexes</li> <li>• System-defined join indexes (SJIs) for temporal tables with temporal unique constraints</li> </ul>	<p><b>Note:</b></p> <p>When level-three checks use the DOWN ONLY option, the CheckTable results show only the subtables and regions (ranges of rows in subtables) that have been marked down.</p>

**Note:**

Databases and tables within databases are checked in alphabetical order, with the exception of CHECK AGAIN. For table recheck processing, the table order is determined by the order of the table appearance within the error file.

***Level Pendingop Checks***

Level-pendingop checks table headers and displays a message if a table is in any of the following states:

- Pending FastLoad
- Pending Restore
- Pending Reconfig
- Pending Rebuild
- Pending Replicate copy
- Pending MultiLoad
- Number of primary index (PI) tables, if any, with and without column partitioning
- Number of usable primary AMP index (PA) tables, if any, with column partitioning.
- Number of unusable primary AMP index (PA) tables, if any, with column partitioning. These are tables that need to be altered to NoPI tables before they can be used because they were restored from archives to a differently-configured system.
- Number of NoPI tables, if any, with and without column partitioning.

***Level One Checks***

In addition to performing level-pendingop checks, a level-one check inspects the following:

- Specified system data structures
- Data subtables
- Large object (LOB) subtables
- Unique and nonunique secondary indexes (USIs and NUSIs), including geospatial indexes
- System-defined join indexes (SJIs)

The following table summarizes the specific level-one checks. These checks are in addition to the general checks performed for all check levels. For more information on these general checks, see [CheckTable General Checks](#).

Object Checked	What CheckTable Does
table dictionary or table header	<ul style="list-style-type: none"> <li>checks the partitioning definition in the table header for a partitioned table for validity for this operating environment, and checked for consistency and correctness.</li> <li>if the COMPRESSCHECK option is used, compares the compress multi-value and name of compression algorithm UDF (if specified) from the table header to the corresponding information stored in the CompressValueList column of the DBC. TVFields table.</li> </ul> <p>For more information on compression, see <i>Teradata Vantage™ - Database Design</i>, B035-1094 and <i>Teradata Vantage™ - Data Types and Literals</i>, B035-1143.</p> <p>To correct the table header or the partitioning of rows, use the ALTER TABLE statement with the REVALIDATE PRIMARY INDEX option. For more information on ALTER TABLE, see <i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i>, B035-1144.</p>
reference index	<ul style="list-style-type: none"> <li>checks whether any AMP has any reference index flagged as invalid.</li> <li>compares the primary row count with the fallback row count, if the index is hashed and the table was defined with fallback.</li> <li>compares the row count in the index subtable with the row count in the data subtable.</li> </ul>
data subtable	<ul style="list-style-type: none"> <li>compares the physical row count in the primary data subtable with the physical row count in the fallback data subtables per AMP in each cluster.</li> </ul> <p>“Physical row” means the low-level row structure that is stored by the file system. A physical row may store a table row or the data from one or more column partitions. This check is performed only for tables that have fallback.</p> <p><b>Note:</b></p> <p>This check requires access to all AMPs in a cluster. If any AMP in a cluster is down or unavailable, no AMPs in that cluster are checked.</p>
LOB subtable	<ul style="list-style-type: none"> <li>compares row counts in the LOB subtable.</li> <li>verifies that the primary copy of the LOB subtable matches the row count in the fallback subtables.</li> <li>verifies that the logical row count of the base data table matches that of each LOB subtable associated with the base table. Note that in cases where the associated LOB value is NULL, there will be no LOB row in the LOB subtable.</li> </ul> <p>In the case of LOB subtables, there is not a one-to-one correspondence in the row counts of the base data table and the LOB subtable, since LOBs may span multiple physical rows in the LOB subtable.</p> <p>For example if a base data table row points to a LOB that is 198,000 bytes in length, there will be four rows in the LOB subtable for this object (three rows containing 64,000 bytes and one row containing 6,000 bytes). When the counts are compared against the base data table, these four rows are counted by CheckTable as a single row.</p>
USI	<p>checks if any AMP has any USIs flagged as invalid. CheckTable compares the following:</p> <ul style="list-style-type: none"> <li>logical row counts between data subtables across all primary AMPs.</li> <li>logical row counts in corresponding USI subtables across all primary AMPs.</li> </ul>

Object Checked	What CheckTable Does
	<p>If the primary AMP is unavailable, and the table is fallback, then the appropriate fallback row counts are used instead. If the AMP is unavailable and the table is not fallback, this check is skipped.</p> <ul style="list-style-type: none"> <li>compares logical row counts in the USI subtable, on each primary AMP, with the row count in the corresponding USI subtables of the other AMPs in the same cluster. However, this is done only if the table has fallback.</li> </ul> <p>For USIs and NUSIs, an invalid index is not an error. Unless the index is excluded explicitly from the check, CheckTable issues a warning.</p> <p><b>Note:</b></p> <p>This check requires access to all AMPs in a cluster. If any AMP in a cluster is down or unavailable, no AMPs in that cluster are checked.</p>
NUSI	<ul style="list-style-type: none"> <li>checks if any AMP has any NUSIs flagged as invalid.</li> <li>compares logical row counts of NUSI subtables to row counts for the corresponding data subtables, primary and fallback.</li> <li>For geospatial indexes CheckTable performs additional checks of the integrity and correctness of the geospatial index R-tree structures.</li> </ul> <p>Because the NUSI rows reside on the same AMP as the data rows, this check can be run without having all AMPs online. However, relationships between data and index subtables on unavailable AMPs are not checked.</p> <p>For USIs and NUSIs, an invalid index is not an error. Unless the index is excluded explicitly from the check, CheckTable issues a warning.</p>
SJI	<ul style="list-style-type: none"> <li>compares the row counts of SJIs to row counts for the corresponding temporal primary data subtable (or fallback subtable, if an AMP in the cluster is down) to ensure counts are appropriate to the types of temporal table and temporal unique constraint.</li> </ul> <p>SJIs are generated by Vantage for temporal tables that have temporal unique constraints. For more information on temporal tables and constraints, see <i>Teradata Vantage™ - Temporal Table Support</i>, B035-1182.</p>

## Level Two Checks

In addition to performing level-pendingop and level-one checks, level-two checking does the following:

- Determines whether row IDs on any given subtable are consistent with row IDs on other subtables by comparing lists of these IDs in those objects.
- Computes and compares the checksum of primary and fallback rows.
- Verifies that hash codes reflect correct row distribution in each subtable.

The following table summarizes the specific level-two checks. These checks are in addition to the general checks performed for all check levels. For more information on these general checks, see [CheckTable General Checks](#).

Object Checked	What CheckTable Does
data subtable	<ul style="list-style-type: none"> <li>performs checksum calculations for primary and corresponding fallback rows, and detects duplicate, out-of-order, and incorrectly distributed rows.</li> <li>for fallback tables, it checks primary against fallback.</li> <li>only row IDs of physical rows are checked for column-partitioned objects.</li> </ul> <p>Level-two checks are performed on all primary and fallback data rows on all online AMPs. CheckTable does not check data on down AMPs. If an AMP is unavailable, the primary-to-fallback data check bypasses rows whose alternate (primary or fallback) copy would be expected to be on the unavailable AMP. However, CheckTable still scans all subtables to check for problems other than primary-to-fallback inconsistencies.</p>
LOB subtable	<ul style="list-style-type: none"> <li>checks for duplicate row IDs, out-of-order row IDs, and incorrectly distributed rows in the LOB subtables.</li> <li>checks for missing primary or fallback row, where fallback exists for LOBs.</li> <li>compares checksum of the primary and fallback copies of the LOB row.</li> <li>verifies that hash codes reflect correct row distribution in each subtable.</li> <li>checks for stale locators/OIDs where the update tag of the OID in the base row does not match the update tag in the large LOB row.</li> <li>checks for gaps in the LOB, for example, if the first and third portions of the LOB are present, but the second piece is missing from the subtable.</li> </ul>
USI	<ul style="list-style-type: none"> <li>compares the list of row IDs indexed in the primary or fallback index subtable with logical row IDs in primary or fallback data subtable copy.</li> <li>for tables with fallback, compares row IDs in fallback copy to logical row IDs in primary.</li> <li>detects non-indexed data rows, indexed data rows that no longer exist in the data subtable, and multiply indexed data rows.</li> <li>checks for duplicate, out-of-order, and incorrectly distributed row IDs in the primary and fallback data subtables, if such checks have not already been performed during a different phase of checking.</li> </ul> <p>For checks that compare primary to fallback rows, CheckTable checks only tables that have fallback.</p> <p>CheckTable does not check rows whose primary or fallback copies are inaccessible due to an unavailable AMP.</p>
NUSI	<ul style="list-style-type: none"> <li>checks for duplicate, out-of-order, and incorrectly distributed row IDs, if such checks have not already been performed during a different phase of checking.</li> <li>detects non-indexed data rows, indexed data rows that no longer exist in the data subtable, and multiply indexed data rows.</li> <li>checks any indexed data row IDs that belong to another AMP or belong in another subtable.</li> <li>compares lists of row IDs (indexed by index rows in the primary index subtable and each fallback index subtable) with the actual list of logical row IDs of data rows in the corresponding data subtables to ensure the list is in order.</li> <li>For geospatial indexes CheckTable performs additional checks of the integrity and correctness of the geospatial index R-tree structures.</li> </ul>
reference index check	<ul style="list-style-type: none"> <li>checks specified reference index subtables for duplicate, out-of-order row IDs, and incorrectly distributed rows.</li> </ul>



Object Checked	What CheckTable Does
	<ul style="list-style-type: none"> <li>checks whether reference rows in the primary reference index subtable have corresponding rows in the fallback subtables.</li> </ul> <p><b>Note:</b> If the reference index subtable is fallback, Vantage uses the row IDs of the reference index rows from the primary copy of the subtables.</p>
SJI	<ul style="list-style-type: none"> <li>checks subtables for specified SJIs to detect non-indexed valid data rows, indexed valid data rows that no longer exist in the corresponding temporal table, and valid data rows that are indexed more than once.</li> <li>checks subtables for specified SJIs for duplicate, out-of-order row IDs, closed rows, and incorrectly distributed rows.</li> <li>compares the list of row IDs in the SJI with the actual row IDs of valid rows in the corresponding temporal table.</li> </ul> <p>SJIs are generated by Vantage for temporal tables that have temporal unique constraints. For more information on temporal tables and constraints, see <i>Teradata Vantage™ - Temporal Table Support</i>, B035-1182.</p>

**Note:**

If you run SCANDISK and correct any problems before using CheckTable, then it should not detect any duplicate or out-of-order rows during level-two checking.

**Spool Space Requirements for Level Two Checks**

Critical to level-two and level-three checking is the amount of spool space available on the system. The following formulas determine the amount of spool space required to perform these two levels of checking:

- $RID = 32,020 * (\# \text{ of rows} / 3,200)$
- $SIS = (52 * \# \text{ of rows}) + (3,000,000 * \# \text{ of AMPs})$

To determine the number of rows, count the rows in the primary data table, which is the same as the primary index table. The following table shows the values and spool space required in the above formulas.

The value ...	Is the spool space required ...
RID	for the data subtable.
SIS	by the largest secondary index subtable, whether it is unique or nonunique.

Therefore, to perform a level-two check for a nonfallback table that involves the secondary index, SIS requires a total spool space equal to the following:

$$RID + (SIS * 2)$$

If the table contains fallbacks, then SIS requires a total spool space equal to the following:

$$2 * (RID + (SIS * 2))$$

If you specify PARALLEL mode, then the total required spool space equals the sum of the spool space required for each of the tables being checked.

### ***Level Three Checks***

Level-three checking provides the most detailed check and requires more system resources than any of the other checking levels. Because of the cost in system resources, use this checking level rarely and only for specific diagnostic purposes.

In addition to checks that are unique to this level, level-three checking includes most of the same checks that are performed at lower check levels. While no additional level-three checks are performed on large objects (LOBs), a level-three check will automatically perform level-two checking of LOBs.

If an AMP is unavailable and the table was not defined with fallback, CheckTable bypasses all USI checks.

If an AMP is unavailable and the table is fallback, then CheckTable uses the fallback copies of index and data subtables on the available AMP in place of the primary copies on the unavailable AMP.

The following table summarizes the specific level-three checks. These checks are in addition to the general checks performed for all check levels. For more information on these general checks, see [CheckTable General Checks](#).

Object Checked	What CheckTable Does
data subtable	<ul style="list-style-type: none"> <li>• for tables that have fallback, compares each physical row in the primary data subtable with the corresponding physical row in the fallback data subtable. Makes a byte-by-byte comparison at all levels.</li> <li>• checks for duplicate, out-of-order, or incorrectly distributed rows, and duplicate unique primary keys.</li> <li>• verifies row hashes.</li> <li>• verifies the field size of each field in the data row with the defined maximum field size for the corresponding column. The field sizes are checked in the primary data row, not the fallback data row. Any field size error in the fallback data rows is discovered in the byte-by-byte comparison between the primary and fallback data rows.</li> <li>• for row-partitioned tables, verifies the internal partition numbers. This check is not performed for column-partitioned tables.</li> <li>• for column-partitioned tables, the physical data row IDs of each container row is verified to be greater than the physical row IDs of the previous container row, if any, plus the number of logical row IDs minus one represented by that container row. Container rows are verified for the integrity of their format.</li> <li>• for temporal tables, checks for value equivalent rows with overlapping temporal time values.</li> <li>• If the COMPRESSCHECK option was specified, CheckTable verifies compressed column data. For example, for multi-value compressed columns, CheckTable compares</li> </ul>

Object Checked	What CheckTable Does
	<p>the compress list in the table header to the column values in each row to ensure the appropriate values were compressed.</p> <p>If an AMP is unavailable, the primary-to-fallback data check bypasses rows whose alternate (primary or fallback) copy would be expected to be on the unavailable AMP. However, CheckTable still scans all subtables to check for problems other than primary-to-fallback inconsistencies</p> <p>For multilevel partitioned primary indexes, the external partition number in the error message is the number resulting from the combined partition expression. For more information on partitioned primary indexes and on partitioning expressions, see <i>Teradata Vantage™ - Database Design</i>, B035-1094.</p> <p>To correct the partitioning of rows for a table, use the ALTER TABLE statement with the REVALIDATE PRIMARY INDEX WITH DELETE or INSERT option. For more information on ALTER TABLE, see <i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i>, B035-1144.</p>
LOB subtable	<ul style="list-style-type: none"> <li>• checks for duplicate row IDs, out-of-order row IDs, and incorrectly distributed rows in the LOB subtables.</li> <li>• checks for missing primary or fallback row, where fallback exists for LOBs.</li> <li>• compares checksum of the primary and fallback copies of the LOB row.</li> <li>• verifies that hash codes reflect correct row distribution in each subtable.</li> <li>• checks for stale locators/OIDs where the update tag of the OID in the base row does not match the update tag in the large LOB row.</li> <li>• checks for gaps in the LOB, for example, if the first and third portions of the LOB are present, but the second piece is missing from the subtable.</li> </ul>
USIs	<ul style="list-style-type: none"> <li>• detects non-indexed data rows, indexed data rows that no longer exist in the data subtable, and multiply indexed data rows.</li> <li>• checks for duplicate, out-of-order, and incorrectly distributed row IDs in primary and fallback index subtables.</li> <li>• verifies the field size of each field in the USI row with the defined maximum field size for the corresponding column. The field sizes are checked in the primary USI rows, not the fallback USI rows. Any field size error in the fallback USI rows is discovered in the byte-by-byte comparison between the primary and fallback USI rows.</li> <li>• for tables with fallback, makes a byte-by-byte comparison of primary and fallback index subtables.</li> <li>• compares the list of row IDs indexed in the primary or fallback index subtable with row IDs in primary or fallback data subtable.</li> <li>• compares the key of each USI row to the expected key extracted from the data row.</li> <li>• verifies row hashes</li> <li>• verifies that every logical row is indexed only once.</li> <li>• checks for duplicate, out-of-order, and incorrectly distributed row IDs in the primary and fallback data subtables, if such checks have not already been performed during a different phase of checking.</li> </ul>

Object Checked	What CheckTable Does
	<p><b>Note:</b></p> <p>Checking for duplicates on MULTiset tables with unique indexes can be expensive with respect to performance. When there are many rows in such a table that have the same hash value, this check can take a long time. This would be the case for a table with a highly nonunique PI. For each hash value, CheckTable scans through the entire hash, row by row, and checks for duplicates on the unique index. Since NoPI and PA tables generally have only one hash value per AMP, USI checks can be slow.</p>
NUSIs	<ul style="list-style-type: none"> <li>• checks that each data row on index subtables is indexed with the correct index key.</li> <li>• checks that the hash code for each NUSI row corresponds to the key value of the row.</li> </ul> <p><b>Note:</b></p> <p>This check is not valid for geospatial indexes.</p> <ul style="list-style-type: none"> <li>• checks for duplicate and out-of-order index rows.</li> <li>• checks for duplicate, out-of-order, and incorrectly distributed row IDs in the primary and fallback data subtables, if such checks have not already been performed during a different phase of checking.</li> <li>• verifies the field size for each field in the primary and fallback NUSI rows with the defined maximum field size for the corresponding column.</li> <li>• for geospatial indexes CheckTable performs additional checks of the integrity and correctness of the geospatial index R-tree structures.</li> </ul>
reference index subtables	<ul style="list-style-type: none"> <li>• checks for duplicate index values.</li> <li>• checks for incorrect hash code.</li> <li>• checks reference index subtables against parent tables. If CheckTable detects an error, it places a read lock on the parent table specified in the reference index. If the AMP cannot lock the parent table, CheckTable reports an error.</li> <li>• verifies the field size for each field in the reference index row with the defined maximum field size for the corresponding column.</li> </ul>
SJIs	<ul style="list-style-type: none"> <li>• checks for out of order, duplicate, misplaced row IDs, or closed rows in SJI.</li> <li>• compares the SJI contents with the corresponding columns in the base temporal table rows.</li> <li>• checks for value equivalent rows that have overlapping temporal values.</li> <li>• verifies the field size for each field in the SJI row with the defined maximum field size for the corresponding column.</li> </ul>

As data subtables are scanned, the internal partition number of each table row is verified if the table has row partitioning. If a scanned internal partition number does not match the number calculated using the combined partitioning expression, CheckTable displays an error message. If the table is column partitioned, this check is not performed.

## CheckTable General Checks

CheckTable checks certain aspects of tables regardless of the level of checking that is specified. These general checks include things like checking the table headers, checking for obsolete subtables, and checking the Data Dictionary.

Check Type	Description
Data Dictionary	<p>Two special checks involve the Data Dictionary:</p> <ul style="list-style-type: none"> <li>• <b>Table List Validation</b> This check compares the table IDs found in the DBC.TVM and DBC.TempTables with the table IDs on each AMP. It looks for missing tables (tables found in the dictionary but not found on one or more AMPs), extra tables (tables found on one or more AMPs that are not found in the dictionary), and table data that resides on one or more AMPs where those AMPs are not in the defined map for the table.</li> <li>• <b>Contiguous Map Info Validation</b> This check compares contiguous map information stored in the DBC.Maps table with the corresponding information stored in the MapInfo GDO. The MapInfo GDO is a memory-resident data structure that stores information about all contiguous maps on the system. For more information about contiguous and sparse maps, see <i>Teradata Vantage™ - Database Design</i>, B035-1094, <i>Teradata Vantage™ - Database Administration</i>, B035-1093, and <i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i>, B035-1144.</li> </ul> <p>These special checks are only performed when the user issues a CHECK DBC or CHECK ALL TABLES command and the system has logons disabled and is quiescent.</p>
Table headers	Compares table header components on all AMPs with the table header from the AMP with the lowest virtual processor (vproc) number of all online AMPs.
Table structure	Compares the table structure version stored in DBC.TVM to the table structure version stored in the header.
Obsolete subtables	checks for extraneous subtables of a given table.
Parent/Child count	Verifies if the ParentCount and ChildCount in DBC.TVM match the ParentCount and ChildCount in the table header.
Table Map Info	Verifies that the map information stored in DBC.TVM and DBC.MAPS match the map information stored in table headers.
Pending operations	Checks the table header to determine if any table has any pending operations. This check works for global temporary tables as well as base tables.

### Note:

Tables with online archive active logs are checked by CheckTable, however their online archive log subtables are not checked.

## CheckTable and Deadlocks

Because the system does not detect deadlocks that exist between utility functions, CheckTable has a built-in timeout feature to prevent long-standing deadlocks. However, this feature makes it difficult for CheckTable to apply a lock on a table that has frequent lock requests.

For non-concurrent mode, the first time a lock is requested, the time-out interval is one minute. Each subsequent retry adds one minute to the interval until the interval is five minutes. Then, all subsequent retries employ a five-minute time-out. The number of times CheckTable can attempt a lock request is not limited.

If you specify the SKIPLOCKS option, then CheckTable requests a lock on a table only once. If CheckTable does not obtain the lock the first time, then CheckTable does not request any further locks and does not check the table.

To determine the current status of CheckTable, type status at the CheckTable command prompt.

To abort the table check, enter “abort table” to abort the current table check, or “abort” to abort the current CHECK command and all pending table checks.

For more information, see [CheckTable and System Activity](#).

For non-concurrent mode, when the CheckTable Table Lock Retry entry in the DBS Control General fields is set to a positive number, CheckTable will retry the lock request until this specified time in minutes is exhausted. CheckTable will display that the table is skipped due to pending lock and will continue to check the next table. For more information, see [ChecktableTableLockRetryLimit](#) in [DBS Control \(dbscontrol\)](#)

For concurrent mode, the first time a table lock is requested, the timeout interval is 15 seconds. CheckTable will skip the locked table and continue to check the remaining tables. All subsequent retries employ a five-minute timeout interval per table. All skipped tables will be retried forever if RETRY LIMIT is not specified. If RETRY LIMIT is positive, all skipped tables will be retried until CheckTable reaches the RETRY LIMIT. For more information on concurrent mode, see [CheckTable and System Activity](#).

## CheckTable Messages

CheckTable issues the following types of messages:

- Syntax error
- Check completion
- Error

## Syntax Error Messages

When a syntax error occurs, CheckTable does the following:

- Displays the part of the input line where the error occurs
- Prints a dollar sign (\$) in the command input line beneath the character that caused the error

- Displays an error message

For example, assume the following command:

```
CHECK db[2-] AT LEVEL ONE;
```

The following error message appears:

```
*** Syntax error ***
CHECK db[2-] AT LEVEL ONE;
      $
Invalid range specified in dbname or dbname.tablename.
```

## Error Messages

CheckTable aborts when it encounters problems, depending on the severity and context of the problem. CheckTable can crash the database on rare occasions, such as when unable to clean up resources allocated by the database.

When CheckTable aborts, it does the following:

- Displays an error message
- Saves a snapshot dump for further analysis
- Retains backtrace information

The backtrace information is located in the event log file, `/var/log/messages`.

---

### Note:

For more information on specific CheckTable error messages by number, see *Teradata Vantage™ - Database Messages*, B035-1096.

---

## Auxiliary Information

Auxiliary information included in messages may include a table row identifier.

A RowID consists of a 16-byte value that includes the following information:

- The first 8-bytes is the internal partition number of the row.  
(For a nonpartitioned table, the internal partition number is zero, and the internal partition number is not actually stored in the row itself.)
- The next 8-bytes include a hash bucket value and uniqueness value for the row.

For more information on internal table row formats, see *Teradata Vantage™ - Database Design*, B035-1094.

## Error Handling

While CheckTable is processing, a system failure could occur for a variety of reasons, such as the following:

- Programming errors
- Hardware problems

After a system failure, one of the following might occur:

- CheckTable aborts, displaying a 7496 or 7492 error message. You should inspect the message log for a backtrace or other related messages.
- CheckTable hangs. Verify the status using the `status` command. If you do not detect any problems, then try to stop and restart CheckTable.
- The partition in which CheckTable is running resets.

## Valid Characters in Object Names

The names of databases, tables, other objects or workload names specified in the `CHECK` command can consist of the following inclusive characters:

- Lowercase alphabet (a ... z)
- Uppercase alphabet (A ... Z)
- Digits (0 ... 9)
- The following special characters.

Special characters ...	Include ...
Parentheses, braces, and brackets	<ul style="list-style-type: none"> <li>◦ ( ) (parentheses)</li> <li>◦ { } (curly braces)</li> <li>◦ [ ] (square brackets)</li> <li>◦ &lt; &gt; (angle brackets)</li> </ul>
Punctuation marks	<ul style="list-style-type: none"> <li>◦ ` (grave accent)</li> <li>◦ ! (exclamation point)</li> <li>◦ ; (semicolon)</li> <li>◦ : (colon)</li> <li>◦ ' (apostrophe)</li> <li>◦ ? (question mark)</li> <li>◦ . (period)</li> <li>◦ , (comma)</li> </ul>
Other	<ul style="list-style-type: none"> <li>◦   (vertical line)</li> <li>◦ ~ (tilde)</li> <li>◦ @ (at sign)</li> </ul>



Special characters ...	Include ...
	<ul style="list-style-type: none"> <li>◦ \$ (dollar sign)</li> <li>◦ = (equals sign)</li> <li>◦ % (percent sign)</li> <li>◦ + (plus)</li> <li>◦ # (number sign)</li> <li>◦ ^ (circumflex accent or caret)</li> <li>◦ &amp; (ampersand)</li> <li>◦ * (asterisk)</li> <li>◦ - (hyphen-minus)</li> <li>◦ _ (low line or underscore)</li> <li>◦ / (forward slash)</li> <li>◦ \ (backward slash)</li> </ul>

**Note:**

Workload names can only include the following: letters, numbers, underscores, hyphens-minuses, and spaces.

You must specify any name containing one or more special characters or blank spaces within apostrophes or double quotation marks, except for the following:

- ? (question mark)
- % (percent sign)
- \$ (dollar sign)
- \_ (low line or underscore)
- [ ] (square brackets)
- # (number sign)

**Note:**

A name cannot begin with a digit (0 ... 9).

For more information on creating names, see *SQL Fundamentals*.

**Examples of Database and Table Names**

The following examples show valid database or table names:

- Table1
- MYTABLE10
- \$\$MyAccount
- #Your\_Account\_\$100
- %mydatabase?

- %
- ???

The following examples show irregular but acceptable names:

- '123'
- "First&Second table"
- 'my db1'

The following examples show unacceptable and non-valid names:

- 123
- First&Second table
- my db1

## Wildcard Characters in Names

Use wildcard characters % and ? to specify a pattern for database names or table names.

The following table shows how CheckTable interprets wildcard characters.

Wildcard character ...	Matches any ...
% (percent sign)	string of characters of any length, including the Null string.
? (question mark)	single character.

You can use wildcard characters in any combination. However, you cannot use wildcard characters in hexadecimal form.

The following table shows the use of wildcard characters in names.

Wildcard character ...	Matches any ...
%	database name.
%.%	table name.
%database%	database name containing the string: database.
SalesDB%	database name beginning with the following: SalesDB.

CheckTable supports the use of wildcard syntax to represent a list of possible characters at a particular position in the names of databases or tables. Use the wildcard syntax to specify lists of tables and databases you want CheckTable to check or not check. The wildcard syntax begins with a left square bracket ([) and ends with a right square bracket (]).

Other special characters can appear in table or database names but not in wildcard syntax. If any syntax error occurs in the wildcard syntax, then CheckTable aborts, and an error message appears.

For rules regarding use of Kanji and other Japanese characters in names, see *Teradata Vantage™ - SQL Fundamentals*, B035-1141. For information on syntax error messages, see [Syntax Error Messages](#).

## Wildcard Syntax

```
{ start [ remainder ... ] | "char [ char ]..." | 'char [ char ]...' }
```

### start

```
{ starting_char | [ { start_range_char | alphabet_range }... ] }
```

#### Note:

You must type the colored or bold brackets.

### remainder

```
{ remaining_characters | [ { start_range_char | digit | hyphen_range } ] }
```

#### Note:

You must type the colored or bold brackets.

## Wildcard Syntax Elements

### char

One of the following:

- alphabet (uppercase or lowercase)
- digit between 0 and 9
- special character
- Kanji character

### starting\_char

One of the following:

- alphabet (lowercase or uppercase)
- ? (question mark)
- % (percent sign)

***start\_range\_char***

One of the following:

- alphabet (lowercase or uppercase)
- \$ (dollar sign)
- \_ (low line or underscore)
- # (number sign)

***alphabet\_range***

Two alphabets separated by a hyphen.

The range can be in ascending or descending order. Both the alphabets should be the same type, either uppercase or lowercase.

***remaining\_characters***

One of the following:

- alphabet (uppercase or lowercase)
- digit between 0 and 9
- ? (question mark)
- % (percent sign)
- \$ (dollar sign)
- \_ (low line or underscore)
- # (number sign or hash mark)

***digit***

Any digit between 0 and 9.

***hyphen\_range***

Two alphabets or two digits separated by a hyphen.

The range can be in ascending or descending order. Both the characters should be the same type, either uppercase or lowercase alphabet or digit.

***Usage Notes***

The following rules apply to the use of wildcard syntax in the CHECK command. Assume that the databases and tables in the examples exist in the system, unless stated otherwise.

Rule	Example
<p>You can specify the following valid ASCII characters in the wildcard syntax:</p> <ul style="list-style-type: none"> <li>• A ... Z</li> <li>• a ... z</li> <li>• 0 ... 9</li> <li>• _ (low line or underscore)</li> <li>• \$ (dollar sign)</li> <li>• # (number sign)</li> </ul> <p>You cannot use digits 0 ... 9 as wildcards to describe the first character in the name.</p>	<p><b>Example 1:</b> The following is a valid command:</p> <pre>CHECK db1.t[#af_2r]1 AT LEVEL ONE;</pre> <p><b>Example 2:</b> The following is not a valid command:</p> <pre>CHECK db[#,kA-d159]xy AT LEVEL ONE;</pre> <p>The above command results in a syntax error because the wildcards specified for database name include the non-valid comma (.). For information on syntax error messages, see <a href="#">Syntax Error Messages</a>.</p>
<p>You must specify the wildcard characters within square brackets. The wildcard syntax begins with a left square bracket ([) and ends with a right square bracket (]).</p>	<p><b>Example 1:</b> Databases db1, db2, db3, db4, and db5 exist, and you want only the tables in db1 and db5 checked. Type the following:</p> <pre>CHECK db[15] AT LEVEL ONE;</pre> <p>CheckTable checks all the tables in databases db1 and db5 at level one. The wildcard syntax defines two possible values (1 and 5) for the third character in the database name.</p> <p><b>Example 2:</b> Databases db1, dc1, dd1, and so on exist, and each database contains tables t1, t2, t3, and so on. Using the wildcard syntax in any place in the name, type the following:</p> <pre>CHECK d[bd]1.t[123] AT LEVEL ONE;</pre> <p>CheckTable checks tables t1, t2, t3 in databases db1 and dd1.</p> <p><b>Example 3:</b> To specify wildcard syntax in multiple places in a name, type the following:</p> <pre>CHECK db[12][pq] AT LEVEL TWO;</pre> <p>CheckTable checks databases db1p, db2p, db1q, and db2q at level two. The wildcard syntax defines the possible values for the third and fourth characters of the database name.</p>
<p>You cannot specify the special characters % and ? within wildcard syntax. However, you can use the special characters % and ? with any valid wildcard syntax.</p>	<p><b>Example 1:</b> Databases dba1, dba2, db11 and db12 exist, and you want to check databases dba1, dba2, db11, and db12. Type the following:</p> <pre>CHECK db[a1]? at level one;</pre> <p>This command is valid, because the '?' is outside the wildcard syntax.</p> <p><b>Example 2:</b> The following is not a valid command, because the '?' is not allowed in wildcard syntax.</p> <pre>CHECK db[a1?] at level one;</pre>

Rule	Example
<p>You can use wildcard syntax to specify the names or lists of the databases and tables to check and the list of databases or tables not to check.</p>	<p><b>Example 1:</b> Databases db1, db2, db3 and db4 exist, and you type the following:</p> <pre>CHECK db% exclude db[34] at level one;</pre> <p>Databases db1 and db2 are checked.</p> <p><b>Example 2:</b> Databases db1, db2, db3 and db4 exist, and all these databases have tables t1, t2, t3 and t4. You type the following:</p> <pre>CHECK db[23] exclude t[14] at level one;</pre> <p>CheckTable checks tables t2 and t3 in databases db2 and db3.</p>
<p>You can use wildcard syntax to specify a range of characters by separating two characters with a hyphen (-). For example, C and J separated by the hyphen (C-J) represent any characters lexically between C and J inclusive.</p> <ul style="list-style-type: none"> <li>The two characters should be of the same type: uppercase, lowercase, or digit.</li> <li>The two characters can be in ascending or descending lexical order. For example, [A-D] and [D-A] both specify the same range of characters: A through D inclusive.</li> </ul>	<p><b>Example 1:</b></p> <pre>CHECK db1.t[1-35] AT LEVEL ONE;</pre> <p>CheckTable checks the tables t1, t2, t3, and t5 in database db1 at level one. 1-3 is considered a range, and 5 is an additional value.</p> <p><b>Example 2:</b></p> <pre>CHECK db[a-5] AT LEVEL ONE;</pre> <p>The check does not take place. CheckTable reports a syntax error because the range specified in dbname is invalid. For information on syntax error messages, see <a href="#">Syntax Error Messages</a>.</p>
<p>Wildcard syntax can include characters that might not have any matching object names in the system. If the syntax contains some characters that do not have a match at the position specified in any object names in the system, CheckTable checks (or excludes from checking) all the objects whose names match the specified wildcards. CheckTable also ignores the characters that do not have any matching objects. This is true of any number of wildcards.</p>	<p><b>Example 1:</b> Assume a system contains only databases db1 and db5 but not db2, db3, and so on. Type the following:</p> <pre>CHECK db[125] AT LEVEL ONE;</pre> <p>CheckTable checks all the tables in databases db1 and db5 at level one. Since database db2 does not exist, CheckTable ignores character 2 in the wildcard syntax.</p> <p><b>Example 2:</b> Assume a system contains the database db1 but not db2, db3, or db4. Type the following:</p> <pre>CHECK db[1-4] AT LEVEL ONE;</pre> <p>CheckTable checks all the tables in the database db1 and ignores the remaining wildcard characters.</p>
<p>Multiple occurrences of the same character in the wildcard syntax are valid. If you repeat the same character in the syntax for the same position, then CheckTable recognizes the first occurrence and ignores the repeated instances.</p>	<p><b>Example 1:</b> In the following command, character b is repeated in the same position.</p> <pre>CHECK d[abb]1 AT LEVEL ONE;</pre> <p>CheckTable checks all tables in the databases da1 and db1 at level one and ignores the second instance of character b. No warning appears.</p>

Rule	Example
	<p><b>Example 2:</b> In the following command, character 3 is specified as part of the hyphen range 1-5 and is repeated separately in the same position.</p> <pre>CHECK db[1-53] AT LEVEL ONE;</pre> <p>CheckTable checks all tables in the databases db1, db2, db3, db4, and db5 at level one. CheckTable ignores the repeated character 3.</p>
The wildcard syntax does not apply when enclosed between apostrophes or double quotation marks.	<p>In the following command, character p is a wildcard enclosed in double quotation marks.</p> <pre>CHECK db1."[p]" AT LEVEL ONE;</pre> <p>CheckTable ignores the square brackets and checks only table "[p]", if it exists in database db1. If table "[p]" does not exist in db1, then a warning appears.</p>

## Examples

### Example: Using wildcard syntax

```
CHECK DB[15] AT LEVEL ONE;
```

The wildcard syntax defines two possible values (1 and 5) for the third character in the database name. CheckTable checks all the tables in databases DB1 and DB5 at level one.

### Example: Using multiple wildcard syntax

```
CHECK D[BD]1.T[123] AT LEVEL ONE;
```

You can use the wildcard syntax in any place in the database name or table name. CheckTable checks tables T1, T2, T3 in databases DB1 and DD1.

## CHECK Command Examples

This section shows some examples using the options of the CHECK command.

### Example: CHECK ALL TABLES EXCLUDE

To exclude one or more databases or tables from the check, use CHECK ALL TABLES EXCLUDE. If a specified object does not exist in the system, then the object appears in a message at the end of the summary report.

CheckTable does the following:

- Checks the dictionary and database DBC.  
(If database DBC is in the EXCLUDE list, it is not checked.)
- Checks other non-excluded databases in database-name order

The following table shows different CHECK ALL TABLES EXCLUDE examples based on database DBC at level-one checking.

IF you want to check ...	THEN type ...
the Data Dictionary and all databases except DBC, SalesDB1, and PurchaseDB1	CHECK ALL TABLES EXCLUDE DBC, SalesDB1, PurchaseDB1 AT LEVEL ONE;
only the Data Dictionary	one of the following: • CHECK ALL TABLES EXCLUDE % AT LEVEL ONE; • CHECK ALL TABLES EXCLUDE %.% AT LEVEL ONE;
all tables except those in db1 and t2 and t4 of db2	CHECK ALL TABLES EXCLUDE db1,db2.t2, db2.t4 AT LEVEL THREE;
all tables except those in database1 with table names beginning with Sales	CHECK ALL TABLES EXCLUDE database1.Sales% AT LEVEL ONE;

### Example: CHECK ALL TABLES AT LEVEL PENDINGOP

To check all tables to determine if they have any pending operations, type the following:

```
CHECK ALL TABLES AT LEVEL PENDINGOP;
```

The following output appears:

```
Check beginning at 11:13:31 11/06/26.
```

```
Data dictionary check started at 11:13:31 17/06/26.
```

```
...
```

```
Table: "PROD"."CUSTOMERS" starting at 11:14:22 17/06/26.
```

```
Map No 1025, Table id 0000H 624BH, Database id 0000H 03FCH, No fallback.
```

```
Table check bypassed due to pending MultiLoad.
```

```
Table Result: Skipped
```

```
Table: "PROD"."SHIPMENT" starting at 11:14:24 17/06/26.
```

```
Map No 1025, Table id 0000H 04E1H, No fallback.
```

```
Table check bypassed due to pending Table Rebuild.
```

```
...
```

```
Summary:
```

```
1,435 table(s) checked.
```

```
420 fallback table(s) checked.
```



```
1,015 non-fallback table(s) checked.
```

```
1 table(s) bypassed due to pending MultiLoad.
```

```
1 table(s) bypassed due to pending Table Rebuild.
```

```
0 table(s) failed the check.
```

```
0 Dictionary error(s) were found.
```

```
Check completed at 11:14:58 17/06/26.
```

### Example: CHECK *dbname* EXCLUDE

To check all tables in a specified database, except for those listed after EXCLUDE, use

```
CHECK dbname EXCLUDE
```

You can use wildcards in the EXCLUDE list. For example:

```
check all tables exclude mydb% at level two
```

will exclude database having names starting with mydb.

The following table shows different CHECK *dbname* EXCLUDE examples at level-one checking.

IF you want to ...	THEN type ...
check database dbname1 except for tables t1, t2, t3	CHECK dbname1 EXCLUDE t1, t2, t3 AT LEVEL ONE;
check database dbname2 except for tables beginning with the word table and followed by any single character as well as tables containing the string weekly	CHECK dbname2 EXCLUDE table?, %weekly% AT LEVEL ONE;
exclude any tables whose names either begin with week1 and end with any character or begin with the word month in the SalesDB database	CHECK SalesDB EXCLUDE week1?, month% AT LEVEL ONE;

For more information, see [Wildcard Characters in Names](#). If a specified table includes a *dbname*, then CheckTable only checks the table in that referenced database.

### Example: NO ERROR LIMIT

To perform level-two checking with no error limits on the MfgDb database, type the following:

```
CHECK MfgDb AT LEVEL TWO WITH NO ERROR LIMIT;
```

Output similar to the following appears.

```
Check beginning at 13:08:03 17/04/26.
```

```
Table: "MFGDB"."INVENTORY" starting at 13:08:06 17/04/26.
```

```
Map No 1025, Table id 0000H 0C31H, Database id 0000H 03FCH, Fallback.
```

```
Table result: No errors.
```

```
Table: "MFGDB"."PARTS" starting at 13:08:07 17/04/26.
```

```

Map No 1025, Table id 0000H 0C33H, Database id 0000H 03FCH, No fallback.
  2753: Primary data row is on wrong AMP.
    AMP 00000
    Row id 0000H 79B6H 9E37H 0000H 0001H
    Expected primary AMP 00001
  Table result: Error(s) reported.
Table: "MFGDB"."RETURNS_TEMP" starting at 13:08:08 17/04/26.
  Map No 1025, Table id 0000H 0C32H, Fallback.
  No errors reported.
Summary:
  3 table(s) checked.
  2 fallback table(s) checked.
  1 non-fallback table(s) checked.
  1 table(s) failed the check.
  0 Dictionary error(s) were found.
Check completed at 13:08:09 17/04/26.

```

In the above output, the first line of the output is the header, which is displayed before the first table is checked. The header shows the exact time and date that you started the check in the following format:

Check beginning at HH:MM:SS YY/MM/DD

If one or more AMPs are unavailable, then the following message appears:

```

The following AMPs are not operational.  As a result, certain checks will not
be complete:
nnnnn  nnnnn... ...

```

Unavailable AMPs are indicated by the format *nnnnn*.

The next set of lines displayed indicates the database name, table name, map number, internal ID, database ID, fallback status, and the date and time:

```

Table: "dbname"."tablename" starting at YY/MM/DD HH:MM:SS
Map No nnnn, Table ID nnnnH nnnnH, Database id nnnnH nnnnH, {Fallback |
No fallback}

```

---

#### Note:

When the checking process begins and CheckTable detects inconsistencies, CheckTable displays specific messages that pertain to those inconsistencies. CheckTable checks tables in alphabetical order by database name and table name. If you specify to check DBC, then CheckTable always checks DBC before the named database and table. CHECK AGAIN IS AN exception to alphabetical table ordering. In table recheck processing, the table order is determined by the order of the table appearance within the error file.

---

**Example: SKIPLOCKS**

CheckTable automatically skips any tables when it tries to lock them and finds that they are locked already. For example, suppose you type the following:

```
CHECK fb3.t1 at level two SKIPLOCKS IN PARALLEL;
```

If fb3.t1 is locked when you submit the command, then the following output appears:

```
Check Beginning at 07:57:25 17/12/30
Table: "FB3"."T1" skipped at 07:57:26 17/12/30 due to pending lock.
  Map No 1025, Table id 0000H 3E9H, Database id 0000H 041BH.
Summary:
  0 table(s) checked.
  0 fallback table(s) checked.
  0 non-fallback table(s) checked.
  1 table(s) skipped due to pending lock.
  0 table(s) failed the check.
  0 Dictionary error(s) were found.
```

Skiplocks applies only to non-concurrent mode lock handling.

CheckTable enforces CONCURRENT MODE with RETRY LIMIT=1. If CheckTable cannot get a lock on a table, it attempts one retry after the rest of the tables are checked.

**Example: SERIAL/PARALLEL**

Serial mode allows CheckTable to check a single table at a time. Parallel mode allows CheckTable to check multiple tables simultaneously. The following tables shows examples of both modes.

IF you want to check database DB0 with tables t1, t10, t100, t1000, and t11 at level TWO in ...	THEN type ...
SERIAL mode	CHECK db0.t1% at level two IN SERIAL;
PARALLEL mode	CHECK db0.t1% at level two IN PARALLEL;

**Example: Serial Mode Output**

Tables are checked one at a time and all table messages are grouped together under one table message.

```
Table: "DB01"."T1" starting at 17:08:06 11/04/26.
  Map No 1025, Table id 0000H 0C4EH, Database id 0000H 041BH,
  Header missing on 1 AMP:
  00000
  Further checking skipped because of missing table header(s).
  Table result: Error(s) reported.
```

When processing completes, the following termination message or summary of the check results appears:

```
Summary:
  5 table(s) checked.
  3 fallback table(s) checked.
  2 non-fallback table(s) checked.
  3 tables failed check.
  0 Dictionary error(s) were found.
Check completed at 7:57:24 17/12/30.
```

---

**Note:**

The entire output is not shown.

---

**Example: Parallel Mode Output**

Tables are concurrently checked by multiple worker tasks. Table messages display separately with no message grouping and may appear in three types of table messages: starting, checking, and ending. Consequently, these messages may display with other messages for other tables due to concurrent table checking.

The database name and the table name precede the report for each message to help you identify the table to which the output message belongs:

```
Check beginning at 07:57:14 17/12/30.
Table: "DB0"."T1" starting at 07:57:15 17/12/30.
  Map No 1025, Table id 0000H 3F92H, Database id 0000H 041BH, Fallback.
Table:"DB0"."T10" starting at 07:57:15 17/12/30.
  Map No 1025, Table id 0000H 0D14H, Database id 0000H 041BH, Fallback.
Table:"DB0"."T100" starting at 07:57:15 17/12/30.
  Map No 1025, Table id 0000H 0D6EH, Database id 0000H 041BH, Fallback.
Table:"DB0"."T1000" starting at 07:57:15 17/12/30.
  Map No 1025, Table id 0000H 12E6H, Database id 0000H 041BH, Fallback.
Table: "DB0"."T10" checking at 07:57:15 17/12/30.
  Map No 1025, Table id 0000H 0D6EH, Database id 0000H 041BH, Fallback.
    2741: Table header not found.
      Table id 0000H 0D14H
      Header missing on 1 AMP:
      00000
      Further checking skipped because of missing header(s).
Table: "DB0"."T10" ending at 07:57:15 17/12/30.
  Map No 1025, Table id 0000H 0D6EH, Database id 0000H 041BH, Fallback.
  Table Result: No errors
Table: "DB0"."T1" checking at 07:57:15 17/12/30.
```

```

Map No 1025, Table id 0000H 3F92H, Database id 0000H 041BH, Fallback.
  2757: Primary data row is missing.
    Fallback AMP 00001, Fallback subtable 2048
    Row id 0000H 8C49H CDABH 0000H 0001H
    Expected primary AMP 00000
Table: "DB0"."T111" starting at 07:57:16 17/12/30.
  Map No 1025, Table id 0000H 12E8H, Database id 0000H 041BH, Fallback.
Table: "DB0"."T1" checking at 07:57:17 17/12/30.
  Map No 1025, Table id 0000H 3F92H, Database id 0000H 041BH, Fallback.
    2757: Primary data row is missing.
      Fallback AMP 00001, Fallback subtable 2048
      Row id 0000H BD81H 0459H 0000H 0001H
      Expected primary AMP 00000
Table: "DB0"."T100" checking at 07:57:17 17/12/30.
  Map No 1025, Table id 0000H 0D6EH, Database id 0000H 041BH, Fallback.
    2757: Primary data row is missing.
      Fallback AMP 00003, Fallback subtable 2048
      Row id 0000H 1897H 9B57H 0000H 0001H
      Expected primary AMP 00002
Table: "DB0"."T1000" ending at 07:57:18 17/12/30.
  Map No 1025, Table id 0000H 12E6H, Database id 0000H 041BH, Fallback.
  Table Result: No errors
Table: "DB0"."T1" checking at 07:57:18 17/12/30.
  Map No 1025, Table id 0000H 3F92H, Database id 0000H 041BH, Fallback.
    2880: Reference index row indexes non existent data row.
      AMP 00000, Primary subtable
      Reference index id 0
      Reference index row id 0000H 8C49H CDABH 0000H 0001H
      Reference index row count exceeds data row count by 1
Table: "DB0"."T1" checking at 07:57:18 17/12/30.
  Map No 1025, Table id 0000H 3F92H, Database id 0000H 041BH, Fallback.
    2888: Invalid reference index row.
      AMP 00002, Primary subtable
      Reference index id 0
      Reference index row id 0000H 1897H 9B57H 0000H 0001H
Table: "DB0"."T100" checking at 07:57:18 17/12/30.
  Map No 1025, Table id 0000H 0D6EH, Database id 0000H 041BH, Fallback.
    2757: Primary data row is missing.
      Fallback AMP 00003, Fallback subtable 2048
      Row id 0000H 3133H 36AEH 0000H 0001H
      Expected primary AMP 00002
Table: "DB0"."T100" ending at 07:57:19 17/12/30.
  Map No 1025, Table id 0000H 0D6EH, Database id 0000H 041BH, Fallback.
  Table Result: Error(s) reported.

```

```
Table: "DB0"."T1" checking at 07:57:19 17/12/30.
Map No 1025, Table id 0000H 3F92H, Database id 0000H 041BH, Fallback.
AMP 00002, Primary subtable
Reference index id 0
Reference index row id 0000H 3133H 36AEH 0000H 0001H
```

### Example: TABLES=*n* Clause

The following example uses the TABLES=*n* clause option to specify the number of tables to check simultaneously in parallel mode.

```
check all tables at level one in parallel tables=3;
Check beginning at 10:55:23 17/01/06.
Data dictionary check started at 10:55:23 17/01/06.
F2
>>>> Status: CheckTable running in PARALLEL mode.
      3 CheckTable tasks started.
      3 CheckTable tasks ACTIVE.
      0 CheckTable tasks IDLE.
Task   STATUS
```

### Example: PRIORITY

A priority level controls resource usage and improves performance. To run CheckTable in the High timeshare workload group (H-WD), type the following:

```
CHECK db3 at level two priority=H;
```

For the PRIORITY option, the behavior depends on whether TASM is enabled.

To run CheckTable in PARALLEL mode in a workload named WAdminM, type the following:

```
CHECK eb3, db3, fb3.t1 at level two IN PARALLEL PRIORITY='WAdminM';
```

Assume the workload name specified for PRIORITY in the following example is invalid:

```
CHECK eb3, db3, fb3.t1 at level two IN PARALLEL PRIORITY='WDSdminM';
```

If TASM is not enabled, CheckTable displays an error message indicating that the priority setting is invalid, and waits for the next command.

If TASM is enabled and you specify an invalid workload name, CheckTable uses the default workload mapping, specified in the Workload Designer portlet, for example, workload WD-Default.

### Example: Concurrent mode and retry limit

Because CheckTable overrides CONCURRENT mode with Retry Limit=1, the first try and one re-try are shown for a locked table. To run CheckTable in CONCURRENT mode with Retry Limit=1:

```
check rfc66706.table_1 at level three;
```

The following appears:

```

Table: "RFC66706"."TABLE_1" starting at 16:15:32 17/02/09.
  Map No 1025, Table id 0000H 04D3H, Database id 0000H 041BH, Fallback.
  Table check skipped at 16:15:32 17/02/09 due to pending lock.

Table: "RFC66706"."TABLE_1" starting at 16:16:32 17/02/09.
  Map No 1025, Table id 0000H 04D3H, Database id 0000H 041BH, Fallback.
  Table check skipped again at 16:16:32 17/02/09 due to pending lock.
  Table Result: Skipped

```

### Example: Checking table headers

To check all table headers and all tables in DBC, type the following:

```
check dbc at level one;
```

Checking table headers requires a table read lock on DBC.TVM. For each table in DBC, CheckTable will obtain a table read lock, check the table, and release the table lock. These locks will probably block DDL operations. However, the duration of DBC check should be short.

### Example: Excluding a database from being checked

To check all tables in all databases excluding DBC, type the following:

```
check all tables exclude dbc at level one;
```

A table access lock on DBC.DBASE is obtained for a short time to get a list of databases. This access lock should cause minimal contention. For each database, a table access lock on DBC.TVM is obtained for a short time to get a list of tables in the current database. This access lock should also cause minimal contention.

For each table in the current database, CheckTable will obtain a table read lock, check the table, and release the table lock. Any operation that requires either a write lock or exclusive lock on the table being checked will be blocked. The locking duration may be longer for a large table. An EXCLUDE clause can be used to skip large tables that are actively modified to avoid blocking.

### Example: Reporting down subtables and regions (DOWN ONLY)

Level-one checks report down subtables and down regions, but do not define the region that is down.

```

check fiu.onedr at level one down only;
Check beginning.
Table: "FIU"."ONEDR" checking.
  Map No 1025, Table id 0000H 06A1H, Database id 0000H 041BH. Fallback.
    9131:Check was skipped due to detection of error <9130>.
      AMP 00000, Primary Data Subtable 1024 has down region marked.
  Table Result: Error(s) reported
Summary:
  1 table(s) checked.
  1 fallback table(s) checked.

```

```

0 non-fallback table(s) checked.
1 table(s) skipped due to presence of down regions and/or down subtables.

```

### Example: Reporting start and end of down region

Level-two and level-three checks specify the starting and ending rows that define the down region.

```

check fiu.onedr at level three down only;
Check beginning.
Table: "FIU"."ONEDR" checking.
  Map No 1025, Table id 0000H 06A1H, Database id 0000H 041BH, Fallback.
    9131:Check was skipped due to detection of error <9130>.
      AMP 00000, Primary Data Subtable 1024 has down region marked.
        Region #1
          Start Row Id 0000000000000000 CDC4H F1BBH 0000H 0001H
          End Row Id   0000000000000000 DA53H B54CH 0000H 0000H
    Table Result: Error(s) reported
  Summary:
    1 table(s) checked.
    1 fallback table(s) checked.
    0 non-fallback table(s) checked.
    1 table(s) skipped due to presence of down regions and/or down subtables.

```

### Example: Marking AMPs as down due to excessive down regions

If the number of down regions in a subtable exceeds the threshold defined by the DBS Control setting MaxDownRegions, the subtable is marked down on all AMPs, and CheckTable reports the subtable down. For the following example, assume the subtable on AMP 0 had a number of down regions exceeding the limit set by MaxDownRegions.

```

check fiu.onedt at level two down only;
Check beginning.
Table: "FIU"."ONEDT" checking.
  Map No 1025, Table id 0000H 06A2H, Database id 0000H 041BH, Fallback.
    9131:Check was skipped due to detection of error <9129>.
      AMP 00001, Primary Data Subtable 1024 is marked down.
      AMP 00000, Primary Data Subtable 1024 is marked down.
    Table Result: Error(s) reported
  Summary:
    1 table(s) checked.
    1 fallback table(s) checked.
    0 non-fallback table(s) checked.
    1 table(s) skipped due to presence of down regions and/or down subtables.

```



**Example: Rechecking failed tables from previous CHECK (CHECK AGAIN)**

To recheck the failed tables from the previous CHECK command:

```
CHECK AGAIN AT LEVEL TWO;
```

Before re-running a Level 2 check, you must specify Level 2 again.

You must specify the check level and CHECK options to use the table recheck process on the CHECK AGAIN command line. For example, you could run the table recheck at a higher level.

**Example: Rechecking failed tables from a specific error file**

To recheck the failed tables in a specific error file:

```
CHECK AGAIN CheckTableErrors20130503164640 AT LEVEL TWO;
CHECK AGAIN /var/tmp/CheckTableErrors20130503164640 AT LEVEL TWO;
CHECK AGAIN '/var/tmp/CheckTableErrors20130503164640' AT LEVEL TWO;
CHECK AGAIN "/var/tmp/CheckTableErrors20130503164640" AT LEVEL TWO;
```

**Example: Dictionary check only**

To initiate only a dictionary check:

```
check all tables at level one in parallel scope dictionary;
```

```
Running
```

```
Reading
```

```
Check beginning at 12:16:41 20/04/20.
```

```
Checktable will run from Host: 0 Session: 0,61128
```

```
System Map check started at 12:16:41 20/04/20.
```

```
No errors reported for System Map check.
```

```
Data dictionary check started at 12:16:41 20/04/20.
```

```
No errors reported for Data Dictionary check.
```

```
Summary:
```

```
0 table(s) checked.
```

```
0 fallback table(s) checked.
```

```
0 non-fallback table(s) checked.
```

```

0 table(s) failed the check.
0 Dictionary error(s) were found.
0 System Map error(s) were found.

```

```
Check completed at 12:16:43 20/04/20.
```

### Example: User table check only

To initiate only a user table check:

```
check all tables at level one in parallel scope user;
```

```
Running
```

```
Check beginning at 12:16:18 20/04/20.
```

```
Reading
```

```
Checktable will run from Host: 0 Session: 0,55914
```

```
Checktable will run from Host: 0 Session: 0,55915
```

```
Checktable will run from Host: 0 Session: 0,55916
```

```
Checktable will run from Host: 0 Session: 0,55917
```

```
Checktable will run from Host: 0 Session: 0,55918
```

```
Checktable will run from Host: 0 Session: 0,55919
```

```
Table: "DBC"."ACCLOGRULETBL" starting at 12:16:19 20/04/20.
```

```
Map No 2, Table id 0000H 0159H, Database id 0000H 0001H, Fallback.
```

```
Table: "DBC"."ACCESSRIGHTS" starting at 12:16:19 20/04/20.
```

```
Map No 2, Table id 0000H 0131H, Database id 0000H 0001H, Fallback.
```

```
...
```

```
Table: "TD_SYSEXML"."XSLT_SHRED_BATCH" ending at 12:16:26 20/04/20.
```

```
Map No 2, Table id 0000H 08F0H, Database id 0000H 03F2H, Fallback.
```

```
Table Result: No errors
```

```
Summary:
```

```
1,193 table(s) checked.
```

```
1,174 fallback table(s) checked.
```

```
19 non-fallback table(s) checked.
```

```
4 no primary-index table(s) checked.
```

16 table(s) bypassed due to being unhashed.

0 table(s) failed the check.

Check completed at 12:16:26 20/04/20.

### **Example: Dictionary and user table check**

To initiate a dictionary and user table check:

Check all tables at level one in parallel scope all;

Reading

Check beginning at 12:19:34 20/04/20.

Checktable will run from Host: 0 Session: 0,63954

System Map check started at 12:19:34 20/04/20.

No errors reported for System Map check.

Data dictionary check started at 12:19:34 20/04/20.

No errors reported for Data Dictionary check.

Checktable will run from Host: 0 Session: 1,1244

Checktable will run from Host: 0 Session: 1,1245

Checktable will run from Host: 0 Session: 1,1246

Checktable will run from Host: 0 Session: 1,1247

Checktable will run from Host: 0 Session: 1,1248

Table: "DBC"."ACCLOGRULETBL" starting at 12:19:36 20/04/20.

Map No 2, Table id 0000H 0159H, Database id 0000H 0001H, Fallback.

Table: "DBC"."ACCLOGTBL\_TD15" starting at 12:19:36 20/04/20.

Map No 2, Table id 0000H 0146H, Database id 0000H 0001H, Fallback.

...

Table: "TD\_SYSEXML"."XSLT\_SHRED\_BATCH" ending at 12:19:43 20/04/20.

Map No 2, Table id 0000H 08F0H, Database id 0000H 03F2H, Fallback.

Table Result: No errors

**Summary:**

1,193 table(s) checked.  
1,174 fallback table(s) checked.  
19 non-fallback table(s) checked.  
4 no primary-index table(s) checked.

16 table(s) bypassed due to being unhashed.

0 table(s) failed the check.  
0 Dictionary error(s) were found.  
0 System Map error(s) were found.

Check completed at 12:19:43 20/04/20.

## CHECKTABLEB

The CHECKTABLEB command runs the non-interactive batch mode of CheckTable. CHECKTABLEB cannot be aborted. It runs until the check process has completed.

To run CHECKTABLEB in batch mode, use the cnsrun utility. For information on cnsrun, see [CNS Run \(cnsrun\)](#).

### Syntax

The syntax of CHECKTABLEB is identical to the syntax of the [CHECK](#) command.

### Example

To run CHECKTABLEB:

```
cnsrun -utility checktableb -commands "{check dbc at level one;} {quit;}"
```

## OUTPUT

The OUTPUT command is used to display or to change the output destination. OUTPUT creates a new log file, overwrites an existing log file, appends an existing log file, and displays output to the screen.

### Syntax

```
{ OUTPUT | OUT }
  [ [ ECHO ] { TO | OVER | INTO } { file | STDOUT | ME } ] ;
```

### Syntax Elements

#### OUTPUT

##### OUT

If OUTPUT or OUT is specified by itself, the current output destination is displayed.

#### ECHO

If specified, output is written to both the console and file.

If not specified, the default is to redirect output to the file.

#### TO

Specifies that a new file should be created. If the file already exists, then an error is returned.

#### INTO

Appends output to an existing file.

If the file does not exist, it will be created.

#### OVER

Overwrites an existing file with new output.

If the file does not exist, it will be created.

#### *file*

The fully qualified or a relative path filename.

A fully qualified filename starts with '/' (for example, /var/tmp/checklog).

If a relative path filename is specified (such as checklog or checktable/checklog), then it is created under the default directory, which is the PDE Temp directory, /var/opt/teradata/tdtemp.

If a directory is specified in the file path, then that directory must exist or an error occurs, and the directory is not be created.

**STDOUT**

Default file to which output is written.

**ME**

Synonym for STDOUT.

**Usage Notes**

- Specifying OUTPUT/OUT by itself displays the current output destination. By default, the output destination is the console (STDOUT), so you must explicitly specify the OUTPUT command to create a log file.

By sending output to both the console and log file, the ECHO option allows users to monitor progress of the CHECK command on the console and generates a log file for reference purposes. This is especially helpful if the CHECK command runs for a long time or generates a great deal of output. Use the ECHO option to see table failures in the log file or to review output for a particular table.

Because the log file provides a complete record of the CHECK command run, it can be helpful to browse through the log file using a text editor.

The log file remains open until the CheckTable session is closed or until the next OUTPUT command is issued. Command output is written to the log file as long as the file is open. The log file can contain multiple command outputs, so a copy of the command line is written to the log file to distinguish one output from another.

If output is redirected to a file, some commands may still output to the console. This occurs for commands that are more user-interactive and need to display output to the user. Commands like HELP, STATUS, OUTPUT, and interactive help (F7) fall into this category. The STATUS and interactive help commands will only display output to the console. The HELP and OUTPUT commands will display output to both the console and log file.

CheckTable does not manage the log files. The files remain until deleted by the user.

- The CheckTable log file is located on the node that runs the control AMP. To determine which node is running the control AMP, use the STATUS command of the Vproc Manager utility. The control AMP is designated under the Logical Configuration with an asterisk. For more information, see [CheckTable \(checktable\)](#).
- If a file system error occurs during a log file write operation, an error message is written on the console and log file processing is halted. The current output destination reverts automatically back to the default (STDOUT).
- When running multiple instances of the CheckTable utility on the same system, do not use a single log file for both CheckTable instances.

**Example: Displaying current output destination**

To display the current output destination:

```
OUTPUT;
```

**Example: Redirecting output to a file**

To redirect output to a new file using a relative path filename:

```
OUT TO checklog;
```

**Example: Echoing output to the console and a file**

To echo output to the console and overwrite a file using a fully qualified filename:

```
OUTPUT ECHO OVER /var/tmp/checklog;
```

**Example: Redirecting output and appending to a file**

To redirect output and append to an existing log file:

```
OUT INTO checklog;
```

**Example: Specifying an irregular output file name**

To specify an irregular file name:

```
OUT TO 'check log';
```

**Example: Resetting output destination to console screen**

To reset output back to the console:

```
OUT TO STDOUT;
```



## ERRORFILEDIR

Displays or changes the location of the error file for the current CheckTable session.

### Syntax

```
ERRORFILEDIR [ Directory-name ] ;
```

### Syntax Elements

#### *Directory-name*

The name of the directory location of the error file.

### Usage Notes

If ERRORFILEDIR is specified by itself, then the current error file location displays.

The default error file location is the PDE Temp directory.

To determine the PDE Temp directory path, enter `pdepath -S` on the Linux command line.

If *Directory-name* is specified, the error file location for the current CheckTable instance is set to this path.

The *Directory-name* can be a fully qualified directory name (for example, “/var/tmp/checktable”) or a relative path directory name (such as “checktable” or “checktable/errors”). For a relative name, the root directory is the PDE Temp directory.

### Example: Displaying the current error file location

To display the current error file location:

```
ERRORFILEDIR;
```

### Example: Changing the error file location

To change the error file location:

```
ERRORFILEDIR CheckErrorFiles;
ERRORFILEDIR /var/tmp/CheckErrorFiles;
```

## HELP

Displays CheckTable online help.

### Syntax

```
HELP ;
```

### Usage Notes

If you are using Database Window, the shortcut key F7 can be used for interactive help, where function keys are used to navigate within the help system.

F7 only invokes the interactive help in interactive mode.

## QUIT

Terminates the CheckTable utility.

### Syntax

```
QUIT ;
```

## STATUS

Displays the status of the currently running CHECK command.

### Syntax

```
STATUS ;
```

### Usage Notes

The STATUS command is only supported in interactive mode.

If you are using Database Window, the shortcut key F2 is equivalent to the STATUS command.

## ABORT

Aborts the currently running CHECK command.

### Syntax

```
ABORT ;
```

### Usage Notes

The ABORT command is only supported in interactive mode.

If you are using Database Window, the shortcut key F4 is equivalent to the ABORT command.

## ABORT TABLE

Aborts the check on the table currently being checked.

### Syntax

```
ABORT TABLE ;
```

### Usage Notes

The ABORT TABLE command is only supported in interactive mode.

Use the ABORT TABLE command to abort the check of the current single table in Serial mode or the current group of tables in Parallel mode. CheckTable continues with the next table.

If you are using Database Window, the shortcut key F3 is equivalent to the ABORT TABLE command.

## CNS Run (cnsrun)

---

### Note:

The utility described is not available for as-a-service customers.

---

The CNS Run utility, `cnsrun`, provides the ability to run a database utility from a script. It is suitable for running other utilities as pre-scheduled tasks or cron jobs.

### Runs From

CNS Run is intended to be run from operating system scripts, which can be run either from the command line or by daemons, such as cron.

For general information on starting the utilities from different interfaces, see [Starting the Utilities](#).

### Syntax

```
cnsrun {
  {
    { -utility uname | -join uname -screen number }
    { -file file_name | -commands clist }
    [ -machine host_name ]
    [ -force ]
    [ -ok ]
    [ -multi ]
    [ -nostop ]
    [ -prompt string ]
    [ -tool cns_tool_path ] |

    -query |

  } [ -debug n | -output ] [ -log file ] |

  -help
}
```

### Syntax Elements

#### **-utility *uname***

Specifies the utility program to start.

**-join *uname***

Specifies the name of an already running database utility.

**-screen *number***

Specifies the screen number where a utility that is to be joined is already running. -screen must be used only with the -join option.

**-file *file\_name***

Specifies a file that contains input commands to send to the utility program.

**-commands *clist***

Specifies the commands to send to the utility.

Each command must be enclosed within braces { }, and there must be one or more spaces between commands.

---

**Note:**

If there is more than one command, enclose *clist* in double quotation marks.

---

**-machine *host\_name***

The host name associated with the Control Node for the TPA instance on which to run the utility program. The Control Node is the node running with the lowest Node ID number. The Control Node can move between TPA restarts. To determine the name of the Control Node, type `cnsdim -host` at the command prompt.

The default is localhost, which only works on SMP systems or when the program is actually running on the Control Node of an MPP system.

**-force**

This utility is to be started even if the database software is not running.

**-ok**

This utility is to be started even if logons are disabled.

In the absence of this option and the -force option, the utility is not started if logons are not enabled.

**-multi**

Permits more than one instance of the specified utility program to run simultaneously. Without the -multi option, cnsrun fails to start a utility if an instance of the utility is already running.



**-nostop**

Considers the operation successful even if the utility continues to run after all input is exhausted.

**-prompt *string***

When running a database utility that accepts input after a text prompt, this switch is required to enable cnsrun to know when to send input.

String is a regular expression that allows cnsrun to recognize an input prompt from the utility. For more information, see [Regular Expressions for the -prompt Option](#).

**-tool *cns\_tool\_path***

The full path to the cnstool program.

**-debug *n***

Specifies the level of debug output that will be shown on screen.

A value of 0 (the default) shows no output if the utility runs successfully.

A value of 1 shows the utility program output and its state changes.

Other larger values of *n* produce additional debugging output. All output, including error messages, appears in STDOUT, the standard console output.

**-output**

The utility output should be directed to STDOUT, the standard console output.

**Note:**

The -output option is effective only when the -debug option is set to zero.

**-log *logfile***

Produces a log of the utility run to the specified file.

**-query**

Displays the names of the utilities currently running in each of the screens, after which cnsrun exits.

**-help**

Displays information on how to use cnsrun.

## Usage Notes

This program runs the utility specified by the `-utility` option passing it the input provided by either the `-file` or `-commands` option. The command is run by this program connecting to CNS on the node specified by the `-machine` option and starting the utility from the Supervisor subwindow of Database Window (DBW). When the utility issues a read, the commands specified are sent to it until all of the commands are exhausted or until the program exits.

To use this command, the user running `cnsrun` must have access permission to run a DBW session on the machine. To set up a user to have CNS access, use the `GET PERMISSIONS`, `GRANT`, and `REVOKE` commands in DBW. For more information about the DBW Supervisor Commands, see [Database Window \(xdbw\)](#).

An error occurs if all of the interactive partitions are in use.

By default, this command produces no output unless an error occurs.

When running the Ferret utility with CNS Run, you must specify the Ferret command

```
enable script
```

as the first command. This enables the script mode of Ferret, which allows scripts that run Ferret to run uninterrupted. For more information, see [Ferret Utility \(ferret\)](#).

## Regular Expressions for the -prompt Option

Some database utilities prompt for input or commands. While those commands can be included in the command list, `cnsrun` needs to be able to recognize the utility prompts to know when to send the response. The `-prompt` option accepts regular expressions that represent utility program text prompts, to which `cnsrun` will respond by sending the next command in the command list.

The regular expression syntax follows that used in the Tool Command Language (Tcl), a subset of which is described below. For the full documentation on Tcl regular expressions, see <http://wiki.tcl.tk/396>.

Regular Expression Character	Meaning
.	A period in the regular expression matches any literal character at that position in the prompt string, to ensure that the prompt matches the regular expression.
?	A question mark indicates the preceding character or group of characters in the regular expression can occur zero or one time at that position in the prompt string, to ensure that the prompt matches the regular expression. Example: "fee?d" would match either "feed" or "fed" in the prompt string.
*	An asterisk indicates the preceding character or group of characters in the regular expression can occur zero or more times in succession at that position in the prompt string, to ensure that the prompt matches the regular expression. This means that the pattern may or may not occur at that position in the prompt string. Example: "fe*d" would match fed, feed, feeed, and would match fd, at the corresponding position in the prompt string.

Regular Expression Character	Meaning
+	A plus character indicates the preceding character or group of characters in the regular expression can occur one or more times in succession at that position in the prompt string, to ensure that the prompt matches the regular expression. Example: "fe+d" would match fed, feed, feeed, but would not match fd.
^	A circumflex character in the regular expression indicates the character or characters that follow the circumflex in the regular expression must occur at the beginning of the prompt string, to ensure that the prompt matches the regular expression. Example: "^E" requires that the prompt string begin with an upper case E so that it matches the regular expression.
\$	A dollar sign character in the regular expression indicates the character or characters that precede the dollar sign in the regular expression must occur at the end of the prompt string, to ensure that the prompt matches the regular expression. Example: ":\$" requires that the prompt string ends with a colon so that it matches the regular expression.
()	Parentheses enclose a group of characters that is to be treated as a single regular expression entity. Example: "(ABC)+" would match one or more successive occurrences of ABC at the corresponding position in a prompt string.
[]	Square braces enclose a subset of characters to match. Example: "[0-9]" would match any single digit at the corresponding position in a prompt string.

## Example: Using cnsrun to run a utility

To run the utility updatespace with the input:

```
update space for all databases;
```

```
quit
```

connecting to CNS on an SMP system, use the following command:

```
cnsrun -utility updatespace -commands "{update space for all databases;} {quit}"
```

# Control GDO Editor (ctl)

---

**Note:**

The utility described is not available for as-a-service customers.

---

The Control GDO Editor, `ctl`, also called the PDE Control program, lets you display and modify PDE configuration settings. These settings affect how the PDE handles startup, responds to resets, and the functions during the normal operation of Teradata.

Globally distributed objects (GDOs) are binary files that store Vantage configuration settings. They are distributed to and used by every node in the system. The PDE layer of Vantage ensures that the GDO is consistent across all virtual processors.

## Runs From

Ctl runs from the Linux command line.

Starting `ctl` from the command line invokes a command shell, recognized by the `ctl` command prompt, a right angle bracket (`>`).

For general information on starting the utilities from different interfaces, see [Starting the Utilities](#).

## Syntax

To start `ctl` from the command line, use the following command syntax:

```
ctl { command_spec [...] | -help }
```

### *command\_spec*

```
{ -first | -last } { command [;...] | "command [;...]" }
```

## Syntax Elements

### **-help**

Provides information on `ctl` command-line options.

### **-first**

First command or commands to run before other processing.

No default.

Example:

```
ctl -first "Node Logging Rate = 600;screen rss; write; quit"
```

Sets the RSS screen Node Logging Rate field to 600, displays the RSS screen to confirm the changes, writes the changes to the Control GDO, and quits the ctl utility.

### **-last**

Last command or commands to run just before exiting.

No default.

Example:

```
ctl -last screen
```

Displays the current screen before exiting the ctl interactive mode.

### ***command***

Command from [Ctl Commands](#).

If any *command* includes blank spaces, enclose the command list in double quotation marks.

---

### **Note:**

Teradata recommends using the full field names, rather than the alphanumeric field identifiers in command lists for the -first and -last options.

---

## **Ctl Commands**

This topic omits some internal and rarely-needed options. For information about those options, see the ctl command-line help.

## EXIT

The EXIT command exits ctl. This is synonymous with the QUIT command.

### Syntax

```
{ EXIT | EX } [ WRITE ]
```

### Usage Notes

If the `-last` option was used to specify a command when ctl was started, that command will be executed before ctl exits.

If you modified values in ctl and specify the `write` option, ctl will write the changes to the GDO before exiting.

If you modified values in ctl and do not specify the `write` option, ctl will ask if you want the changes written before ctl exits.

## HARDWARE

The **HARDWARE** command displays a screen containing the PDE hardware configuration information for this system. This information is recomputed every time the database is restarted, so it reflects the actual running configuration. The information is read only and cannot be directly modified by the user.

### Syntax

```
{ HARDWARE | HA }
```

### Example: **HARDWARE** command output

The following example shows sample output from the hardware command.

Node			min	max	avg		min	max	avg
#nodes	1	cpus/node	1	1	1	nodes/clique	1	1	1
		mips/cpu	1189	1189	1189	arrays/clique	0	0	0
		segmem/vproc(mb)	1080	1080	1080				
VPROCS									
Type	Name	##							
1	AMP	4	vprocs/node	4	4	4	kmem/vproc	413696	413696
2	PE	2	vprocs/node	2	2	2	kmem/vproc	0	0
3	GTW	1	vprocs/node	1	1	1	kmem/vproc	0	0
4	RSG	0	vprocs/node	0	0	0	kmem/vproc	0	0
5	TVS	2	vprocs/node	2	2	2	kmem/vproc	0	0
6		0	vprocs/node	0	0	0	kmem/vproc	0	0
7		0	vprocs/node	0	0	0	kmem/vproc	0	0
8		0	vprocs/node	0	0	0	kmem/vproc	0	0
9		0	vprocs/node	0	0	0	kmem/vproc	0	0
10		0	vprocs/node	0	0	0	kmem/vproc	0	0
11		0	vprocs/node	0	0	0	kmem/vproc	0	0
12		0	vprocs/node	0	0	0	kmem/vproc	0	0
13		0	vprocs/node	0	0	0	kmem/vproc	0	0
14		0	vprocs/node	0	0	0	kmem/vproc	0	0
15		0	vprocs/node	0	0	0	kmem/vproc	0	0
ARRAYS:									
Rand. Reads/Sec			Rand. Writes/Sec						
4k:	1120	1120	1120	4k:	740	740	740		
8k:	1020	1020	1020	8k:	650	650	650		
16k:	920	920	920	16k:	560	560	560		
32k:	775	775	775	32k:	445	445	445		
64k:	540	540	540	64k:	290	290	290		

128k:	325	325	325	128k:	170	170	170
256k:	160	160	160	256k:	80	80	80
512k:	80	80	80	512k:	40	40	40
768k:	50	50	50	768k:	20	20	20
1024k:	40	40	40	1024k:	20	20	20

## NETWORK:

Monocast rates: Opers/Node/second

0 b:	200106	200106	200106
100 b:	181510	181510	181510
400 b:	140660	140660	140660
1600 b:	74022	74022	74022
6400 b:	25569	25569	25569
25600 b:	7066	7066	7066

Multicast rates: Opers/Node/second

0 b:	200106	200106	200106
100 b:	181510	181510	181510
400 b:	140660	140660	140660
1600 b:	74022	74022	74022
6400 b:	25569	25569	25569
25600 b:	7066	7066	7066



## HELP

The HELP command displays the online help screen.

### Syntax

```
{ HELP | HE } [ cmd | TOPICS ]
```

### Syntax Elements

#### *cmd*

[Optional] Specify the name of the command for which you want help.

Default behavior: Command shows general help information or a ctl command syntax summary.

#### TOPICS

[Optional] Lists all help topics.

## PRINT *group*

The PRINT *group* command displays groups of variables by variable types.

### Syntax

```
{ PRINT | PR } [
  ALL |
  BUTTONS |
  CHECKBOX |
  LABELS |
  SCALES |
  TEXT |
  TRACE
]
```

### Syntax Elements

#### ALL

All variables in the PDE Control GDO. This is the default.

#### BUTTONS

Variables that take one value from a set of two or more possible values.

#### CHECKBOX

Variables that take binary values.

#### LABELS

Variables that are labels (not normally modified by users).

#### SCALES

Variables that are integer values.

#### TEXT

Variables entered as strings, which may include some numeric fields.

#### TRACE

All trace entries.

## PRINT *variable*

The PRINT *variable* command displays the value of specified variables.

### Syntax

```
{ PRINT | PR } variable [;...]
```

### Syntax Elements

#### *variable*

Variable whose value is to be displayed.

The value is displayed in the form of an assignment command for that variable.

Some variables that can be changed with ctl do not appear on any of the ctl output screens.

To see a listing of all variable names, use the print all command.

### Example: Displaying the values of specified ctl fields

For example, if you type the following two variables

```
> print minimum node action;fsg cache percent
```

then the following might be displayed, depending on your system:

```
Minimum Node Action=Clique-Down  
FSG cache Percent=80
```

## QUIT

The QUIT command exits ctl. This is synonymous with the EXIT command.

### Syntax

```
{ QUIT | QU } [ WRITE ]
```

### Usage Notes

If the `-last` option was used to specify a command when `ctl` was started, that command will be executed before `ctl` quits.

If you modified values in `ctl` and specify the `write` option, `ctl` will write the changes to the GDO before quitting.

If you modified values in `ctl` and do not specify the `write` option, `ctl` will ask if you want the changes written before `ctl` exits.

## READ

The READ command re-reads the control GDO, resetting any changes.

### Syntax

```
{ READ | RE }
```

## SCREEN

The SCREEN command displays a screen of PDE configuration information from the PDE Control GDO. Some of this information can be modified using the [variable = setting](#) and [WRITE](#) commands.

### Syntax

```
{ SCREEN | SC } [ DBS | DEBUG | RSS | VERSION ]
```

Entering the screen command alone, without a screen name, redisplayes the current screen (the one that was most recently displayed).

### Syntax Elements

#### DBS

Displays parameters that control how the database software responds to unusual conditions.

The DBS screen contains the following control fields:

Setting	Description
Minimum Node Action	<p>Determines what action to perform when a clique contains fewer than the Minimum Nodes Per Clique field.</p> <ul style="list-style-type: none"> <li>• Clique Down means the clique is not started, and the vprocs associated with the clique are marked OFFLINE. This is the default.</li> <li>• DBS Down means the database will not start. All the vprocs are marked OFFLINE.</li> </ul>
Minimum Nodes Per Clique	<p>Specifies the number of nodes required for a clique to operate. (Inactive hot standby nodes are not considered.) If a clique contains fewer than this number of nodes when the database is started, the Minimum Node Action setting determines what action to take.</p> <p><b>Note:</b></p> <p>Minimum Nodes Per Clique does not affect cliques in which all nodes are running properly, including “single-node cliques” such as AMP-less Channel nodes. These cliques are exempt from the Minimum Node Action, regardless of the Minimum Nodes Per Clique setting.</p> <p>Changes to this value do not take effect until the database is restarted. When the database starts, the system determines the minimum number of nodes required for each clique, based on the number of vprocs, nodes, and available memory in the clique. For systems containing cliques of different sizes, Teradata determines a minimum node requirement for each clique, then considers the largest value as the minimum node requirement for all cliques in the system.</p> <ul style="list-style-type: none"> <li>• If Minimum Nodes Per Clique is manually set to a value less than the system-determined minimum, the manually set value is replaced with the system value, and the action is noted in the system log.</li> <li>• If Minimum Nodes Per Clique is manually set to a value greater than the system-determined minimum, the manual setting takes precedence.</li> </ul>

Setting	Description
	<ul style="list-style-type: none"> <li>If Minimum Nodes Per Clique is manually set to a number greater than the total number of nodes in any clique, all nodes in that clique must be fully functional for the clique to be used.</li> </ul> <p>Choosing a value for Minimum Nodes Per Clique involves a trade off between performance and system availability. When one or more nodes in a clique fail, the AMPs assigned to the failed nodes migrate to the remaining nodes in the clique. System performance can degrade when some nodes handle more vprocs than other nodes.</p> <p>Setting a Minimum Nodes Per Clique value allows you to define at what point it is more efficient for the system to consider a partially disabled clique to be entirely unavailable, allowing the Teradata fallback logic to compensate for the problem. Note that running a system with fallback limits some functions, which should be a consideration when choosing an appropriate value for this setting.</p>
FSG Cache Percent	<p>Specifies the percentage of available memory that can be used for the database file segment cache. The available memory is the memory that remains beyond the memory required to run utilities and the Vantage programs.</p> <p>Changes to this value do not take effect until the database is restarted.</p> <p><b>Note:</b></p> <p>Setting the value to zero resets FSG Cache Percent to the Teradata default, which can differ, depending on your system configuration. You cannot disable this setting.</p>
TIM FSG Cache Percent	<p>Specifies the size of the hot cylinder cache as a percentage of FSG cache.</p> <p><b>Note:</b></p> <p>This field is visible only on platforms that support the Teradata Intelligent Memory feature.</p>
Clique Failure	<p>Determines what to do when a clique is down.</p> <ul style="list-style-type: none"> <li>Clique Down means the database will attempt to start without the down clique. This is the default.</li> <li>DBS Down means the database will not be started if a clique is down.</li> </ul>
Cylinder Read	<p>Allows full-table scan operations to run more efficiently by reading the majority of data blocks on a cylinder with a minimal number of I/O operations.</p> <p>A sector is the smallest unit of addressable disk storage. In the Teradata file system, a fixed number of sectors are grouped together to form a logical cylinder. Each cylinder contains several data blocks. A data block stores rows from a single database table, and spans several contiguous sectors within the cylinder. A single data block is the smallest unit of I/O for the Teradata file system. Every data block that is read incurs a disk I/O operation.</p> <p>The majority of data blocks on a single cylinder usually store rows from a single table. Cylinder Read allows the file system to read multiple data blocks from a cylinder with a single I/O. For database operations that require scanning a complete table, this is a more efficient way to read data than performing a separate I/O for every data block of the table.</p>

Setting	Description
	<ul style="list-style-type: none"> <li>• On means Cylinder Read is enabled. The system reads all the data blocks of a table on a cylinder with a minimal number of I/O operations, rather than using a separate I/O operation for reading each data block.</li> <li>• Off means Cylinder Read is disabled. Data blocks are read individually. The file system performs an I/O operation once for every data block that is read.</li> <li>• LogOnly means Cylinder Read is used only for scans of the WAL log.</li> </ul>
Restart After DAPowerFail	<p>Allows you to select whether or not to restart the database after a disk array AC power failure.</p> <ul style="list-style-type: none"> <li>• On means the database is restarted after a disk array AC power failure. This is the default.</li> <li>• Off means the database is not restarted after a disk array AC power failure.</li> </ul>
Cylinder Read Ageing Threshold	<p>Specifies the maximum amount of FSG cache (in MB) that is used for segments loaded by Cylinder Read. A value of 0 represents the Teradata default, which is 96 MB for systems that use large cylinders or 16 MB for systems that use small cylinders.</p>
Maximum Fatal AMPs	<p>Specifies the maximum number of FATAL AMPs at or above which the database will not restart.</p> <p>The default value is 0, which allows the database to restart even with FATAL AMPs. This is intended mainly for systems that have fallback protection. Even for systems without fallback protection, some queries can be completed when AMPs are offline if the queries do not access rows that are on the FATAL AMP.</p> <p>For sites with critical tables that do not use fallback protection, this can be set to 1 to prevent the database from restarting until the problem that caused the FATAL AMP is resolved. For additional protection against running with FATAL AMPs, when this setting is 1, set Minimum Node Action and Clique Failure to DBS-DOWN.</p>

## DEBUG

Used for internal debugging of PDE and the database.

### Note:

Do not modify the debug values unless you are explicitly directed to do so by Teradata Support Center Personnel.

The Debug screen contains the following control fields:

Setting	Description
Start DBS	<p>Determines whether the database is started automatically when PDE starts.</p> <ul style="list-style-type: none"> <li>• Off means the database is not started automatically when PDE is started.</li> </ul>



Setting	Description
	<ul style="list-style-type: none"> <li>On means the database is started automatically when PDE is started. This is the default.</li> </ul>
Break Stop	<p>Controls whether the database restarts automatically or stops for the system debugger when a fatal error occurs.</p> <ul style="list-style-type: none"> <li>Off means the database restarts automatically after performing a dump. This is the default.</li> <li>On means the database stops and waits for the system debugger to be attached so that the problem can be diagnosed interactively.</li> </ul>
Start With Logons	<p>Controls whether logons are enabled or not. The following values apply:</p> <ul style="list-style-type: none"> <li>None means none of the users are allowed to log on to the system.</li> <li>All means all of the users are allowed to log on to the system.</li> <li>DBC means only new DBC users are allowed to log on to the system.</li> <li>LocalDBC means only the local DBC user is allowed to log on to the system.</li> </ul> <p><b>Note:</b> Changes to this setting take effect after the next database restart. Start With Logons commands issued from the Supervisor window of Database Window have the same effect, but do not require a database restart. For more information, see <a href="#">Database Window (xdbw)</a>.</p>
Start with Debug	<p>Halts the database software startup until after the system debugger has been attached.</p> <ul style="list-style-type: none"> <li>Off means the database is started normally. This is the default.</li> <li>On means the database will not run until the system debugger is attached and used to continue operations.</li> </ul>
Save Dumps	<p>Specifies whether database dumps are to be loaded into the database. The following values apply:</p> <ul style="list-style-type: none"> <li>Off means database dumps are not loaded into the database. This is the default.</li> <li>On means database dumps are loaded into the database.</li> </ul>
Snapshot Crash	<p>Specifies whether the database continues to run after a snapshot dump.</p> <ul style="list-style-type: none"> <li>Off means the database continues to run after a snapshot dump. This is the default.</li> <li>On means a snapshot dump is accompanied by a full database restart. If Break Stop is also On, the system stops for debugging.</li> </ul>
Maximum Dumps	<p>Applies to a per-node basis and is meaningful only for database dumps. Controls the maximum number of crash dumps that will be saved. A value of -1 means the system will save as many dumps as can fit on the disk containing the dump directory. The default is 5. Setting this field to 0 disables database dumps.</p>
Start PrgTraces	<p>Allows you to save or not save prgtraces to files.</p> <ul style="list-style-type: none"> <li>Off means prgtraces are not saved to files. This is the default.</li> </ul>

Setting	Description
	<ul style="list-style-type: none"> <li>On means prgtraces are saved to files.</li> </ul>
Restart Dump Type	<p>Specifies the type of memory dump to be performed for fatal database errors. There are two types:</p> <ul style="list-style-type: none"> <li>System Dumps all PDE processes on all system nodes.</li> <li>Selective: Dumps only those processes likely to be relevant to diagnosing the problem that caused the error. Selective dumps are smaller, and can be collected more quickly. This provides faster database restarts.</li> </ul>
UDF Debugging	<p>Permits debugging sessions in Vantage. You must use a debugger supplied by Teradata to debug UDFs and external stored procedures running in Vantage.</p> <p>Teradata offers command-line and Eclipse (Studio) plug-in debuggers for C/C++ and Java. For more information on these debuggers, see <i>Teradata Vantage™ - SQL External Routine Programming</i>, B035-1147, <a href="#">Teradata Debugger for C/C++ UDF</a>, and <a href="#">Teradata Debugger for Java UDF</a>.</p> <p>The default is Off.</p>

## RSS

Teradata resource usage (ResUsage) statistics are collected by the Resource Sampling Subsystem (RSS). These statistics are logged to special tables in the database.

The RSS screen allows you to specify the rates of ResUsage data logging.

The RSS screen contains the following control fields:

Settings	Description
Node Logging Rate	The interval (in seconds) between writing statistics to the database. The default is 600.
RSS Table Logging Enable	Controls whether logging is enabled to the various ResUsage tables. Only the SPMA table has logging enabled by default.
RSS Summary Mode Enable	<p>When logging is enabled for certain ResUsage tables, multiple rows of resource usage data are written during each logging period. Summary mode reduces the amount of data collected per logging period by causing the RSS to store a single summary row per type per node, instead of one row per logging entity.</p> <p>For example, if regular logging is enabled for the SCPU table, separate rows storing statistics for every CPU are written during each logging period. If summary mode is enabled, only a single row is written for each node, regardless of the number of CPUs in that node. The single row includes summary data for all node CPUs.</p>

Settings	Description
	<p>Similarly, if regular logging is enabled for the SVPR table, separate rows are written for every individual vproc. If summary mode is enabled for this table, one row is written for each vproc type (AMP, PE, and others).</p> <p><b>Note:</b> RSS Summary Mode is effective for a table only if RSS Table Logging is also enabled.</p>

**Note:**

- To run ResUsage reports, you must select the logging rate and enable the appropriate tables for logging.
- Fields without alphanumeric identifiers should not be modified.
- RSS aligns the logging periods to the clock on the top of every hour. Therefore, these values must divide evenly into 3600 seconds (one hour). The following set of values are legal for RSS logging rates: 0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, 16, 18, 20, 24, 25, 30, 36, 40, 45, 48, 50, 60, 72, 75, 80, 90, 100, 120, 144, 150, 180, 200, 225, 240, 300, 360, 400, 450, 600, 720, 900, 1200, 1800, 3600.
- For more information on the ResUsage tables and the types of information each table stores, see *Teradata Vantage™ - Resource Usage Macros and Tables*, B035-1099.

**VERSION**

The Version screen displays the version numbers of the running PDE and Database System software. Input fields on this screen let you specify different versions of the software to be installed and run the next time the system is restarted.

The Versions screen contains the following control fields:

Setting	Description
Running PDE	The currently running version of Teradata Parallel Database Extensions.
Running DBS	The currently running version of the database.
Running TGTW	The currently running versions of Teradata Gateway software.
Running TCHN	The currently running version of Teradata Channel software.
Running RSG	The currently running version of Teradata Relay Services Gateway software.
Running TDGSS	The currently running version of Teradata Generic Security Services software.

Setting	Description
Running PDEGPL	The currently running GNU general public license version of Teradata Parallel Database Extensions.
Desired PDE	Changing this field to an installed PDE version causes that version to be run the next time the database restarts.
Desired DBS	Changing this field to an installed database version causes that version to be run the next time the database restarts.
Desired TGTW	Changing this field to a an installed TGTW version causes that version to be run the next time the database restarts.
Desired TCHN	Changing this field to an installed TCHN version causes that version to be run the next time the database restarts.
Desired RSG	Changing this field to an installed RSG version causes that version to be run the next time the database restarts.
Desired TDGSS	Changing this field to an installed TDGSS version causes that version to be run the next time the database restarts.
Desired PDEGPL	Indicates an installed PDEGPL version to be run the next time the database restarts.  <b>Note:</b> Whenever a user sets the Desired PDE to a different version, the CTL tool automatically sets the Desired PDEGPL to the same version.

## Usage Notes

- Each screen displays groups of related control fields. To change the values of modifiable fields, use the *variable* = setting command, identifying the field either by the full field name or by the alphanumeric identifier that appears next to the field in the screen output. Fields lacking alphanumeric identifiers should not be modified.
- Scripts that change field values should use the full variable names, rather than the alphanumeric identifiers.

## Example: SCREEN DBS output

```
>screen dbs

(0) Minimum Node Action:      Clique-Down
(1) Minimum Nodes Per Clique: 1
(2) FSG cache Percent:       90
(3) Clique Failure:          Clique-Down
(4) Cylinder Read:           On
(5) Restart After DAPowerFail: On
(6) Cylinder Read Ageing Threshold: 0
(7) Maximum Fatal AMPs:      0
(7) TIM FSG cache Percent    50
```

**Example: SCREEN DEBUG output**

```
> scr debug

(0) Start DBS:           On           (1) Break Stop:           Off
(2) Start With Logons: All           (3) Start With Debug:    Off
(4) Save Dumps:           Off          (5) Snapshot Crash:      Off
(6) Maximum Dumps:        5           (7) Start PrgTraces:     Off
(8) Restart Dump Type: Selective      (9) UDF Debugging:       Off
```

**Example: SCREEN RSS output**

```
>screen RSS

(0) Node   Logging Rate:   600 sec

                        RSS Table Logging Enable

(1) SPMA : On      (2) IPMA : Off      (3) SCPU: Off
(4) SVPR : On      (5) IVPR : Off      (6) SLDV: On      (7) SHST: Off
(8) SPDSK: On      (9) SVDSK: On      (A) SAWT: On      (B) SPS : On
(C) SMHM : Off

                        RSS Summary Mode Enable

Summarize SPMA: Off      Summarize IPMA : Off      (D) Summarize SCPU : Off
(E) Summarize SVPR: Off  (F) Summarize IVPR : Off      (G) Summarize SLDV : Off
(H) Summarize SHST: Off  (I) Summarize SPDSK: Off      (J) Summarize SVDSK: Off
(K) Summarize SAWT: Off  Summarize SPS : Off      (L) Summarize SMHM : off
```

**Example: SCREEN VERSION output**

The following is an example of Screen Version command output:

```
Running PDE: 14.00.00.00
(0) Desired PDE:

Running DBS: 14.00.00.00
(1) Desired DBS:

Running RSG:
(2) Desired RSG:

Running TGTW: 14.00.00.00
(3) Desired TGTW:

Running TCHN:
(4) Desired TCHN:

Running TDGSS: 14.00.00.00
```

(5) Desired TDGSS:

Running PDEGPL: 14.00.00.00

Desired PDEGPL:

## *variable=setting*

For a specified variable or field, this command either gives it a specified value or displays information about its valid values.

### Syntax

```
variable = { setting | ? }
```

### Syntax Elements

#### *variable*

Name of variable or name or identifier of field.

If you have used the [SCREEN](#) command to display a group of related control fields, modifiable fields are prefaced by alphanumeric identifiers. You can use either those identifiers or the exact field names with the *variable = setting* command to change the field values.

#### *setting*

Value to give *variable*.

The setting must be an appropriate type for the variable being assigned, or an error is reported.

1, yes, and true can be used as synonyms for on.

0, no, and false can be used as synonyms for off.

Entering an empty string for string variables can cause the current value to be replaced by an empty string.

#### *?*

Displays information about valid values for *variable*.

### Usage Notes

- Changes to the configuration settings are logged automatically. A history of the changes made to the GDO can be viewed using the gdomview command-line tool from the system console. The gdomview online documentation is accessible from a system console using the man and pdehelp commands.

### Example: Setting a ctl field value

```
> scr dbs
```

```
(0) Minimum Node Action:      Clique-Down
```

```
(1) Minimum Nodes Per Clique: 1          (2) FSG cache Percent: 0
```

```

(3) Clique Failure:           Clique-Down      (4) Cylinder Read:           On
(5) Restart After DAPowerFail: On              (6) Cylinder Read Ageing Threshol
d: 0
(7) Maximum Fatal AMPs:       0

> minimum nodes per clique=?
CTL: Minimum Nodes Per Clique
Default value is 1. Valid range is from 1 to 2147483646.
> minimum nodes per clique=3
> screen

(0) Minimum Node Action:      Clique-Down
(1) Minimum Nodes Per Clique: 3              (2) FSG cache Percent: 0
(3) Clique Failure:           Clique-Down      (4) Cylinder Read:           On
(5) Restart After DAPowerFail: On              (6) Cylinder Read Ageing Threshol
d: 0
(7) Maximum Fatal AMPs:       0

> 1=2
> screen

(0) Minimum Node Action:      Clique-Down
(1) Minimum Nodes Per Clique: 2              (2) FSG cache Percent: 0
(3) Clique Failure:           Clique-Down      (4) Cylinder Read:           On
(5) Restart After DAPowerFail: On              (6) Cylinder Read Ageing Threshol
d: 0
(7) Maximum Fatal AMPs:       0

```



## WRITE

The WRITE command saves any configuration changes made during the ctl session back to the source from which they were read. This is usually the PDE Control GDO, if ctl was started without the -file option.

### Syntax

```
{ WRITE | WR }
```

### Usage Notes

The write command does not cause ctl to exit.

Because different users may be modifying the PDE control settings at the same time, ctl merges only the changed settings from the current ctl session to the PDE Control GDO. This minimizes the chances that concurrent users will overwrite each others' changes.

If no changes have been made during the current ctl session, issuing a write command does nothing.

# Cufconfig Utility (cufconfig)

---

**Note:**

The utility described is not available for as-a-service customers.

---

The Cufconfig utility, `cufconfig`, allows you to view and change configuration settings for the user-defined function and external stored procedure subsystem. These configuration settings are stored in the UDF GDO, a globally distributed object. Globally distributed objects store global configuration settings available to all nodes of a Teradata system. The Cufconfig utility is also known as the UDF GDO Configuration utility.

For more information about UDFs, external stored procedures, or user-defined methods (UDMs), see *Teradata Vantage™ - SQL External Routine Programming*, B035-1147.

---

**Note:**

Modifications to `cufconfig` settings take effect after the next database restart.

---

## Runs From

The Cufconfig utility runs from the Linux command line.

For general information on starting the utilities from different interfaces, see [Starting the Utilities](#).

## Syntax

```
cufconfig { -h | -i | -f | -o filename } [...]
```

If you specify multiple options, `cufconfig` processes them in this order: `-i`, `-f`, `-o`

### Syntax Elements

**-h**

Displays the `cufconfig` help.

**-i**

Initializes the UDF GDO field values to the default values of the shipped software version.

**-f *filename***

Modifies the field values of the UDF GDO.

The *filename* specifies the path to the file that is used to modify the GDO. The file should contain a line for each field whose value is to be changed. The available field names and current values are shown in the output of the `cufconfig -o` command.

**-o**

Displays the contents of the UDF GDO.

## Cufconfig Fields

## CLIEnvFile

Specifies the path and name of the file containing environment settings used by CLI-based external stored procedures.

### Default

This field has no default value.

### Usage Notes

An environment file with the same path and file name must exist on all nodes of the system.

### Example: CLI environment settings file

The environment file contains a list of the required environment variables formatted using the standard shell script semantics. There is no support for shell-like parameter substitution. The following shows sample content for an environment file.

```
COPLIB=/usr/lib;  
COPERR=/usr/lib;
```

## CLIncPath

Specifies the path of the directory that contains the header files required by CLI-based external stored procedures.

### Default

/opt/teradata/client/include

## CLILibPath

Specifies the path of the libraries required by CLI-based external stored procedures.

### Default

`/opt/teradata/client/lib`

## CompilerPath

Specifies the path to the C/C++ compiler.

---

### Note:

This field should be changed only under the direction of Teradata Support Center personnel.

---

### Default

Directory containing the C/C++ compiler used for stored procedures (SQL and external stored procedures) and other external routines that are written in C, C++, or Java.

### Usage Notes

This field is set to the path of the C/C++ compiler used for creating external routines.

## CompilerTempDirectory

Specifies the path for the intermediate files used to compile or link external routines.

---

**Note:**

This field should be changed only by Teradata Support Center personnel.

---

**Default**

*PDE\_temp\_path*/UDFTemp

where *PDE\_temp\_path* is the PDE Temp directory.

Typically, this is /var/opt/teradata/tdtemp/UDFTemp/

To determine the path of the PDE Temp directory, enter the following on the command line:

```
pdepath -S
```



## CStdUDFServerTimeout

Specifies the timeout value in minutes that the database server will wait to start a UDF server to execute a UDF or an external stored procedure that is written in C or C++. If the time specified by this setting is exceeded, the server generates an error and aborts the request for a UDF server.

---

**Note:**

Teradata recommends that the value of this field be changed only under the direction of Teradata Support Center personnel.

---

**Valid Range**

1 through 200 minutes

**Default**

2 minutes

**Usage Notes**

This setting may need to be increased if you are receiving 7583 errors when running requests that contain UDFs or external stored procedures on a busy Teradata system.

**Related Information**

For more information on Java external stored procedures and Java user-defined functions, see *Teradata Vantage™ - SQL External Routine Programming*, B035-1147.

## GLOPLockTimeout

Specifies the maximum time, in seconds, that an external routine (UDF, UDM, or external stored procedure) will wait for a lock request to be granted on global and persistent (GLOP) data.

GLOP data is a type of data available to external routines. The persistence in memory of GLOP data is based on specific boundaries, such as a user role or user account.

### Valid Range

0 to 2147483647 (the maximum size of an SQL INTEGER data type)

### Default

130 seconds

### Usage Notes

This should be a value slightly greater than the GLOPLockWait value.

When an external routine requests a lock on GLOP data, and the timeout value specified by GLOPLockTimeout is exceeded, the system returns a -2 error code to the external routine.

### Related Information

For more information on global and persistent data, see *Teradata Vantage™ - SQL External Routine Programming*, B035-1147.

## GLOPLockWait

Specifies the maximum time, in seconds, that an external routine can hold a lock on global and persistent (GLOP) data if another external routine is waiting for the lock to be freed. When the lock exceeds the GLOPLockWait value, the transaction that was holding the lock is aborted, and the lock is freed.

GLOP data is a type of data available to external routines. The persistence in memory of GLOP data is based on specific boundaries, such as a user role or user account.

### Valid Range

0 to 2147483647 (the maximum size of an SQL INTEGER data type)

### Default

120 seconds

### Usage Notes

This should be a value slightly less than the value for GLOPLockTimeout value.

### Related Information

For more information on global and persistent data, see *Teradata Vantage™ - SQL External Routine Programming*, B035-1147.

## GLOPMemMapPath

Specifies the location of the shared memory and cache files used by the global and persistent (GLOP) data feature.

GLOP data is a type of data available to external routines. The persistence in memory of GLOP data is based on specific boundaries, such as a user role or user account.

---

**Note:**

This field is for informational purposes only, and should not be changed.

---

### Related Information

For more information on global and persistent data, see *Teradata Vantage™ - SQL External Routine Programming*, B035-1147.

## GPLUDFServerMemSize

Specifies the maximum size in bytes of the udfGPL server.

### Valid Range

0 - 3758096384 bytes

### Default

3758096384 bytes (3.5 GB)

### Usage Notes

0 means to use the maximum available memory.

### Related Information

For more information about GPLUDFServerMemSize, see *Teradata Vantage™ - SQL External Routine Programming*, B035-1147.

## HybridServer Fields

Each node of a Teradata system can have one hybrid server that provides protected mode, multiple-threaded execution for external language procedures and UDFs to all AMPs and PEs on the node. (Protected mode UDFs are those that do not include the EXTERNAL SECURITY clause in their UDF SQL definitions.)

Each group of HybridServer fields specifies settings that the server uses when servicing procedures and UDFs written in the language specified by the HybridServerLanguage field of the group.

### Example: HybridServer fields

The following group of cufconfig fields, all HybridServer3 settings, specify custom settings used by the hybrid server when running UDFs defined to have a LANGUAGE attribute of EAH.

```
HybridServer3Execution: /usr/tdbms/bin/udfsectsk
HybridServer3Library:
HybridServer3LibPath:
HybridServer3IncPath:
HybridServer3EnvFile:
HybridServer3MemSize: 262144
HybridServer3Threads: 20
HybridServer3Language (display only): EAH
HybridServer3AppCat (display only): 8
HybridServer3FreeSpace: 0
HybridServer3MemFileSize: 0
HybridServer3JREPath: /opt/teradata/jvm64/jre8/jre/
HybridServer3JVMOptions:
```

---

#### Note:

EAH is the external access handler language used exclusively for Teradata QueryGrid functions. This group of HybridServer settings should be changed only under the direction of the Teradata Support Center.

---

## JavaBaseDebugPort

Specifies the base number used to determine the debug port for Java external routines (Java external stored procedures and Java user-defined functions). This field is only used when debugging Java external routines.

### Default

0 (The JVM debug port is disabled.)

### Usage Notes

The actual port number to use for attaching a debugger to a Java external routine is the value of JavaBaseDebugPort plus an offset.

For Java user-defined functions running on a hybrid server, the offset is always 2000, so the debug port for these routines is always the value of JavaBaseDebugPort plus 2000.

For Java external routines running on secure servers, the offset depends on which instance of the server is executing the routine (there can be up to 10 secure servers per vproc), and on the type and ID of the vproc that is executing the routine. There are a total of 1000 offsets per node available for PE processes, and 1000 available for AMP processes. Therefore, only the following vprocs have debug ports available for Java external routines on each node:

- The first hundred AMPs that have the lowest vproc ID numbers
- The top hundred PEs that have the highest vproc ID numbers

The following table demonstrates how representative offsets are determined on a node with continuous vproc assignment.

#### Note:

Vproc IDs for PEs that have debug ports will be lower for systems upgraded from pre-14.0 releases.

Vproc Type	Vproc ID	Secure Server Instance	Offset
PE	30719 (16383 on upgraded systems)	1 through 10	0 through 9
PE	30718 (16382 on upgraded systems)	1 through 10	10 through 19
PE	30717 (16381 on upgraded systems)	1 through 10	20 through 29
	continuing on through:		
PE	(16284 on upgraded systems)	1 through 10	990 through 999
PE	30619 or lower (16283 on upgraded systems)	None available	None available

Vproc Type	Vproc ID	Secure Server Instance	Offset
AMP	0	1 through 10	1000 through 1009
AMP	1	1 through 10	1010 through 1019
AMP	2	1 through 10	1020 through 1029
	continuing on through:		
AMP	99	1 through 10	1990 through 1999
AMP	100 or higher	None available	None available

### Example: Java debug ports

Assume the value of JavaBaseDebugPort is 8000.

- To debug any external routine running on a hybrid server, connect the debugger to port 10,000 (8000 plus the offset of 2000).
- To debug an external routine running on the second secure server on PE 30719, connect the debugger to port 8001 (8000 plus the offset of 1).
- To debug an external routine running on the fifth secure server on PE 30717, connect the debugger to port 8024 (8000 plus the offset of 24).
- To debug an external routine running on the first secure server on AMP 0, connect the debugger to port 9000 (8000 plus the offset of 1000).

### Related Information

For more information on Java external stored procedures and Java user-defined functions, see *Teradata Vantage™ - SQL External Routine Programming*, B035-1147.



## JavaEnvFile

Specifies the environment file for Java Virtual Machine (JVM) startup.

---

**Note:**

This field should be changed only by Teradata Support Center personnel.

---

**Default**

This field has no initial value.

**Usage Notes**

In the environment file, each line specifies a JavaVMInitArgs option or an environment variable for the UDF server process. This permits the database administrator to configure the JVM as needed for Java external stored procedures routines.

The lines for JavaVMInitArgs should be before the lines for environment variables in the file.

An environment file with the same path and file name must exist on all nodes of the system.

**Example: Java environment settings file**

The contents of the environment file is similar to the following:

```
[REG]-Xms50m
[REG]-Xmx100m
[HYB]-Xms200m
[HYB]-Xmx500m

[Environment Variables]
UDF_DEBUG_WAIT=1
```

## JavaHybridThreads

Specifies the maximum number of threads that can be run simultaneously by the Java hybrid server.

### Valid Range

0 through the setting for SecureServerAMPs or SecureServerPEs, whichever is smaller.

### Default

20

### Usage Notes

Each node can have one hybrid server that provides multiple threaded execution of protected mode Java UDFs to all AMPs and PEs on the node. (Protected mode Java UDFs are those that do not include the EXTERNAL SECURITY clause in their UDF SQL definitions.)

When this field is set to zero, Java UDFs cannot be run by the hybrid server, so must be run by Java secure servers, which are single threaded.

If JavaHybridThreads and JavaServerTasks are both set to zero, neither Java UDFs nor Java external stored procedures can be run.

### Related Information

For more information on Java external stored procedures and Java user-defined functions, see *Teradata Vantage™ - SQL External Routine Programming*, B035-1147.

## JavaLibraryPath

Specifies the root directory path where the JAR files are stored. These files are used by Java external routines.

---

**Note:**

This field should be changed only by Teradata Support Center personnel.

---

**Default**

*config\_path*/jarlib/

where *config\_path* is the PDE configuration directory.

Typically, this is /etc/opt/teradata/tdconfig/jarlib/

To determine the path of the PDE configuration directory, enter the following on the command line:

```
pdepath -c
```

## JavaLogPath

Specifies the location of the log files for Java external routines.

### Default

/tmp/

### Usage Notes

If a JavaLogPath is specified and Java logging is enabled, the standard error and standard out of the Java server task are redirected to the files in the JavaLogPath directory for debugging purposes.

## JavaServerTasks

Specifies the number of Java secure servers that can be run simultaneously per vproc, subject to the limitations imposed by the ParallelUserServerAMPs and ParallelUserServerPEs settings.

### Valid Range

0 to 20

### Default

2

### Usage Notes

Java secure servers can be used by Java external stored procedures and secure-mode Java UDFs. (Secure-mode Java UDFs are those that include the EXTERNAL SECURITY clause in their UDF SQL definitions.)

Java secure servers are single threaded. This is in contrast to Java hybrid servers, which can execute multiple Java UDFs, each one in its own thread. The number of threads that can be run by a Java hybrid server is controlled by the JavaHybridThreads field.

If the JavaServerTasks field is zero, you cannot run any Java external stored procedures, and Java UDFs must run by a Java hybrid server.

JavaServerTasks is analogous to the UDFServerTasks field used for native based UDFs, external stored procedures, or UDMs.

### Related Information

For more information on Java external stored procedures and Java user-defined functions, see *Teradata Vantage™ - SQL External Routine Programming*, B035-1147.

## JavaVersion

Specifies the JNI version used by Java external routines. This value is a hexadecimal number.

---

**Note:**

This field should be changed only by Teradata Support Center personnel.

---

## JHybUDFServerTimeout

Specifies the timeout value in minutes that the database server will wait to start a UDF server to execute a UDF or an external stored procedure that is written in Java and run under the default user, tdatuser. If the time specified by this setting is exceeded, the server generates an error and aborts the request for a UDF server.

---

**Note:**

Teradata recommends that the value of this field be changed only under the direction of Teradata Support Center personnel.

---

**Valid Range**

1 through 200 minutes

**Default**

10 minutes

**Usage Notes**

This setting may need to be increased if you are receiving 7583 errors when running requests that contain UDFs or external stored procedures on a busy Teradata system.

**Related Information**

For more information on Java external stored procedures and Java user-defined functions, see *Teradata Vantage™ - SQL External Routine Programming*, B035-1147.

## JREPath

Specifies the path where the Java Runtime Environment (JRE) was installed.

---

**Note:**

This field should be changed only by Teradata Support Center personnel.

---

## Usage Notes

To run Java external routines, the Java server looks at the JREPath to find the required executable files and JVM library files.

The Teradata JRE is the Java runtime engine used by Vantage. To install the Teradata JRE, the Teradata JRE package must be installed prior to installing the TDBMS package. The Teradata JRE package name includes "teradata-jre8".



## JStdUDFServerTimeout

Specifies the timeout value in minutes that the database server will wait to start a UDF server to execute a UDF or an external stored procedure that is written in Java and run under an authorized user. If the time specified by this setting is exceeded, the server generates an error and aborts the request for a UDF server.

---

**Note:**

Teradata recommends that the value of this field be changed only under the direction of Teradata Support Center personnel.

---

**Valid Range**

1 through 200 minutes

**Default**

5 minutes

**Usage Notes**

This setting may need to be increased if you are receiving 7583 errors when running requests that contain UDFs or external stored procedures on a busy Teradata system.

**Related Information**

For more information on Java external stored procedures and Java user-defined functions, see *Teradata Vantage™ - SQL External Routine Programming*, B035-1147.

## JSVServerMemPath

Specifies the location of the hybrid server control files.

### Default

`/var/opt/teradata/tdtemp/jsvsrv/`

### Usage Notes

The hybrid server uses one control shared memory file for each AMP and PE.

## JVMOptions

Specifies Java Virtual Machine settings to be applied to all nodes of the Teradata system.

### Example: JVM options

```
JVMOptions: -server -XX:UseParallelGC -XX:+UseParallelOldGC -Xms7100m -Xmx7100m  
-XX:NewSize=2370m -XX:MaxNewSize=2370m -XX:MaxPermSize=864m
```

## LinkerPath

Specifies the path to the C\C++ linker.

---

**Note:**

This field should be changed only by Teradata Support Center personnel.

---

## MallocLimit

Specifies the upper limit (in bytes) on the amount of memory that an external routine (UDF, UDM, or external stored procedure) can allocate using the Teradata C library function FNC\_malloc.

### Valid Range

0 to 18446744073709551615 bytes (hex FFFF FFFF FFFF FFFF)

### Default

33554432 bytes (32MB)

### Usage Notes

For performance reasons, a function should allocate only as much memory as absolutely required. UDFs execute in parallel on the database. An all-AMP query that includes a UDF which allocates 10MB of memory can use up to 1GB for all instances on a 100 AMP system for just the one transaction.

### Related Information

For more information on the Teradata C-library functions, see *Teradata Vantage™ - SQL External Routine Programming*, B035-1147.

## MaximumCompilations

Specifies the maximum number of UDFs, external stored procedures, or UDMs that can be compiled simultaneously, in different sessions, on any one node.

### Default

10

### Usage Notes

Teradata recommends leaving this setting at the default.

## MaximumGLOPMem

Specifies the maximum amount of memory, in pages, an AMP or PE can use to store global and persistent (GLOP) data mappings. A page of memory is 4096 bytes.

GLOP data is a type of data available to external routines. The persistence in memory of GLOP data is based on specific boundaries, such as a user role or user account.

---

**Note:**

This value should be changed only under the direction of Teradata Support Center personnel.

---

### Valid Range

0 through 16777216 pages (64 GB)

---

**Note:**

System memory availability, system disk availability, and system performance considerations can substantially reduce the upper limit.

---

### Default

2560 pages

### Usage Notes

This value must be larger than the value of MaximumGLOPSize.

### Related Information

For more information on global and persistent data, see *Teradata Vantage™ - SQL External Routine Programming*, B035-1147.

## MaximumGLOPPages

Specifies the maximum number of global and persistent (GLOP) data pages that can be allocated per item of read-only GLOP data. (Read/write GLOP data can consist of only a single page.)

GLOP data is a type of data available to external routines. The persistence in memory of GLOP data is based on specific boundaries, such as a user role or user account.

### Valid Range

0 to 2147483647 (the maximum size of an SQL INTEGER data type)

### Default

8 pages

### Related Information

For more information on global and persistent data, see *Teradata Vantage™ - SQL External Routine Programming*, B035-1147.



## MaximumGLOPSize

Specifies the maximum size, in memory pages, of an item of global and persistent (GLOP) data. A page of memory is 4096 bytes.

GLOP data is a type of data available to external routines. The persistence in memory of GLOP data is based on specific boundaries, such as a user role or user account.

---

**Note:**

This value should be changed only under the direction of Teradata Support Center personnel.

---

### Valid Range

0 through 511985 pages (about 2 GB)

---

**Note:**

System memory availability, system disk availability, and system performance considerations can substantially reduce the upper limit.

---

### Default

256 pages (1 MB)

### Related Information

For more information on global and persistent data, see *Teradata Vantage™ - SQL External Routine Programming*, B035-1147.

## ModTime

Displays the timestamp of the last UDF GDO modification.

---

**Note:**

This field should not be modified.

---

## ParallelUserServerAMPs

Specifies the maximum number of secure servers allowed to run simultaneously using the same operating system user ID on an AMP vproc.

### Valid Range

1 through the value of SecureServerAMPs

### Default

2

### Usage Notes

For example, a value of 2 allows two UDFs using identical operating system user IDs to run simultaneously. A third UDF would have to wait until one of the other UDFs finish.

### Related Information

For information on the maximum number of secure servers allowed to run simultaneously using the same operating system user ID on a PE vproc, see the UDF GDO field, [ParallelUserServerPEs](#).

## ParallelUserServerPEs

Specifies the maximum number of secure servers allowed to run simultaneously using the same operating system user ID on a PE vproc.

### Valid Range

1 through the value of SecureServerPEs

### Default

2

### Usage Notes

For example, a value of 2 allows two external stored procedures using identical operating system user IDs to run simultaneously. A third external stored procedure would have to wait until one of the other external stored procedures finish.

### Related Information

For information on the maximum number of secure servers allowed to run simultaneously using the same operating system user ID on an AMP vproc, see [ParallelUserServerAMPs](#).

## PDESTODisabled

The PDESTODisabled field of Cufconfig is for internal use only.

## ScriptMemLimit

Specifies a limit to the amount of memory (in bytes) that can be allocated by processes spawned by the SCRIPT table operator.

### Valid Range

The value can be zero or any positive integer that specifies the limit as a number of bytes.

When set to zero, the memory used is not limited.

### Default

33,554,432 bytes (32 MB)

### Related Information

For information on the SCRIPT table operator, see *Teradata Vantage™ - SQL Operators and User-Defined Functions*, B035-1210.

## SecureServerAMPs

Specifies the maximum number of secure servers allowed to run simultaneously on an AMP vproc.

### Valid Range

0 through 120

### Default

20

### Usage Notes

Teradata recommends this be set to a value between 15 and 30.

## SecureServerPEs

Specifies the maximum number of secure servers allowed to run simultaneously on a PE vproc.

### Valid Range

0 through 120

### Default

20

### Usage Notes

The required number of secure servers is limited to the value of the SecureServerPEs field.

## SourceDirectoryPath

Specifies the default path of the UDF source directory when copying the UDF source code onto the server.

### Default

*config\_path*/Teradata/tdbs\_udf/usr/

where *config\_path* is the PDE configuration directory.

Typically, this is /etc/opt/teradata/tdconfig/Teradata/tdbs\_udf/usr/

To determine the path of the PDE configuration directory, enter the following on the command line:

```
pdepath -c
```

---

### Note:

Teradata recommends leaving this set to the default value.

---

### Usage Notes

The UDF source code is located on the software distribution node.



## SWDistNodeID

Displays the node ID of the software distribution node.

---

**Note:**

This field is for informational purposes only, and should not be changed.

---

### Usage Notes

The node ID is the location of the source or object code when using the server option of the EXTERNAL clause in a CREATE statement for a UDF, external stored procedure, or UDM. For example:

```
SS!myudf!myudf.c
```

## TDSPLibBase

Specifies the base library directory for stored procedures (SQL and external stored procedures).

---

**Note:**

This field should be changed only by Teradata Support Center personnel.

---

**Default**

`config_path\tdsplib\`

where *config\_path* is the PDE configuration directory.

Typically, this is `/etc/opt/teradata/tdconfig/tdsplib/`

To determine the path of the PDE configuration directory, enter the following on the command line:

`pdepath -c`

## UDFEnvFile

Specifies the path and name of the file containing the environment settings used by UDFs, external stored procedures, or UDMs that specify a data access clause of NO SQL in the CREATE or REPLACE statement for the UDF, external stored procedure, or UDM.

### Default

This field has no default value.

### Usage Notes

An environment file with the same path and file name must exist on all nodes of the system.

### Example: UDF environment settings file

The environment file contains a list of the required environment variables formatted using the standard shell script semantics. There is no support for shell-like parameter substitution. The following shows sample content for an environment file.

```
COPLIB=/usr/lib;  
COPERR=/usr/lib;
```

## UDFIncPath

Specifies a header file path in addition to the standard path for use by UDFs, external stored procedures, or UDMs that specify a data access clause of NO SQL in the CREATE PROCEDURE or REPLACE PROCEDURE statement.

### Default

This field has no default value.

## UDFLibPath

Specifies a library path in addition to the standard library path for use by UDFs, external stored procedures, or UDMs that specify a data access clause of NO SQL in the CREATE PROCEDURE or REPLACE PROCEDURE statement.

### Default

This field has no default value.

## UDFLibraryPath

Specifies the root directory path of the UDF, external stored procedure, or UDM linked libraries.

---

**Note:**

This field should be changed only by Teradata Support Center personnel.

---

**Default**

*config\_path*\udflib\

where *config\_path* is the PDE configuration directory.

Typically, this is /etc/opt/teradata/tdconfig/udflib/.

To determine the path of the PDE configuration directory, enter the following on the command line:

```
pdepath -c
```

## UDFServerMemPath

Specifies the path to the UDF, external stored procedure, or UDM shared memory files.

---

**Note:**

This field should be changed only by Teradata Support Center personnel.

---

**Default**

*PDE\_temp\_path\udfsrv\*

where *PDE\_temp\_path* is the PDE Temp directory.

Typically, this is `/var/opt/teradata/tdtemp/udfsrv/`.

To determine the path of the PDE Temp directory, enter the following on the command line:

```
pdepath -S
```

## UDFServerTasks

Determines the number of protected mode UDFs, external stored procedures, or UDMs that can be run simultaneously per vproc.

### Valid Range

0 through 120

### Default

50

### Usage Notes

Each routine is a separate process that runs as the built-in operating system user named tdatuser.

If the UDFServerTasks field is zero, no C or C++ protected mode UDFs, external stored procedures, or UDMs can run.



## UIFLibraryPath

Specifies the directory path for the storage of user installed files (UIFs), usually script files used with the SCRIPT table operator.

### Default

`/var/opt/teradata/tdtemp/uiflib`

### Usage Notes

If this field is changed after UIFs are already installed on the system, then any pre-existing UIFs must be moved to the new location, either by copying them from the original location or by executing the SYSUIF.REDISTRIBUTE\_FILE external stored procedure.

### Related Information

For more information about UIFs, the SCRIPT table operator, and the SYSUIF.REDISTRIBUTE\_FILE procedure, see *Teradata Vantage™ - SQL Operators and User-Defined Functions*, B035-1210.

## UIFTmpPath

Specifies the path where a temporary directory will be created and available for use during the execution of a script UIF. This temporary directory is deleted after the script execution.

### Default

`/var/opt/teradata/tdtemp/uiftemp`

### Usage Notes

This location should be used for any intermediate UIF script processing.

### Related Information

For more information about UIFs, and the SCRIPT table operator, see *Teradata Vantage™ - SQL Operators and User-Defined Functions*, B035-1210.

## Version

Specifies the version of the UDF GDO. The version only changes when new fields are added or modified.

---

**Note:**

This field should not be modified.

---

## Examples

### Example: Displaying the Cufconfig help

To display the cufconfig help, type the following:

```
cufconfig -h
```

The following appears:

```
valid options:
-o
  outputs the gdo to the screen.
-i
  initializes the gdo to the defaults.
-f filename
  modifies the gdo fields identified
  in filename. Fields not specified in filename
  will not be set or reset by this tool.
```

Note that if several options are specified they are analyzed in the following order:

```
-i -f -o
```

If only a few fields need to be set differently than the defaults, specify the -i option with the -f option. Use the -o option to verify the settings.

Exiting cufconfig...

### Example: Displaying the contents of the UDF GDO

To display the contents of the UDF GDO, type the following:

```
cufconfig -o
```

```
Version: 7
ModTime: 1308613095
SWDistNodeID: 33
SourceDirectoryPath: /etc/opt/teradata/tdconfig/Teradata/tdbs_udf/usr/
CompilerTempDirectory: /var/opt/teradata/tdtemp/UDFTemp/
UDFLibraryPath: /etc/opt/teradata/tdconfig/udflib/
CompilerPath: /usr/bin/gcc
LinkerPath: /usr/bin/ld
UDFServerMemPath: /var/opt/teradata/tdtemp/udfsrv/
MaximumCompilations: 10
UDFServerTasks: 2
```

```

SecureServerAMPs: 20
ParallelUserServerAMPs: 2
SecureServerPEs: 20
ParallelUserServerPEs: 2
TDSPLibBase: /etc/opt/teradata/tdconfig/tdsplib/
UDFLibPath:
UDFIncPath:
UDFEnvFile:
CLILibPath: /opt/teradata/client/14.00/lib64
CLIIncPath: /opt/teradata/client/14.00/include
CLIEnvFile:
JavaLibraryPath: /etc/opt/teradata/tdconfig/jarlib/
JREPath: /opt/teradata/jvm64/jre8/jre/
JavaLogPath: /tmp/
JavaEnvFile:
JavaServerTasks: 2
JavaHybridThreads: 20
JavaVersion: 0x10004
JavaBaseDebugPort: 0
JSVServerMemPath: /var/opt/teradata/tdtemp/jsvsrv/
MallocLimit: 33554432
GLOPLockTimeout: 130
GLOPLockWait: 120
MaximumGLOPSize: 256
MaximumGLOPMem: 2560
MaximumGLOPPages: 8
GLOPMapMemPath: /var/opt/teradata/tdtemp/udfglopdata/
CServerMemSize: 262144
JavaHybridServerMemSize: 262144
HybridServer1Execution: /usr/tdbms/bin/udfsas
HybridServer1Library: /usr/tdbms/lib/libudf3.so
HybridServer1LibPath:
HybridServer1IncPath:
HybridServer1EnvFile:
HybridServer1MemSize: 262144
HybridServer1Threads: 20
HybridServer1Language (display only): SAS
HybridServer1AppCat (display only): 6
HybridServer2Execution:
HybridServer2Library:
HybridServer2LibPath:
HybridServer2IncPath:
HybridServer2EnvFile:
HybridServer2MemSize: 262144

```

```

HybridServer2Threads: 0
HybridServer2Language (display only):
HybridServer2AppCat (display only): 7
HybridServer3Execution: /usr/tdbms/bin/udfsectsk
HybridServer3Library:
HybridServer3LibPath:
HybridServer3IncPath:
HybridServer3EnvFile:
HybridServer3MemSize: 262144
HybridServer3Threads: 20
HybridServer3Language (display only): EAH
HybridServer3AppCat (display only): 8
HybridServer3FreeSpace: 0
HybridServer3MemFileSize: 0
HybridServer3JREPath: /opt/teradata/jvm64/jre8/jre/
HybridServer3JVMOptions:
HybridServer4Execution:
HybridServer4Library:
HybridServer4LibPath:
HybridServer4IncPath:
HybridServer4EnvFile:
HybridServer4MemSize: 262144
HybridServer4Threads: 0
HybridServer4Language (display only):
HybridServer4AppCat (display only): 9
HybridServer5Execution:
HybridServer5Library:
HybridServer5LibPath:
HybridServer5IncPath:
HybridServer5EnvFile:
HybridServer5MemSize: 262144
HybridServer5Threads: 0
HybridServer5Language (display only):
HybridServer5AppCat (display only): 10
HybridServer6Execution:
HybridServer6Library:
HybridServer6LibPath:
HybridServer6IncPath:
HybridServer6EnvFile:
HybridServer6MemSize: 262144
HybridServer6Threads: 0
HybridServer6Language (display only):
HybridServer6AppCat (display only): 11
HybridServer7Execution:

```

```

HybridServer7Library:
HybridServer7LibPath:
HybridServer7IncPath:
HybridServer7EnvFile:
HybridServer7MemSize: 262144
HybridServer7Threads: 0
HybridServer7Language (display only):
HybridServer7AppCat (display only): 12
HybridServer8Execution:
HybridServer8Library:
HybridServer8LibPath:
HybridServer8IncPath:
HybridServer8EnvFile:
HybridServer8MemSize: 262144
HybridServer8Threads: 0
HybridServer8Language (display only):
HybridServer8AppCat (display only): 13
HybridServer9Execution:
HybridServer9Library:
HybridServer9LibPath:
HybridServer9IncPath:
HybridServer9EnvFile:
HybridServer9MemSize: 262144
HybridServer9Threads: 0
HybridServer9Language (display only):
HybridServer9AppCat (display only): 14
CStdUDFServerTimeout: 2
JStdUDFServerTimeout: 5
JHybUDFServerTimeout: 10
USRLibraryPath: /usr/tdbms/lib
GPLUDFServerMemSize: 33554432
ScriptMemLimit: 33554432
JVMOptions:

```

### Example: Modifying fields of the UDF GDO

#### Note:

Before you begin, make sure your computer has a minimum of 256 KB of free disk space to run the protected or secure mode server process. Each server process requires 256 KB of space.

Do the following steps to modify the fields of the UDF GDO:

1. Open a new text file, or open an existing UDF configuration file.
2. In a text editor, type the fields and field values you want to modify. For example:

```

UDFServerTasks:10
ParallelUserServerAMPs:4
ParallelUserServerPEs:4
UDFEnvFile: /home/sally/config/udf/UDFEnvFile.txt
CLILibPath: /opt/teradata/client/14.00/lib64
CLIIncPath: /opt/teradata/client/14.00/include
CLIEnvFile: /home/sally/config/CLI/CLIEnvFile.txt
JavaBaseDebugPort: 8000
JavaLogPath: /tmp/
JavaEnvFile: /home/sally/config/jvm/jvmenv.txt
JavaServerTasks: 10

```

**Note:**

The field names are case sensitive and must follow the same format as the output from the -o option. The UDF configuration file should contain only the fields you want to modify, with each field on a separate line as shown above.

3. Save the UDF configuration file as a text file. For example, save the file as:
4. Use the -f option to modify the UDF GDO with the field values specified in the UDF configuration file, and use the -o option to display the updated contents of the UDF GDO.

Using the UDF GDO contents in [Example: Displaying the contents of the UDF GDO](#) if you apply the field changes from this example, the new output of the UDF GDO would be the following:

```

Version: 6
ModTime: 1206484172
SWDistNodeID: 33
SourceDirectoryPath: /etc/opt/teradata/tdconfig/Teradata/tdbs_udf/usr/
CompilerTempDirectory: /var/opt/teradata/tdtemp/UDFTemp/
UDFLibraryPath: /etc/opt/teradata/tdconfig/udflib/
CompilerPath: /usr/bin/gcc
LinkerPath: /usr/bin/ld
UDFServerMemPath: /var/opt/teradata/tdtemp/udfsrv/
MaximumCompilations: 10
UDFServerTasks: 10
SecureServerAMPs: 20
ParallelUserServerAMPs: 4
SecureServerPEs: 20
ParallelUserServerPEs: 4
TDSPLibBase: /etc/opt/teradata/tdconfig/tdsplib/
UDFLibPath:
UDFIncPath:
UDFEnvFile: /home/sally/config/udf/UDFEnvFile.txt

```



```

CLILibPath: /opt/teradata/client/14.00/lib64
CLIIncPath: /opt/teradata/client/14.00/include
CLIEnvFile: /home/sally/config/CLI/CLIEnvFile.txt
JavaLibraryPath: /etc/opt/teradata/tdconfig/jarlib/
JREPath: /opt/teradata/jvm64/jre8/jre/
JavaLogPath: /tmp/
JavaEnvFile: /home/sally/config/jvm/jvmenv.txt
JavaServerTasks: 10
JavaHybridThreads: 20
JavaVersion: 0x10004
JavaBaseDebugPort: 8000
JSVServerMemPath: /var/opt/teradata/tdtemp/jsvsrv/
MallocLimit: 33554432
GLOPLockTimeout: 130
GLOPLockWait: 120
MaximumGLOPSize: 256
MaximumGLOPMem: 2560
MaximumGLOPPages: 8
GLOPMapMemPath: /var/opt/teradata/tdtemp/udfglopdata/

```

... and so forth.

### Example: Using a JavaEnvFile environment file

This example illustrates a use of the JavaEnvFile environment file. By default, a DBS Java external stored procedure uses the tdgssconfig.jar file located in the TDBMS\bin directory when connecting to the DBS session. To have the DBS use nondefault values for the tdgssconfig.jar file, copy, modify, and place the file in the tdconfig directory on each node. Then create a JavaEnvFile to configure the JVM to use the copy of the tdgssconfig.jar file.

---

#### Note:

When modifying the tdgssconfig.jar file, make a copy of it. The file may be overwritten when the DBS package is re-installed.

---

```
-Djava.class.path=/usr/tdbms/bin/javFnc.jar;/usr/tdbms/bin/terajdbc4.jar;/usr/tdbms/bin/tdgssjava.jar;/etc/opt/teradata/tdconfig/tdgssconfig.jar;
```

This configures the JVM to use the standard javFnc.jar, terajdbc4.jar, and tdgssjava.jar files, but use the alternate tdgssconfig.jar file.

For more information about the Teradata JDBC driver, see *Teradata JDBC Driver Reference*, available at <https://teradata-docs.s3.amazonaws.com/doc/connectivity/jdbc/reference/current/frameset.html>.

For more information about the javFnc.jar file, see *Teradata Vantage™ - SQL External Routine Programming*, B035-1147.

# Database Initialization Program (dip)

---

## Note:

The utility described is not available for as-a-service customers.

---

The Database Initialization Program, dip (abbreviated throughout the documentation as DIP), is a series of executable SQL script files packaged with Advanced SQL Engine. Each DIP script creates one or more database objects for or used by the following:

- Advanced SQL Engine:
  - To implement a feature, such as security access logging, client character sets, and calendar arithmetic.
  - To store system information, such as PDE crashdumps, resource usage, query text, and error message text.
- Users:
  - To view the contents of the Data Dictionary (system catalog) tables.
  - To generate system reports, such as resource usage statistics

DIP allows you to execute one, several, or all script files, which create a variety of database objects. The most commonly needed DIP scripts are executed during Advanced SQL Engine installation, but you can run the DIP utility at any time to add optional features.

## Runs From

- Database Window or comparable interface to the Teradata console subsystem, such as cnstern
- Linux command line

For general information on starting the utilities from different interfaces, see [Starting the Utilities](#).

## Running DIP Interactively

1. Do either of the following:
  - From the **Supervisor** window of CNS, type `start dip`.
  - From the Linux command line, type `dip`.

The DIP utility presents a numbered list of available DIP scripts (also available in [DIP Scripts](#)).

2. Type the number or name of the script to execute and press **Enter**.

A message similar to the following appears:

```
Executing DIPERR at Tue Feb 20 13:04:29 2001
Please wait...
```

```
DIPERR is complete at Tue Feb 20 13:04:46 2001
Please review the results in /tmp/dip1.txt on node 1- 4
```

**Note:**

The results of the script are saved into a text file, which is named using DIP and the number of the script.

3. Do either of the following:
  - To run another script, type **Y**, press **Enter**, repeat the preceding step.
  - To exit DIP, type **N** and press **Enter**.

## Running DIP Noninteractively

You can run DIP with a single-line command from the Supervisor window of Database Window (or any interface to the console subsystem (CNS), such as the cnstern command-line utility).

### Syntax

```
dip {
  dbc_password { script_number | script_name | -s list_of_scripts } |
  -h
}
```

### Syntax Elements

#### *dbc\_password*

The password for user DBC.

If the password contains the # character, enclose the *dbc\_password* within quotes or apostrophes: "our#dbc#password"

#### *script\_number*

The DIP menu number of the script to be run.

#### *script\_name*

The DIP menu name of the script to be run, for example, DIPALL or DIPVIEWS.

#### *-s list\_of\_scripts*

Displays a list of script numbers or names separated by spaces to be executed in that order.

For example:

- 1 2 5 6
- DIPERR DIPDB DIPVIEWS DIPVIEWSV

For a summary of DIP scripts, see [DIP Scripts](#).

**-h**

Displays DIP online help information.

## DIP Scripts

The following table provides a summary of the different DIP scripts.

SQL Script File	Purpose
DIPACC	<ul style="list-style-type: none"> <li>Runs the access logging script.</li> <li>Enables users with EXECUTE privilege on the DBC.AccLogRule macro to submit BEGIN LOGGING and END LOGGING statements, which enable and disable logging of privilege checking.</li> </ul>
DIPACR	Loads and Initializes the TD_ARRAY data type.
DIPAEXT	Creates Teradata Analytic Functions.
DIPALL	Runs all DIP scripts that appear above DIPALL on the DIP menu screen.
DIPANSI	Initializes ANSI supported stored procedures.
DIPARR	Creates the TD_ARRAY data type and corresponding functions.
DIPBAR	Provides infrastructure to support Data Stream Architecture (DSA) Backup And Recovery (BAR).
DIPCAL	Creates TD_SYSFNLIB database, if it does not already exist, and enables Teradata embedded services system functions, calendar views, and calendar macros.
DIPCCS	<p>Installs the default client character sets for the system.</p> <p>Creates pre-defined translation tables for Unicode and non-English languages, including Japanese and various European languages. For more information on character sets, see <i>Teradata Vantage™ - Advanced SQL Engine International Character Set Support</i>, B035-1125.</p>
DIPCDC	Creates system-reserved replication groups and other infrastructure used for Teradata® Unity™ change data capture protocol for replication. This script is visible only when Teradata Unity CDC is enabled in DBC Control.
DIPCRASH	Creates the database and views where crash dump information is stored.
DIPDATASET	Adds support for the DATASET data type and the Avro storage format. If a user-defined data type named DATASET already exists, this script is aborted, and the Teradata DATASET type is not created. The script may be re-run after the user-defined DATASET type is removed or renamed.
DIPDB	Creates all of the databases required by DIP scripts, and initializes the secondary indexes for DBC tables.

SQL Script File	Purpose
DIPDBCCMT	Initializes the comment string values for DBC tables, views, and macros. Consequently, HELP statement output displays information about expected values for table and view fields, and for macro parameters.
DIPDEM	<p>Supports Database Extensibility Mechanisms that create the SYSLIB database and the supporting objects within it for managing user-created DEM objects.</p> <p>DEMs include stored procedures, UDFs, and UDTs distributed as packages that can be used to extend the capabilities of Vantage.</p> <p><b>Note:</b> Successful creation of the stored procedures (SQL and external stored procedures), UDFs, and use of the created table functions requires a C compiler to be available on the Teradata system.</p>
DIPDSCSV	Enables the DATASET data type CSV storage format.
DIPERR	Loads error message files for storing the text of messages that Vantage components, software, and client connections generate.
DIPGEO	Initializes the geospatial features of the database, creating the SYSSPATIAL spatial_ref_sys database.
DIPGLOP	<p>Enables global and persistent (GLOP) data support.</p> <p>GLOP data is a type of data available to external routines. The persistence in memory of GLOP data is based on specific boundaries, such as a user role or user account.</p> <p>For more information on GLOP data, see <i>Teradata Vantage™ - SQL External Routine Programming</i>, B035-1147.</p> <p><b>Note:</b> DIPGLOP is available only with appropriate licensing. If you require GLOP support, contact your Teradata representative.</p>
DIPJDBC	Support functions for the JDBC API.
DIPJSON	<p>Enables JSON functionality in Vantage.</p> <p><b>Note:</b> This script is only available after JSON has been enabled by setting a DBS control field.</p>
DIPMAPS	Creates initial system data distribution maps, in addition to procedures, tables, and views used to manage maps.
DIPMIG	Runs the following 5 scripts in preparation for complete system migrations: DIPERR, DIPDB, DIPVIEWS, DIPVIEWSV, and DIPBAR.
DIPOLH	Initializes the Online Help Syntax.
DIPOCES	Initiates the default cost estimate statistics.
DIPPATCH	Initializes standalone patch versions. Automatically executed by PUT for all maintenance and efix upgrades.
DIPPCR	Creates a centralized repository for performance data, and infrastructure to support collection and reporting of performance data.

SQL Script File	Purpose
DIPPOST	Initializes objects that have dependencies on other functions and procedures that were previously created by DIP.
DIPPWRSTNS	Populates the DBC.PasswordRestrictions table with the list of words disallowed within a password.
DIPRCO	Finalizes the initialization of the DBC tables used for reconfiguration.
DIPRSS	Adds the ResUsage tables data columns. Allows logging of system performance statistics.
DIPRTBLOP	Creates ExecR table operator, which provides in-database analytics capability to users of R. R is an open-source programming language for statistical modeling and graphics. For more information on ExecR, see <i>Teradata Vantage™ - SQL External Routine Programming</i> , B035-1147.
DIPRUM	<p>Loads Resource Usage views and macros and enables report generation from data stored in the ResUsage system tables.</p> <p><b>Note:</b> If your Teradata platform has co-existence nodes, during installation or upgrade, the identifiers for each node group should be defined in the CASE clauses of the appropriate view definitions in the diprum.bteq file.</p>
DIPSQLJ	<p>Enables Java-based external stored procedures.</p> <p><b>Note:</b> Successful creation of the external stored procedures requires that there be a C compiler on the Teradata system.</p>
DIPSTATS	<p>For more information, see Application Programming Reference.</p> <p><b>Note:</b> This script should not be run independently. It is run automatically as part of the DIPALL script.</p>
DIPSYSFE	Initializes the macros, procedures, and views that reside within the SystemFE database.
DIPSYSFNC	Enables Teradata system functions and Embedded Services Functions.
DIPSYSUIF	<p>Creates infrastructure to manage user installed files (UIFs). UIFs are text, binary, or script files (such as perl, python, or operating system shell scripts) that users can install in the database.</p> <p>Use the SCRIPT table operator to run these scripts, pass them arguments, and run operating system commands. For more information on the SCRIPT table operator, see <i>Teradata Vantage™ - SQL Operators and User-Defined Functions</i>, B035-1210.</p>
DIPSZVIEWS	<p>Creates legacy system views infrastructure for Secure Zones feature. This script must be run at the time Secure Zones is enabled, and is usually run by Teradata Services personnel.</p> <p><b>Note:</b> This script is not visible in the DIP menu unless the Secure Zones feature is enabled. Contact your Teradata representative to enable this feature.</p>

SQL Script File	Purpose
DIPSZVIEWSV	Creates Unicode system views infrastructure for Secure Zones feature. This script must be run at the time Secure Zones is enabled, and is usually run by Teradata Services personnel. <b>Note:</b> This script is not visible in the DIP menu unless the Secure Zones feature is enabled. Contact your Teradata representative to enable this feature.
DIPTDWM	Creates or upgrades the database infrastructure for workload management and alerting. <b>Note:</b> DIPTDWM can also be started from the Linux command line by running <code>/usr/tdbms/bin/tdwmdip</code> . Running it in this way provides an interactive session that allows specification of various parameters, which cannot be specified when DIPTDWM is invoked from the DIP program.
DIPUDT	Initializes the infrastructure for creating UDT objects and Period Data Type objects.
DIPVIEWS	Creates the non-Unicode compliant views and macros for access to DBC objects.
DIPVIEWSV	Creates the Unicode compliant views and macros for access to DBC objects, and populates all the comments for DBC system tables.
DIPXML	Enables XML functionality in Vantage.

## Location of DIP Scripts

DIP scripts are SQL code files with `.bteq` file extensions. To view the SQL code of a DIP script, you can inspect the script file itself. DIP scripts are located in the `/usr/tdbms/etc` directory on database system nodes:

Any output from running DIP scripts is captured to `dip n.txt` files, where *n* corresponds to the menu number of the DIP script. These output files are located in the `/var/opt/teradata/tdtemp` directory.

## Usage Notes

The DIPALL script executes all DIP scripts that appear in the menu above it. It is normally run as part of Vantage installation, so running the scripts above it individually is not normally necessary.

Scripts that appear below DIPALL in the menu are for special purposes, and should generally only be run with guidance from the Teradata Support Center.

## Example: DIP menu screen

The following example shows the DIP menu screen when the program is run interactively. Some menu choices appear only if you have licensed related Vantage optional features.

Select one of the following DIP SQL scripts to execute:

(Press the Enter key to quit)

1. DIPERR - Error Messages
2. DIPDB - System Databases
3. DIPDEM - UDF/UDT/XSP/SPL Macros
4. DIPRSS - ResUsage Tables
5. DIPVIEWS - Legacy System Views
6. DIPVIEWSV - Unicode System Views
7. DIPDBCCMT - DBC Table Comments
8. DIPACR - Access Rights
9. DIPSYFNC - System Functions
10. DIPRUM - ResUsage Views/Macros
11. DIPOLH - Online Help
12. DIPSYFE - System FE Macros
13. DIPCRASH - CrashDumps Database
14. DIPCCS - Client Character Sets
15. DIPOCES - Cost Profiles
16. DIPWRSTNS - Password Restrictions
17. DIPUDT - UDT Macros
18. DIPARR - ArrayVec Data Type
19. DIPGEO - Geospatial Data Type
20. DIPXML - XML Data Type
21. DIPJSON - JSON Data Type
22. DIPDATASET - DATASET Data Type
23. DIPDSCSV - DATASET CSV Data Type
24. DIPAEEXT - Analytics Extensions
25. DIPCAL - Calendar Tables/Views
26. DIPSQLJ - SQLJ Views/Procedures
27. DIPSYSUIF - SYSUIF Views/Procedures
28. DIPJDBC - JDBC Functions
29. DIPRCO - Reconfig
30. DIPSTATS - Automated Stats Mgmt
31. DIPBAR - BAR Databases/Views
32. DIPPOST - Post commands
33. DIPSZVIEWS - Secure Zones Legacy System Views
34. DIPSZVIEWSV - Secure Zones Unicode System Views
35. DIPMAPS - Multiple Hash Maps
36. DIPTDWM - TDWM Configuration
37. DIPANSI - ANSI Functions
38. DIPALL - All of the above
39. DIPMIG - Scripts for Migration
40. DIPDCR - PDCR Tables/Views



- 41. DIPACC - Access Logging
- 42. DIPPATCH - Stand-alone patches
- 43. DIPGLOP - GLOP Tables/Procedures
- 44. DIPRTBLOP - R Table Operator
- 45. DIPCDC - Unity Change Data Capture

# Database Window (xdbw)

---

**Note:**

The utility described is not available for as-a-service customers.

---

Database Window, xdbw (abbreviated throughout the documentation as DBW), allows Teradata system administrators to control the operation of Advanced SQL Engine.

## Runs From

DBW runs from:

- Linux systems, which generally include X server software. Type xdbw to start DBW.
- Windows systems, such as those that may be attached to a Teradata system over a secure SSH connection. Windows clients generally require separate X server software to be installed.

For general information on starting the utilities from different interfaces, see [Starting the Utilities](#).

## Running DBW

You can run DBW from the following locations:

- System Console
- Remote workstation or computer

---

**Note:**

In order to use DBW, you must either be a member of the tdtrusted user group, or have been explicitly granted the right to run console utilities with the CNS GRANT command.

---

## Related Information

For instructions on how to...	See...
use the GRANT command	<a href="#">GRANT</a> .
run DBW	<a href="#">Starting the Utilities</a> .
use database commands	<a href="#">Commands</a> .
start DBW utilities	<a href="#">START</a> .

## DBW Main Window

The following table describes the three main areas of the DBW main window.

This Area ...	Contains ...
Menu Bar	two pull-down menus: <ul style="list-style-type: none"> <li>• File</li> <li>• Help</li> </ul>
Status line	<ul style="list-style-type: none"> <li>• the connection identifier. This consists of the connection number by which this DBW console is known, followed by a slash and the total number of DBW consoles connected to the system.</li> <li>• the name of the node to which this DBW is connected.</li> <li>• the current status of the system.</li> </ul> <p><b>Note:</b> When the PDE is running, the Supervisor Status line shows the status as <b>Reading</b> to allow you to enter commands into the Supervisor window.</p>
Buttons	<p>the following subwindows:</p> <ul style="list-style-type: none"> <li>• Four DBW application windows. In each DBW application window, you can run one database utility or program at a time.</li> <li>• DBS I/O window. This window contains messages from database programs that are not running in DBW application windows.</li> <li>• Supervisor window. In the Supervisor window, you can run commands and execute utilities.</li> </ul> <p><b>Note:</b> Database commands run in the Supervisor window. Database utilities run in one of the DBW application windows.</p> <p>These subwindows can communicate simultaneously with the database.</p>

## Granting CNS Permissions

To run DBW on any node, you must be a member of the Linux tdtrusted user group or logged in as root. The GRANT command can be used to allow CNS access to non-tdtrusted users. Non-tdtrusted users must have permissions granted to them on the Control Node. Subsequently, they can run DBW from any node.

To grant all permissions to a non-tdtrusted user in DBW, type:

```
grant xxx @localhost all
```

where xxx is the user ID.

---

### Note:

The permissions should be granted or revoked only when all TPA nodes are running.

---

## Related Information

For information on the three commands that manage CNS permissions, see:

- [GET PERMISSIONS](#)
- [GRANT](#)
- [REVOKE](#)

For information on the DBW commands that are always accepted if CNS accepts the connection, see:

- [Accessing Help on DBW](#)
- [CNSGET](#)
- [GET ACTIVELOGTABLE](#)
- [GET CONFIG](#)
- [GET LOGTABLE](#)
- [GET PERMISSIONS](#)
- [GET RESOURCE](#)
- [GET SUMLOGTABLE](#)
- [GET TIME](#)
- [GET VERSION](#)

## Repeating Commands

The following table describes how to repeat database commands.

IF you ...	THEN the command...
double-click on the command in the command history area of the Supervisor window	re-executes immediately.
single-click on the command	is placed on the command input line. You can edit the command before pressing <b>Enter</b> .

## Saving Output

There are two methods for saving output in DBW:

- [Logging Information to Files](#)
- [Saving the Window Buffer](#)

### Logging Information to Files

This method maintains an ongoing log of information from the output display area.

When logging is enabled, the session from the current window is saved to a default or specified text-format log file. These logs can be useful for reviewing or printing out the session.

When logging all windows, standard log files are opened. If a log file already exists, DBW appends the current session text to the existing file.

To start a log file for all windows, click **File > Log All Windows** from the DBW main window.

DBW immediately begins logging output from all DBW subwindows into separate files. If log files already exist for each of the subwindows, new text is appended to these files.

To start a log file for a single application window:

1. Click **File > Logging** from one of the active application windows.
2. In the **Select Log File For Window** dialog box, click the log file or specify a new file name, and click **Okay** or **Save**.

DBW immediately begins logging the text in the output display area to the specified log file. If the log file already exists, the new text is appended to it.

## Saving the Window Buffer

This method saves only the current contents of the window buffer to a file. Any output created after the window is saved is lost.

To save current contents of an application or Supervisor window:

1. Click **File > Save Buffer** from the active subwindow.
2. In the **Save Buffer** dialog box, click the log file or specify a new file name, and click **Okay** or **Save**.

The text in the output display area of the application window is saved in the specified file. If the file already exists, it is overwritten.

## Viewing Log and Buffer Files

1. Click **File > View** from the DBW main window.
2. In the **Select File to View** dialog box, click the log file or specify a new file name, and click **Okay** or **Open**.
3. In the **Viewing File** window, click **Okay** to exit.

# DBW Scripts

## Creating Scripts

To save time, you can create scripts of frequently used commands in any DBW application window. When you run the scripts, DBW treats these scripts as if you typed the commands on the console.

You can create a script file with any text editor, such as vi or Emacs. The file should contain a list of commands with one command on each line.

## Running Scripts in DBW

1. Click **File > Send Script File** from an active application window.
2. In the **Send Script** dialog box, click the log file or specify a new file name, and click **Okay** or **Open**.

## Accessing Help on DBW

- Click the **Help** menu from any window.
- Press **F1** from any window (Linux only).
- Click the **Help** button in dialog boxes.

## Commands

## ABORT SESSION

Aborts any outstanding request or transaction of one or more sessions, and optionally logs those sessions off the Teradata system.

### Syntax

```
ABORT SESSION {
  hostid:sessionno |
  hostid.username |
  *.username |
  hostid.* |
  *.*
} [ LOGOFF ] [ LIST ] [ OVERRIDE ]
```

### Syntax Elements

#### *hostid*

logical ID of a host (or client) with sessions logged on. A *hostid* of 0 identifies internal sessions or system console sessions. The range of values are 0 to 32767.

#### *sessionno*

session number. *sessionno* combined with *hostid*, represents a unique session ID. The range of values is 0 to 4,294,967,295.

#### *username*

user who is running the session. *username* can have a maximum length of 30 characters.

#### LOGOFF

indicator of whether or not to log the requested session off Vantage in addition to aborting them.

#### LIST

indicator of whether or not to display a list of sessions that meet criteria.

#### OVERRIDE

indicator of whether or not to override an ABORT SESSION failure.

### Usage Notes

Aborting a session is useful when a session causes a production job to block other sessions waiting for locks, or when a session takes up so many resources that a critical application runs too slowly.

If an identified session cannot be aborted, the following occurs:

- A 3268 error message returns.
- Additional information is logged to the error log.
- Processing of the request is terminated.

Retry the ABORT SESSION command at a later time after queued up abort requests complete.

You can issue the ABORT SESSION command on both mainframe-attached and workstation-attached clients.

### **Related Information**

For more information about ABORT SESSION, see *Teradata Vantage™ - Application Programming Reference*, B035-1090.

For a complete description of error 3268, see *Teradata Vantage™ - Database Messages*, B035-1096.



## CNSGET

Returns parameters used by CNS to control the connections to console windows.

### Syntax

```
CNSGET
```

### Usage Notes

To set the value for ...	Which is the ...	Use the command ...
mailbox timeout	interval of time in seconds between the time you type a request and the time CNS rejects the request because a program did not solicit the input	<a href="#">CNSSET TIMEOUT.</a>
system state polling interval	interval of time in seconds between checking the database state and substate	<a href="#">CNSSET STATEPOLL.</a>
number of lines to be queued	number of historical output lines that are sent to newly connected CNS connections	<a href="#">CNSSET LINES.</a>

### Example: Showing the current CNS parameters

The following example shows the current CNS parameters.

```
cnsget

mailbox timeout is 5.
system state polling interval is 2.
number of lines to be queued is 9.
```

## CNSSET LINES

Sets the number of lines that are saved from the output display area and sent to DBW after a reconnect with CNS.

### Syntax

```
CNSGET LINES n
```

### Syntax Elements

*n*

number of lines to be saved.

Valid values range from 2 to 72 lines.

The default is 10 lines.

### Usage Notes

Teradata recommends setting large values (for example, 10 or more lines), unless the display of old input and output after a reconnect is a security concern.

To check the current number of lines set, issue the CNSGET command. For more information, see [CNSGET](#).

### Example: Setting the number of lines to be saved from the DBW display

In this example, the number of lines set to be queued on the system is 12.

```
cnsset lines 12
```

```
CNSSUPV: cns will save 12 lines per screen.
```

## CNSSET STATEPOLL

Sets how often CNS checks the state of the DBS.

### Syntax

```
CNSSET STATEPOLL n
```

### Syntax Elements

*n*

Time interval, in seconds, between state checks.

Valid intervals range from 1 to 10 seconds.

The default is 2 seconds.

### Usage Notes

The states are displayed in the status line of the main DBW window.

Values greater than 2 seconds reduce system overhead, but increase response time to state changes.

To check the current setting, use the CNSGET command. For more information, see [CNSGET](#).

### Example: Setting the CNS polling rate

In this example, the rate at which CNS checks the DBS state is set to 4 seconds.

```
cnsset statepoll 4
```

```
CNSSUPV: system state polling interval is 4 seconds.
```

## CNSSET TIMEOUT

Sets the interval between the time you type a request and the time CNS rejects the request because a program did not respond to the input.

### Syntax

```
CNSSET TIMEOUT n
```

### Syntax Elements

*n*

Time interval, in seconds, between receipt of input and receipt of error message, if a utility is not reading input.

Valid intervals range from 1 to 60 seconds.

The default is 5 seconds.

### Usage Notes

The key-in timeout signals that a utility is not reading any input.

Values less than 5 seconds can cause input to be lost too quickly. Values greater than 5 seconds could cause a delay in delivery of error messages.

To check the current setting, issue the CNSGET command. For more information, see [CNSGET](#).

### Example: Setting the CNS timeout

In this example, the current timeout value is set to 5 seconds.

```
cnsset timeout 5
```

```
CNSSUPV: timeout for input request is 5 seconds.
```

## DISABLE LOGONS/ DISABLE ALL LOGONS

Prevents users from logging on to the system until the ENABLE LOGONS or ENABLE ALL LOGONS command is sent.

### Syntax

```
DISABLE [ALL] LOGONS
```

### Usage Notes

DISABLE LOGONS and DISABLE ALL LOGONS are synonymous.

To use this command, the current logon state must be one of the following:

- Logons are enabled - The system is quiescent.
- Only user DBC logons are enabled - The system is quiescent.

To see the current logon state, use one of the following commands:

- from DBW: QUERY STATE
- from a Teradata Command Prompt: pdestate -a

Vantage must be running for the change to be applied. A database restart is not required for the new setting to take effect.

Users logged on to the system before the DISABLE LOGONS or DISABLE ALL LOGONS command is executed remain logged on, and their jobs continue to run. All sessions are reconnected if the system is restarted.

Both commands (DISABLE LOGONS or DISABLE ALL LOGONS) update the Start With Logons value in the PDE Control GDO, which is viewable using the Control GDO Editor (ctl) utility. For more information, see [Control GDO Editor \(ctl\)](#).

### Example: Disabling user logons to the database

The following example shows the current logon status is disabled for all new users.

```
disable logons
```

```
12/02/12 10:05:02 Logons disabled.
```

## ENABLE DBC LOGONS

Allows only new DBC users to log on to the system.

### Syntax

```
ENABLE DBC LOGONS
```

### Usage Notes

To use this command, the current logon state must be one of the following:

- Logons are enabled - The system is quiescent.
- Logons are disabled - The system is quiescent.

To see the current logon state, use one of the following commands:

- from DBW: QUERY STATE
- from a Teradata Command Prompt: pdestate -a

Vantage must be running for the change to be applied. A database restart is not required for the new setting to take effect.

New non-DBC users cannot log on to the system after the ENABLE DBC LOGONS command is executed.

Users, other than the DBC, logged on to the system before the ENABLE DBC LOGONS command is executed remain logged on, and their jobs continue to run. All sessions are reconnected if the system is restarted.

The ENABLE DBC LOGONS command update the Start With Logons value in the PDE Control GDO, which is viewable using the Control GDO Editor (ctl) utility. For more information, see [Control GDO Editor \(ctl\)](#).

### Example: Enabling DBC logons

The following example shows the current logon status is enabled for DBC users only. Non-DBC users cannot log on to the current system.

```
enable dbc logons
```

```
12/02/12 10:05:02 DBC Logons enabled.
```

## ENABLE LOCAL DBC LOGONS

Allows only local DBC users to log on to the system.

### Syntax

```
ENABLE LOCAL DBC LOGONS
```

## Usage Notes

To use this command, the current logon state must be one of the following:

- Logons are enabled - The system is quiescent.
- Logons are disabled - The system is quiescent.

To see the current logon state, use one of the following commands:

- from DBW: `QUERY STATE`
- from a Teradata Command Prompt: `pdestate -a`

Vantage must be running for the change to be applied. A database restart is not required for the new setting to take effect.

New non-DBC users and non-local DBC users cannot log on to the system after the `ENABLE LOCAL DBC LOGONS` command is executed.

Users, other than the DBC, logged on to the system before the `ENABLE LOCAL DBC LOGONS` command is executed remain logged on, and their jobs continue to run. All sessions are reconnected if the system is restarted.

The `ENABLE LOCAL DBC LOGONS` command updates the Start With Logons value in the PDE Control GDO, which is viewable using the Control GDO Editor (ctl) utility. For more information, see [Control GDO Editor \(ctl\)](#).

## ENABLE LOGONS/ ENABLE ALL LOGONS

Allows any new users to log on to the system.

### Syntax

```
ENABLE [ALL] LOGONS
```

### Usage Notes

ENABLE LOGONS and ENABLE ALL LOGONS are synonymous.

To use this command, the current logon state must be one of the following:

- Logons are disabled - The system is quiescent.
- Only user DBC logons are enabled - The system is quiescent.

To see the current logon state, use one of the following commands:

- from DBW: QUERY STATE
- from a Teradata Command Prompt: pdestate -a

Vantage must be running for the change to be applied. A database restart is not required for the new setting to take effect.

Users logged on to the system before ENABLE LOGONS or ENABLE ALL LOGONS is executed remain logged on, and their jobs continue to run. All sessions are reconnected if the system is restarted.

Both commands (ENABLE LOGONS or ENABLE ALL LOGONS) update the Start With Logons value in the PDE Control GDO, which is viewable using the Control GDO Editor (ctl) utility. For more information, see [Control GDO Editor \(ctl\)](#).

### Example: Enabling all logons

The following example shows the current logon status is enabled for new users.

```
enable logons

12/02/12 10:04:29 Logons enabled.
```



## GET ACTIVELOGTABLE

Shows the active row filtering status on any ResUsage table.

### Syntax

```
GET ACTIVELOGTABLE { tablename | ALL }
```

### Syntax Elements

#### *tablename*

name of the ResUsage table for which to show the active row filtering status.

ResUsage Table Name	Description
spma	Contains system-wide node information, intended primarily for Teradata engineers. This table is generally not used at customer sites.
ipma	Contains system-wide node information, intended primarily for Teradata engineers. This table is generally not used at customer sites.
svpr	Contains data specific to each virtual processor and its file system.
ivpr	Contains system-wide virtual processor information, intended primarily for Teradata engineers. This table is generally not used at customer sites.
scpu	Contains statistics on the CPUs within the nodes.
sldv	Contains system-wide, storage device statistics.
spdisk	Contains pdisk I/O, cylinder allocation, and migration statistics.
svdisk	Contains statistics collected from the associated storage of the AMP.
sawt	Contains data specific to the AMP worker tasks (AWTs).
sps	Contains resource usage data about workloads.
shst	Contains statistics on the host channels and LANs that communicate with Vantage.
smhm	Contains MAPS statistics.

#### ALL

the active row filtering status of all ResUsage tables on the system.

### Usage Notes

To enable or disable active row filtering, see [SET ACTIVELOGTABLE](#).

For information about active row filtering and its effect on ResUsage tables, see *Teradata Vantage™ - Resource Usage Macros and Tables*, B035-1099.

### **Example: Showing the active row filtering status for ResUsage tables**

The following example shows the active row filtering status of all ResUsage tables on the current system. The system displays a screen similar to this:

```
get activelogtable all

spma table's active row filtering mode is disabled
ipma table's active row filtering mode is disabled
svpr table's active row filtering mode is disabled
ivpr table's active row filtering mode is disabled
scpu table's active row filtering mode is disabled
sldv table's active row filtering mode is disabled
spdisk table's active row filtering mode is disabled
svdisk table's active row filtering mode is disabled
sawt table's active row filtering mode is disabled
sps table's active row filtering mode is enabled
shst table's active row filtering mode is disabled
smhm table's active row filtering mode is disabled
```

## GET CONFIG

Returns the current system configuration.

### Syntax

```
GET CONFIG
```

### Usage Notes

The following table defines the system configuration.

Component	Description
Vproc Number	Uniquely identifies the vproc across the entire system.
Rel. Vproc#	Represents the number of the vproc relative to the node upon which the vproc resides.
Node ID	Identifies the node upon which the vproc resides. Node ID is formatted as <i>CCC-MM</i> , where <i>CCC</i> denotes the cabinet number and <i>MM</i> denotes the module number. For example, the node ID 003-07 identifies the node in module number 7 of cabinet 3.
Movable	Indicates whether the vproc can be migrated to another node in its defined clique if its primary node fails.
Crash Count	Represents the number of times the vproc has crashed. The count increments with every attempted system restart. When the system restart succeeds, Crash Count is reset to 0 for all vprocs.
Vproc State	<p>Represents the current PDE system state of a vproc.</p> <ul style="list-style-type: none"> <li><b>FATAL:</b> Indicates a serious problem with a vproc and/or its associated storage. For example, if a vproc crashes repeatedly, the number of crashes might have exceeded the allowable crash count, or if there are corrupt tables on the vdisk that require a Table Rebuild of some (or all) of the tables, a vproc is set to the FATAL state.           <p><b>Note:</b> When a TVS vproc is in FATAL state, all AMPs associated with it will be put into FATAL state at the next database restart.</p> </li> <li><b>FATAL **:</b> Indicates an AMP cannot access its storage.           <p><b>Note:</b> The AMP partition is not started for a vproc in this state.</p> </li> <li><b>NEWPROC:</b> Applies only to AMP and PE vprocs. It indicates that either a new vproc is added into the Vantage configuration or an existing vproc is deleted. A vproc with a status of NEWPROC is a member of the PDE message groups but is not a member of the operational database message group. For more information, contact the Teradata Support Center.</li> <li><b>NONODE:</b> Indicates that the physical hardware required to run this vproc is not available. This state is not accepted as an argument to the SET command, although this state might appear in the Vproc Status Table produced by the STATUS command.           <p><b>Note:</b> The AMP or PE partitions are not started for a vproc in this state.</p> </li> </ul>

Component	Description
	<ul style="list-style-type: none"> <li>• <b>NULL:</b> Undefined. It is not accepted as an argument to the SET command, although this state might appear in the Vproc Status Table produced by the STATUS command.</li> <li>• <b>OFFLINE:</b> Generally, indicates vprocs that are not fully operational and have been forced down by the DBA or system administrator. A vproc with status OFFLINE is a member of the PDE message group but is not a member of the operational database message group. For more information, contact the Teradata Support Center. When a TVS vproc is in OFFLINE state, all AMPs associated with it will be put into FATAL state.</li> <li>• <b>ONLINE:</b> Indicates that the vproc is fully operational and actively participating with the database. A vproc with status ONLINE is a member of both the PDE and database message group. For more information, contact the Teradata Support Center.</li> <li>• <b>UTILITY:</b> Is transitional and is used by database recovery, reconfiguration, and the Table Rebuild utility to indicate that a previously OFFLINE/FATAL/NEWPROC is interacting with the online system. This vproc is a member of the PDE message groups but not a member of the operational database message groups. A local crash of a UTILITY vproc causes a database restart.</li> </ul>
Config Status	<p>Displays the Vantage Logical Configuration Map Status of a vproc.</p> <ul style="list-style-type: none"> <li>• <b>Online:</b> The vproc is fully operational. This status usually corresponds to ONLINE Vproc State.</li> <li>• <b>Down:</b> The vproc has been forced down. This status usually corresponds to the OFFLINE, UTILITY, FATAL, and NONODE Vproc states.</li> <li>• <b>Catchup:</b> The Config Status for the AMP was Down and is being recovered in the background. If System RestartKind is COLDWAIT, the Config Status of the vproc becomes Online when recovery is complete. This status usually corresponds to the UTILITY Vproc State.</li> <li>• <b>Hold:</b> The Config Status for the AMP was Catchup or Down and its data is being recovered. The Config Status of this vproc becomes Online when recovery is complete. This status usually corresponds to the ONLINE Vproc State.</li> <li>• <b>NewReady:</b> This is either a newly added vproc or one that has been removed from the Vantage Logical Configuration. This status usually corresponds to the UTILITY or NEWPROC Vproc State.</li> <li>• <b>NewDown:</b> A newly added vproc that is down. It has not yet been added into the DBS configuration. This status usually corresponds to the OFFLINE Vproc State.</li> <li>• <b>Null:</b> This vproc is not yet in the Vantage Logical Configuration. The vproc might have been added to the PDE Physical Configuration or deleted from the Vantage Logical Configuration, and System Startup has not run. Startup notes that this vproc does not appear in the Startup configuration map and changes the Config Status of the vproc to NewReady. This status usually corresponds to the NEWPROC Vproc State.</li> </ul>
Config Type	Represents the Vantage Logical Configuration Map Type of a vproc.
Cluster/Host No.	Displays the Cluster Number if the Config Type is AMP or displays the Host No. if the Config Type is PE.

Component	Description
	<p>Cluster is the cluster number for the AMP as defined during a system configuration or reconfiguration. The valid range of cluster numbers is 0 to 8099.</p> <p>Host No. is the host number that was assigned to the PE using during a system configuration or reconfiguration. The valid range of host numbers is 1 to 1023.</p>
RcvJrnl/Host Type	<p>Displays the RcvJrnl (that is, Recovery Journaling) flag if the Config Type is AMP or the Host Type if the Config Type is PE.</p> <p>The RcvJrnl flag is Off if an AMP is down and the other AMPs in its cluster are not to create a recovery journal for the down AMP.</p> <p><b>Note:</b></p> <p>If you anticipate that an AMP will be down for a long period of time, Teradata recommends an offline rebuild of all tables on the AMP (after the RcvJrnl flag has been set to Off).</p> <p>The Host Type is the host type for the PE as defined during a system configuration or reconfiguration and can be IBM, COP, ATT3B, BULLHN, or OS1100.</p>
TVS Vproc	For AMP vprocs, displays the associated TVS vproc.

The following table defines the node configuration.

Component	Description
Node ID	Defined using PUT. For more information, see <i>Parallel Upgrade Tool (PUT) Reference</i> . The node ID is formatted as CCC-MM, where CCC denotes the cabinet number and MM denotes the module number. For example, the node ID 003-07 identifies the node in module number 7 of cabinet 3.
Node State	The current state of the node, which is either ONLINE, DOWN, or STANDBY.
Clique Number	The clique number for the node as defined using PUT. For more information, see <i>Parallel Upgrade Tool (PUT) Reference</i> .
CPUs	The number of CPUs on the node.
Memory (MB)	The total memory size in megabytes for the node (rounded up to the nearest integer).
CHANs	The number of channels attached to the node.
LANs	The number of LANs attached to the node.
AMPs	The number of AMPs running on the node.
Node Name	The network name for the node.

When applicable, a footnote follows the PDE Status Table, indicating which node is defined as a Non-TPA Node (for nodes in STANDBY).

### Example: Displaying the current system configuration

The following example shows the current system configuration.

SYSTEM NAME: qpdt23

11/06/12 17:09:05

## DBS LOGICAL CONFIGURATION

Vproc Number	Rel. Vproc#	Node ID	Can Move	Crash Count	Vproc State	Config Status	Config Type	Cluster/ Host No.	Rcv Jrnl/ Host Type	TVS Vproc
0*	1	1-01	Yes	0	ONLINE	Online	AMP	0	On	10238
1	2	1-01	Yes	0	ONLINE	Online	AMP	0	On	10237
8192	5	1-01	No	0	ONLINE	N/A	GTW	1	COP	N/A
10237	6	1-01	Yes	0	ONLINE	N/A	TVS	0	N/A	N/A
10238	7	1-01	Yes	0	ONLINE	N/A	TVS	0	N/A	N/A
16382	3	1-01	Yes	0	ONLINE	Online	PE	1	COP	N/A
16383	4	1-01	Yes	0	ONLINE	Online	PE	1	COP	N/A

\* DBS Control AMP

DBS State: Logons are enabled - The system is quiescent

DBS RestartKind: COLD

## PDE PHYSICAL CONFIGURATION

Node ID	Node State	Clique Number	Memory CPUs	(MB)	CHANs	LANs	AMPs	Node Name
1-01	ONLINE	0	1	1337	0	1	2	localhost

PDE State: RUN/STARTED

## GET EXTAUTH

Returns the current value of external authentication.

### Syntax

```
GET EXTAUTH
```

### Usage Notes

To set External Authentication, see [SET EXTAUTH](#). For additional information on external authentication, see *Teradata Vantage™ - Advanced SQL Engine Security Administration*, B035-1100.

### Example: Displaying whether the system is set for external authentication

The following example shows external authentication logons are rejected.

```
get extauth
```

```
The External Authentication mode is set to OFF
```

## GET LOGTABLE

Returns the logging status on any ResUsage table.

### Syntax

```
GET LOGTABLE { tablename | ALL }
```

### Syntax Elements

#### *tablename*

name of the ResUsage table for which to display logging status. ResUsage table names are listed in the following table.

ResUsage Table Name	Description
spma	Contains system-wide node information, intended primarily for Teradata engineers. This table is generally not used at customer sites.
ipma	Contains system-wide node information, intended primarily for Teradata engineers. This table is generally not used at customer sites.
svpr	Contains data specific to each virtual processor and its file system.
ivpr	Contains system-wide virtual processor information, intended primarily for Teradata engineers. This table is generally not used at customer sites.
scpu	Contains statistics on the CPUs within the nodes.
sldv	Contains system-wide, storage device statistics.
spdisk	Contains pdisk I/O, cylinder allocation, and migration statistics.
svdisk	Contains statistics collected from the associated storage of the AMP.
sawt	Contains data specific to the AMP worker tasks (AWTs).
sps	Contains resource usage data about workloads.
shst	Contains statistics on the host channels and LANs that communicate with Vantage.
smhm	Contains MAPS statistics.

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

#### ALL

displays the logging status of all ResUsage tables.



## Usage Notes

You can also view the logging status of the ResUsage tables by issuing the SCREEN RSS command in the Control GDO Editor (ctl) utility. For more information, see [Control GDO Editor \(ctl\)](#).

To enable or disable logging on any ResUsage table, see [SET LOGTABLE](#).

### Example: Displaying the current logging status for ResUsage tables

The following example displays the logging status of all ResUsage tables. The system displays a screen similar to this:

```
get logtable all

spma table's logging is disabled
ipma table's logging is disabled
svpr table's logging is disabled
ivpr table's logging is disabled
scpu table's logging is disabled
sldv table's logging is disabled
spdisk table's logging is disabled
svdisk table's logging is disabled
sawt table's logging is disabled
sps table's logging is disabled
shst table's logging is disabled
smhm table's logging is disabled
```

## GET PERMISSIONS

Returns CNS permissions for the specified connection-ID.

### Syntax

```
GET PERMISSIONS username@host
```

### Syntax Elements

#### *username*

logon user name accessing CNS for whom you want to get or view permissions.

#### *host*

network host name of the computer to which the user is logged on.

### Usage Notes

DBW shows the name of the connection-ID followed by its list of permissions and any granted or revoked utilities.

If the connection-ID does not exist, DBW returns the following message:

```
user@host: no permissions
```

### Example: Displaying the CNS permissions for a specified user

The following example shows all CNS permissions granted to connection-ID jsmith@isdn1954.

```
get permissions jsmith@isdn1954

jsmith@isdn1954: set logons log restart grant abort interactive
                  Revoked Utilities: vprocmanager
```

## GET RESOURCE

Returns the RSS collection and logging rates for the ResUsage tables.

### Syntax

```
GET RESOURCE
```

### Usage Notes

The collection period is used to extract the data for real-time tools to display and the logging period is used for writing the data to the database.

The default for PMPC Resource Collection Rate, Node Logging Rate is 600. A logging or collection rate of 0 disables that operation.

For details about logging rate of resource utilization, see *Teradata Vantage™ - Resource Usage Macros and Tables*, B035-1099.

---

### Note:

You can also view the current logging and collection rates by issuing the SCREEN RSS command in the Control GDO Editor (ctl) utility. For more information, see [Control GDO Editor \(ctl\)](#).

---

### Example: Displaying the current RSS collection and logging rates

The following example shows the RSS collection and logging rates for the ResUsage tables on the current system.

```
get resource
```

```
RSS Rate Information:
```

```
PMPC Resource Collection Rate = 600
```

```
Node Logging Rate      = 600
```

## GET SUMLOGTABLE

Returns the Summary Mode status of ResUsage tables.

### Syntax

```
GET SUMLOGTABLE tablename
```

### Syntax Elements

#### *tablename*

name of the ResUsage table for which to display the Summary Mode status.

ResUsage Table Name	Description
spma	Contains system-wide node information, intended primarily for Teradata engineers. This table is generally not used at customer sites.
ipma	Contains system-wide node information, intended primarily for Teradata engineers. This table is generally not used at customer sites.
svpr	Contains data specific to each virtual processor and its file system.
ivpr	Contains system-wide virtual processor information, intended primarily for Teradata engineers. This table is generally not used at customer sites.
scpu	Contains statistics on the CPUs within the nodes.
sldv	Contains system-wide, storage device statistics.
spdisk	Contains pdisk I/O, cylinder allocation, and migration statistics.
svdisk	Contains statistics collected from the associated storage of the AMP.
sawt	Contains data specific to the AMP worker tasks (AWTs).
sps	Contains resource usage data about workloads.
shst	Contains statistics on the host channels and LANs that communicate with Vantage.
smhm	Contains MAPS statistics.

### Usage Notes

You can display the Summary Mode status for all tables except spma and ipma because these ResUsage tables only report a single row of data for each node in normal mode.

For details on how summary mode affects data reporting for these specific tables, see the table descriptions in *Teradata Vantage™ - Resource Usage Macros and Tables*, B035-1099.

You can also view the status of ResUsage tables logged in Summary Mode from the RSS Screen of the Control GDO Editor (ctl) utility. For more information, see [Control GDO Editor \(ctl\)](#).

To enable or disable logging in Summary Mode on any ResUsage table, see [SET SUMLOGTABLE](#).

### **Example: Displaying the Summary Mode status of ResUsage tables**

The following example shows the Summary Mode status of the Svpr table.

```
get sumlogtable svpr  
  
svpr table's summary mode is disabled
```

## GET TIME

Returns the current date and time on the system.

### Syntax

```
GET TIME
```

### Example: Displaying the current system date and time

The following example shows the current date and time.

```
get time
```

```
The system time is Tue Jul 18 15:30:01 2006
```

## GET VERSION

Returns the current running PDE and DBS version numbers on the system.

### Syntax

```
GET VERSION
```

### Usage Notes

You can also view the versions of the current running PDE and DBS software from the Version screen of the Control GDO Editor (ctl) utility. For more information, and a complete description of the different software versions see [Control GDO Editor \(ctl\)](#).

### Example: Displaying the running PDE and DBS versions

The following example shows the current running PDE and DBS version numbers.

```
get version

The currently running PDE  version: 14.10.00.00
The currently running DBS  version: 14.10.00.00
The currently running TGTW version: 14.10.00.00
The currently running TCHN version: 14.10.00.00
The currently running TDGSS version: 14.10.00.00
The currently running PDEGPL version: 14.10.00.00
```

## GRANT

Grants CNS permissions to a user who is not a member of the tdtrusted user group.

### Syntax

```
GRANT username@host permission_list
```

### Syntax Elements

#### *username*

logon name of the user accessing CNS for whom you want to grant permissions.

#### *host*

network host name of the computer to which you are logged on.

#### *permission\_list*

list of supervisor privileges to be granted, separated by spaces. The following table lists the permissions.

Permission	Grants User Authority to Issue
abort	ABORT SESSION command
all	All CNS commands
grant	GRANT and REVOKE commands
interactive	Input to interactive programs Separate listed programs by a space and enclose the list in parentheses. For example: (dbscontrol vprocmanager)
logons	<ul style="list-style-type: none"> <li>• ENABLE LOGONS/ENABLE ALL LOGONS</li> <li>• ENABLE DBC LOGONS</li> <li>• DISABLE LOGONS/DISABLE ALL LOGONS</li> </ul>
log	LOG command
restart	RESTART command
set	CNSSET and SET commands
start	START and STOP commands



## Usage Notes

The GRANT command only affects future DBW sessions. You must stop and restart the DBW session for the updated permissions to take effect.

If interactive permission has been granted, it is added to the GRANT list. If interactive permission has been revoked, then it is removed from the GRANT list if it exists there. To revoke interactive permissions, see [REVOKE](#).

## Example: Granting CNS permissions

In this example, connection-ID jsmith@isdn1954 is granted interactive privileges to the Vproc Manager utility.

```
grant jsmith@isdn1954 interactive(vprocmanager)

CNSSUPV: permissions for jsmith@isdn1954 updated
jsmith@isdn1954: set logons log restart grant abort interactive
                Granted Utilities: vprocmanager
```

## LOG

Logs text to the database event log table (DBC.EventLog) and the message event log file for the current day.

### Syntax

```
LOG errorlogtext
```

### Syntax Elements

#### *errorlogtext*

text to be sent to the DBC.EventLog table.

### Example: Logging specified text to the event log table and messages log file

In this example, the text “Vproc 1022 added as a parsing engine” is saved to the DBC.EventLog table and log file.

```
log Vproc 1022 added as a parsing engine.
```

## QUERY STATE

Returns the operational status of the system.

### Syntax

```
QUERY STATE
```

### Usage Notes

The following table lists the valid system states.

System State	Description
Database is not running	System has not been started; it cannot be accessed from a client or used for processing.
Database Startup	System is undergoing startup processing and is not yet ready to accept requests.
Logons are disabled - Users are logged on	No new sessions can log on, but existing sessions are still logged on.
Logons are disabled - The system is quiescent	Logons are disabled and no sessions are logged on.
Logons are enabled - Users are logged on	New sessions can log on and work is in process.
Logons are enabled - The system is quiescent	Logons are enabled, but no sessions are logged on.
Only user DBC Logons are enabled	Only new DBC sessions can log on and work is in process.
RECONFIG is running	The system is being reconfigured.
System is operational without PEs - Sessions are not allowed	Either there are no PEs configured into the system or all PEs are offline/down.
TABLEINIT is running	System startup has detected that there are no tables on the system and is running TABLEINIT to create the system tables. This usually occurs during the next system restart after a system initialization.

Following are valid system substates of System Startup:

- Initializing Database VProcs
- Initializing Database Configuration
- Starting AMP Partitions
- Starting PE Partitions
- Voting for Transaction Recovery
- Starting Transaction Recovery

- Recovering Down AMPs
- Recovering Database Partitions

**Example: Displaying the operational status of the system**

This example shows the current operational status of the system.

```
query state
```

```
TPA is in state: Logons are enabled - The system is quiescent
```

## RESTART TPA

Restarts or stops the database after optionally performing a dump.

When you restart the database, all vprocs are reset.

---

### Note:

RESTART TPA does not prompt you to confirm the restart.

---

### Syntax

```
RESTART TPA
  [ NODUMP | DUMP = { YES | NO } ]
  [ COLD | COLDWAIT ] comment
```

### Syntax Elements

#### NODUMP

Restart the database without a dump. This is the default.

#### DUMP

Restart the database with a dump.

- If DUMP is set to YES, then the database restarts with a dump.
- If DUMP is set to NO, then the database restarts without a dump.

#### COLD

Restart the database without waiting for transaction recovery. This is the default.

#### COLDWAIT

Finish all database transaction recovery before allowing users to log on.

#### *comment*

The reason for the restart.

### Example: Restarting the database

In this example, a cold restart is issued with a message stating the reason why.

```
restart tpa cold Scheduled maintenance.
```

```
The reason for the restart is logged as:
Scheduled maintenance.
```



## REVOKE

Revokes CNS permissions from a user who is not a member of the tdtrusted user group. Members of tdtrusted implicitly have a non-revocable CNS permissions.

### Syntax

```
REVOKE username@host permission_list
```

### Syntax Elements

#### *username*

logon name of the user accessing CNS for whom you want to revoke permissions.

#### *host*

network host name of the computer on which that user is logged on.

#### *permission\_list*

list of supervisor privileges to be revoked, separated by spaces. For a list of the available privileges, see [GRANT](#).

### Usage Notes

The REVOKE command only affects future DBW sessions. You must stop the DBW session and then restart it for the updated permissions to take effect.

If interactive permission is revoked, then it is removed from the GRANT list if it exists there. If interactive permission has been granted, it is added to the GRANT list. To grant interactive permissions, see [GRANT](#).

### Example: Revoking a CNS privilege from a specified user

In this example, the following CNS permissions: START, STOP, and LOGONS, and interactive utilities: DBS Control and Vproc Manager, are revoked from connection-ID jsmith@isdn1954.

```
revoke jsmith@isdn1954 start logons interactive (dbscontrol vprocmanager)
```

```
CNSSUPV: permissions for jsmith@isdn1954 updated
jsmith@isdn1954: set log restart abort interactive
                Revoked Utilities: dbscontrol vprocmanager
```

## SET ACTIVELOGTABLE

Enables or disables active row filtering on any ResUsage table.

### Syntax

```
SET ACTIVELOGTABLE { tablename | ALL } { ON | OFF }
```

### Syntax Elements

#### *tablename*

name of the ResUsage table for which to enable or disable active row filtering.

ResUsage Table Name	Description
spma	Contains system-wide node information, intended primarily for Teradata engineers. This table is generally not used at customer sites.
ipma	Contains system-wide node information, intended primarily for Teradata engineers. This table is generally not used at customer sites.
svpr	Contains data specific to each virtual processor and its file system.
ivpr	Contains system-wide virtual processor information, intended primarily for Teradata engineers. This table is generally not used at customer sites.
scpu	Contains statistics on the CPUs within the nodes.
sldv	Contains system-wide, storage device statistics.
spdisk	Contains pdisk I/O, cylinder allocation, and migration statistics.
svdisk	Contains statistics collected from the associated storage of the AMP.
sawt	Contains data specific to the AMP worker tasks (AWTs).
sps	Contains resource usage data about workloads.
shst	Contains statistics on the host channels and LANs that communicate with Vantage.
smhm	Contains MAPS statistics.

#### ALL

enables or disables active row filtering on all ResUsage tables depending if the ON or OFF option is specified.

#### ON

enables active row filtering on any one of the ResUsage tables specified.



**OFF**

disables active row filtering on any one of the ResUsage tables specified.

**Usage Notes**

Active row filtering mode reduces the number of data rows that are logged to the database. For the tables for which this option is ON (enabled), only those rows whose data has been modified during the current logging period will be logged.

For some ResUsage tables, like `sps`, there are a large number of possible rows, and most of them are not used at any one time. Logging the inactive rows would waste a large amount of resources. Therefore, Teradata recommends that Active Row Filter Mode remain enabled for these tables.

To log rows, the tables that are enabled for Active Row Filter Mode must also have the table selected for logging and a corresponding Logging Rate of nonzero.

To learn more about active row filtering, see *Teradata Vantage™ - Resource Usage Macros and Tables*, B035-1099.

For information on Summary Mode, see [SET SUMLOGTABLE](#).

To display the active row filtering status, see [GET ACTIVELOGTABLE](#).

**Example: Enabling active row filtering for all ResUsage tables**

The following example enables active row filtering on all ResUsage tables.

```
set activelogtable all on
```

## SET EXTAUTH

Controls whether Vantage users can be authenticated outside (external) of the Vantage software authentication system.

---

### Note:

In scripts, change the deprecated SET SSO command to SET EXTAUTH.

---

### Syntax

```
SET EXTAUTH { OFF | ON | ONLY }
```

### Syntax Elements

#### OFF

Rejects external authentication logons; traditional logons are accepted.

#### ON

Accepts both external authentication and traditional logons. This is the default.

#### ONLY

Accepts external authentication logons only; traditional logons are rejected.

### Usage Notes

External authentication may eliminate the need for your application to declare or store a password on your client system.

To configure how the network allows or disallows traditional and new external authentication logons, see [Gateway Control \(gtwcontrol\)](#).

The new setting is effective immediately.

To check the current setting, use [GET EXTAUTH](#).

The SET EXTAUTH database command duplicates the functionality of the ExternalAuthentication setting in DBS Control. For more information on DBS Control, see [DBS Control \(dbscontrol\)](#).

For additional information on external authentication, see *Teradata Vantage™ - Advanced SQL Engine Security Administration*, B035-1100.

### Example: Setting the system to accept only externally authenticated logons

This example sets the system to accept external authentication logons only.

```
set extauth only
```

```
The External Authentication mode has been set to ONLY.
```

## SET LOGTABLE

Enables or disables logging to any ResUsage table.

### Syntax

```
SET LOGTABLE { tablename | ALL } { OFF | ON }
```

### Syntax Elements

#### *tablename*

name of the ResUsage table for which to enable or disable logging.

ResUsage Table Name	Description
spma	Contains system-wide node information, intended primarily for Teradata engineers. This table is generally not used at customer sites.
ipma	Contains system-wide node information, intended primarily for Teradata engineers. This table is generally not used at customer sites.
svpr	Contains data specific to each virtual processor and its file system.
ivpr	Contains system-wide virtual processor information, intended primarily for Teradata engineers. This table is generally not used at customer sites.
scpu	Contains statistics on the CPUs within the nodes.
sldv	Contains system-wide, storage device statistics.
spdisk	Contains pdisk I/O, cylinder allocation, and migration statistics.
svdisk	Contains statistics collected from the associated storage of the AMP.
sawt	Contains data specific to the AMP worker tasks (AWTs).
sps	Contains resource usage data about workloads.
shst	Contains statistics on the host channels and LANs that communicate with Vantage.
smhm	Contains MAPS statistics.

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

#### ALL

enables or disables logging on all ResUsage tables depending if the ON or OFF option is specified.

**ON**

enables logging on the ResUsage table specified.

**OFF**

disables logging on the ResUsage table specified.

**Usage Notes**

Before a table can be logged, the table must be enabled for logging and the corresponding Logging Rate must be set to a nonzero value.

You can also set logging from the RSS screen of the Control GDO editor (ctl) utility. For more information, see [Control GDO Editor \(ctl\)](#).

To display whether logging to a ResUsage table is enabled or disabled, see [GET LOGTABLE](#).

**Example: Enabling logging on all ResUsage Tables**

In this example, logging is enabled on all the ResUsage tables on the current system.

```
set logtable all on
```

You can use the GET LOGTABLE command to display whether logging has been enabled on all ResUsage tables. For more information, see [GET LOGTABLE](#).

## SET RESOURCE

Changes the rates at which the resource sampling subsystem (RSS) data is requested by applications such as PMPC and ResUsage.

### Syntax

```
SET RESOURCE {
  { COLLECTION | COLL } seconds [ { LOGGING | LOG } seconds ] |
  { LOGGING | LOG } seconds [ { COLLECTION | COLL } seconds ]
}
```

### Syntax Elements

**COLLECTION seconds**

**COLL seconds**

the interval between PMPC interface requests for RSS data.

Valid values are 0 - 3600 seconds.

**LOGGING seconds**

**LOG seconds**

how frequently ResUsage writes statistics to the database.

Valid values are 0 - 3600 seconds.

A setting of 0 means RSS data is not logged.

### Usage Notes

The PMPC Resource Collection Rate is the rate at which the RSS data will be requested by the PM/API application. This affects applications that use the PM/API to display system performance information, such as Teradata Viewpoint. For more information on the PM/API, see *Teradata Vantage™ - Application Programming Reference*, B035-1090.

The Node Logging Rate is the rate at which RSS data is logged to the enabled ResUsage tables in the database.

RSS aligns the logging periods to the clock on the top of every hour. Therefore, these values must divide evenly into 3600 seconds (one hour). The following set of values are legal for RSS logging rates: 0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, 16, 18, 20, 24, 25, 30, 36, 40, 45, 48, 50, 60, 72, 75, 80, 90, 100, 120, 144, 150, 180, 200, 225, 240, 300, 360, 400, 450, 600, 720, 900, 1200, 1800, 3600.

The SET RESOURCE command is effective immediately.

For details about logging, see *Teradata Vantage™ - Resource Usage Macros and Tables*, B035-1099.

**Example: Setting the resource collection and logging rates**

The current RSS Rate Information is as follows:

```
RSS Rate Information:
  PMPC Resource Collection Rate =    30
  Node Logging Rate      =    600
```

In this example, the Collection Rate is set to 600, and the Node Logging Rate is set to 1200.

```
set resource collection 600 logging 1200
```

```
RSS rates set.
  PMPC Resource Collection Rate =    600
  Node Logging Rate      =    1200
```

## SET SESSION COLLECTION

Sets the session collection rate period.

### Syntax

```
SET SESSION { COLLECTION | COLL } n
```

### Syntax Elements

*n*

period in seconds.

Valid rates range from 0 through 3600.

### Usage Notes

If the SET SESSION COLLECTION command is set to OFF, monitoring of the session rate is disabled. However, if you type a Set Session Collection rate between 0 and 3600, the monitoring is enabled automatically.

See also the information about the SET SESSION RATE statement in *Teradata Vantage™ - Application Programming Reference*, B035-1090.

### Example: Setting the session collection rate

In this example, the session collection rate is set to 10 seconds on the current system.

```
set session collection 10
```

```
Monitor Session Rate set to 10 seconds, prior rate was disabled.
```



## SET SUMLOGTABLE

Enables or disables logging in Summary Mode on most ResUsage tables.

### Syntax

```
SET SUMLOGTABLE tablename { ON | OFF }
```

### Syntax Elements

#### *tablename*

name of the ResUsage table for which you want to enable or disable Summary Mode. Summary Mode for the tables is OFF by default.

ResUsage Table Name	Description
spma	Contains system-wide node information, intended primarily for Teradata engineers. This table is generally not used at customer sites.
ipma	Contains system-wide node information, intended primarily for Teradata engineers. This table is generally not used at customer sites.
svpr	Contains data specific to each virtual processor and its file system.
ivpr	Contains system-wide virtual processor information, intended primarily for Teradata engineers. This table is generally not used at customer sites.
scpu	Contains statistics on the CPUs within the nodes.
sldv	Contains system-wide, storage device statistics.
spdisk	Contains pdisk I/O, cylinder allocation, and migration statistics.
svdisk	Contains statistics collected from the associated storage of the AMP.
sawt	Contains data specific to the AMP worker tasks (AWTs).
sps	Contains resource usage data about workloads.
shst	Contains statistics on the host channels and LANs that communicate with Vantage.
smhm	Contains MAPS statistics.

You cannot enable or disable Summary Mode for spma, sps, and ipma tables.

For details on how Summary Mode affects data reporting for these tables, see the table descriptions in *Teradata Vantage™ - Resource Usage Macros and Tables*, B035-1099.

#### ON

enables Summary Mode on the ResUsage table specified.

**OFF**

disables Summary Mode on the ResUsage table specified.

**Usage Notes**

Summary Mode is not applicable to the spma or ipma table because these ResUsage tables only report a single row of data for each node in normal mode.

Enabling Summary Mode reduces database I/O by consolidating and summarizing data on each system node.

To log summary rows for a table, the table must be enabled in the RSS Table Logging Enable group and the RSS Summary Mode Enable group. The corresponding Logging Rate (node or vproc) must be set to a nonzero value.

You can enable Summary Mode for individual tables in which great detail is not needed using the SET SUMLOGTABLE command. For more information on Summary Mode, see *Teradata Vantage™ - Resource Usage Macros and Tables*, B035-1099.

You can also enable or disable logging in Summary Mode from the RSS screen of the Control GDO Editor (ctl) utility. For more information, see [Control GDO Editor \(ctl\)](#).

To display the Summary Mode status of any ResUsage table, see [GET SUMLOGTABLE](#).

**Example: Enabling summary mode logging for a ResUsage table**

The following example enables the sawt table to be logged in Summary Mode.

```
set sumlogtable sawt on
```

# START

Starts a utility in one of the DBW application windows.

## Syntax

```
START
  [ 1 | 2 | 3 | 4 ]
  [ , VPROC=n | -v[=n] ]
  utilityname
  [ utilityargs ]
```

## Syntax Elements

1  
2  
3  
4

number of the DBW application window where you want to start the utility.  
The default is the lowest numbered application window available.

---

**Note:**  
Do not insert spaces between the word START and the window number. For example, START 1 produces an error, but START1does not.

---

*n*

the vproc where the program is to be started.  
The default is the database control AMP, which is normally AMP 0.

---

**Note:**  
The -v options must be lower case.

---

*utilityname*

name of the utility to be started.

---

**Note:**  
Must be a utility or program in one of the PDE or TPA directories.

---

***utilityargs***

remaining part of the command line to pass to *utilityname*.

**Usage Notes**

When a DBW application window is not specified, or the specified window is not available, the utility starts in the first available DBW application window.

Select a vproc number based on the particular utility being run. For example, if the utility you are starting provides information on a specific AMP or vproc, be sure to start the utility on that AMP or vproc.

**Example: Starting a utility in one of the DBW application windows**

The following example starts the Vproc Manager utility in window 3.

```
start3 vprocmanager
Started 'vprocmanager' in window 3
at Mon Jul 16 14:16:22 2007
```

## STOP

Stops the utility running in the specified DBW application window.

### Syntax

```
STOP { 1 | 2 | 3 | 4 }
```

### Syntax Elements

1  
2  
3  
4

numbers of the DBW application windows where a utility is running.

### Usage Notes

You must specify a DBW application window number.

The STOP command is issued from the Supervisor window.

You can also stop a utility from the DBW application window where the utility is running by typing the appropriate command, such as EXIT, STOP, or QUIT.

### Example: Stopping a utility that is running in a DBW application window

The following example shows DBW application window 2 has stopped running the Show Locks (showlocks) utility.

```
stop 2
```

```
Screen 2 (showlocks) is stopped.  
at Fri Aug 30 16:59:09 1996
```

# DBS Control (dbscontrol)

---

**Note:**

The utility described is not available for as-a-service customers.

---

The DBS Control utility, `dbscontrol`, displays and modifies various Advanced SQL Engine configuration settings.

DBS Control settings, also called DBS Control fields, are stored in the DBS Control Globally Distributed Object (GDO). GDOs are binary files that store Advanced SQL Engine configuration settings. They are distributed to and used by every node in the system. The PDE layer of Advanced SQL Engine ensures that the GDO is consistent across all virtual processors.

---

**Note:**

The default settings documented here apply to the Teradata Active Enterprise Data Warehouse platform. Default settings on other purpose-built platforms, such as data appliances, may be different.

---

## Runs From

- Database Window or comparable interface to the Teradata console subsystem, such as `cnstern`
- Linux command line
- Teradata Viewpoint Remote Console portlet

The parallel database extensions (PDE) must be running for DBS Control to run. The database components of Vantage do not need to be running.

For general information on starting the utilities from different interfaces, see [Starting the Utilities](#). For information on Viewpoint, see *Teradata® Viewpoint User Guide*, B035-2206.

## DBS Control Field Groups

The DBS Control fields are grouped logically based on their use by Vantage, as shown in the following table.

Group	Description
General	Miscellaneous settings that affect how Vantage operates.
File System	Settings that affect the Teradata file system.
Compression	Settings that affect data compression at the data block level.
Native Object Store	Settings that affect access to foreign table data.

Group	Description
or NOS	<i>Foreign tables</i> allow Vantage to access data from external, cloud-based data storage, such as AWS S3, without requiring you to manually move the data into the database first from where it natively resides. Foreign tables are identified by a hostname, path, and other metadata that point to the external storage. Vantage can read and process semi-structured or unstructured external data in foreign tables using standard SQL. For example, you can use Teradata analytic functions to examine the data, join it to the relational data in the database, and issue queries against it as you can for other data in Vantage.
Performance	Settings that affect Vantage performance features.
Storage	Settings that determine the temperature assigned to data loaded into empty tables. The temperature corresponds to expected frequency of data access, and can be used to determine how data is stored.  <b>Note:</b> This group is visible only when temperature-based block-level compression is enabled, or on systems that have licensed Teradata Virtual Storage (VS).
Checksum	Special diagnostic settings that are used to ensure the integrity of disk I/O operations on database tables. For information on the checksum group of settings, see <a href="#">Checksum Fields</a> .
Optimizer Statistics	Settings that are used for statistics recollection.

## DBS Control Commands

The following table summarizes DBS Control commands.

Command	Description
<a href="#">DISPLAY</a>	Displays the current values of DBS Control settings.
<a href="#">HELP</a>	Provides help information for the DBS Control utility and individual DBS Control settings.
<a href="#">MODIFY</a>	Modifies DBS Control settings.
<a href="#">QUIT</a>	Exits DBS Control.
<a href="#">WRITE</a>	Writes the current DBS Control settings to the DBS Control GDO.

These commands are described in more detail in the sections that follow.

## DISPLAY

Displays the current values for all DBS Control fields or for a specified field group.

### Syntax

```
{ DISPLAY | D } [ GroupName ]
```

### Syntax Elements

#### GroupName

Which of the following groups the DISPLAY command is directed towards:

- GENERAL (for general fields)
- FILESYS (for file system fields)
- PERFORMANCE (for performance fields)
- CHECKSUM (for checksum fields)
- COMPRESSION (for block-level compression fields)
- STATISTICS (for optimizer statistics fields)
- STORAGE
- NATIVE OBJECT STORE (for Native Object Store fields)

Rather than type in the full *GroupName*, you can type the first few characters, as many as required to uniquely identify the group.

### Usage Notes

If *GroupName* is provided, only the fields associated with the specified group are displayed. If *GroupName* is omitted, the values of all fields are displayed.

### Example: DBS Control DISPLAY output

#### Note:

Actual field numbers and values may be different on your system from those shown in this example.

```
Enter a command, HELP, or QUIT:
DISPLAY
DBS Control Record - General Fields:

1. Version                = 10
2. SysInit                 = TRUE          (2009-01-20 10:09)
3. DeadLockTimeout        = 240            (Seconds)
4. (Reserved for future use)
5. HashFuncDBC             = 6              (Universal)
6. (Reserved for future use)
7. (Reserved for future use)
8. SessionMode             = 0              (Teradata mode)
9. LockLogger              = FALSE          (Disabled)
10. RollbackPriority        = FALSE          (Disabled)
11. MaxLoadTasks           = 5
```



12. RollForwardLock	= FALSE	(Disabled)
13. MaxDecimal	= 15	
14. Century Break	= 0	
15. DateForm	= 0	(IntegerDate)
16. System TimeZone Hour	= 0	
17. System TimeZone Minute	= 0	
18. System TimeZone String	= America Pacific	
19. (Reserved for future use)		
20. RoundHalfwayMagUp	= FALSE	
21. (Reserved for future use)		
22. Target Level Emulation	= FALSE	
23. Export Width Table ID	= 0	(Expected Defaults)
24. DBQLLogLastResp	= FALSE	
25. DBQL Options	= 0	(No options)
26. ExternalAuthentication	= 0	(On)
27. IdCol Batch Size	= 100000	(Default value)
28. LockLogger Delay Filter	= FALSE	
29. LockLogger Delay Filter Time	= 0	(Seconds; Default value)
30. ObjectUseCountCollectRate	= 0	(Minutes; Disabled)
31. LockLogSegmentSize	= 64	(KB; Minimum value)
32. CostProfileId	= 0	
33. DBQLFlushRate	= 600	(seconds)
34. Memory Limit Per Transaction	= 2	(pages)
35. Client Reset Timeout	= 300	(seconds)
36. Temporary Storage Page Size	= 4	(KB)
37. Spill File Path	= /var/tdrsg	
38. (Available for future use)		
39. ChecktableTableLockRetryLimit	= 0	(Retry Forever)
40. EnableCostProfileTLE	= FALSE	
41. EnableSetCostProfile	= 0	(Disabled)
42. (Reserved for future use)		
43. DisableUDTImplCastForSysFuncOp	= FALSE	
44. CurHashBucketSize	= 20	(Bits)
45. NewHashBucketSize	= 20	(Bits)
46. MaxLoadAWT	= 0	
47. MaxRowHashBlocksPercent	= 50	(Percent; Default value)
48. MonSesCPUNormalization	= FALSE	
49. TempLargePageSize	= 64	(KB; Minimum value)
50. RepCacheSegSize	= 512	(KB; Minimum value)
51. MaxDownRegions	= 6	(Default value)
52. MPS_IncludePEOnlyNodes	= FALSE	
53. PrimaryIndexDefault	= D	(First Column is NUPI)
54. AccessLockForUncomRead	= FALSE	(Default value)
55. EnabNonTempoOp = FALSE	= FALSE	(Disabled)
56. InclINTforGrntOrRevokAll	= FALSE	
57. TimeDateWZControl	= 0	(Enabled; Teradata Default)
58. (Reserved for future use)		
59. SysInitRelease	= 13010000	
60. DefaultCaseSpec	= FALSE	
61. PMPC_TimeoutSecs	= 60	(Seconds; Default value)
62. ExportOrderBySegmentLimit	= 500	
63. MLoadDiscardDupRowUPI	= FALSE	
64. DBQL CPU/IO Collection	= 0	(Algorithm 3, with step adjustments, default)
65. NoDot0Backdown	= TRUE	
66. SnapBypassAggrCache	= TRUE	
67. RedriveProtection	= 1	(Redrive memory-based)
68. RedriveDefaultParticipation	= 1	(Redrive Participation is ON)
69. PMPC_SessionRateThreshold	= 60	(Seconds)
70. RoundNumberAsDec	= TRUE	
71. DMLStatementShipping	= 0	(Disabled, default value)
72. SHOW IN XML Formatted Output	= FALSE	(Disabled, default value)
73. XML_MemoryLimit	= 4	(MB; Default value)
74. Obsolete (DisplacementOnOverlap)		
75. NameValidationRule	= 0	(Default name validation)
76. EnableEON	= TRUE	
77. DisableRecoverableNetProtocol	= FALSE	(RNP enabled, default)
78. Default Character Set	= 1	(LATIN)
79. EnableLogonMsg	= FALSE	
80. COUNT_mode	= 0	

81. AutoProvision	= FALSE
82. EnableJSON	= TRUE
83. JSON_MaxDepth	= 512 (Default 512)
84. JSON_AttributeSize	= 4096 (Default 4096)
85. (Reserved for future use)	
86. DisableJSONValidation	= FALSE
87. FixSRandomRange	= TRUE
88. Temporal Behavior	= 1 (ANSITemporal Behavior)
89. (Reserved for future use)	
90. (Reserved for future use)	
91. (Reserved for future use)	
92. MLOADXUtilityLimits	= FALSE
93. MaxMLOADXTasks	= 30 (Default 30)
94. MaxMLOADXAWT	= 48
95. CPUUpdate	= 3 (0-4, Default 3)
96. EnableDATASET	= TRUE
97. DATASETAttributeSize	= 4096 (Default 4096)
98. DisableDATASETValidation	= FALSE
99. DotNotationOnErrorCondition	= 0 (Default 0)
100. Obsolete (XMLWarnThreshold)	
101. DefaultPermSkewLimitPercent	= 0 (0-10000%, default 0%)
102. DefaultSpoolSkewLimitPercent	= 0 (0-10000%, default 0%)
103. DefaultTempSkewLimitPercent	= 0 (0-10000%, default 0%)
104. GlobalSpaceSoftLimitPercent	= 0.00 (0.0-99.99%, default 0%)
105. ProratedSpaceDistPercent	= 0 (0-100%, default 0%)
106. MaxSetQueryBandSize	= 3 (2-4 KB, Default 3)
107. LegacySpaceAcctg	= FALSE (Default FALSE)
108. CDCProtocol	= 0 (Default, Legacy Change Data Capture)
109. MetadataCapture	= 0 (Default, No Metadata)
110. ClientResetAction	= 0 (Default, continue)
111. TruncRoundReturnTimestamp	= TRUE (Default FALSE)
112. ShowAllUserNames	= FALSE (Default FALSE)
113. FastAlterEnable	= FALSE (default FALSE)
114. FastAlterDefault	= FALSE (default FALSE)
115. EnableDATASETCSV	= TRUE
116 - 118 (Reserved for future use)	
119. LegacyUnAuthBehavior	= FALSE (default FALSE)
120. (Reserved for future use)	

## DBS Control Record - File System Fields:

1. FreeSpacePercent	= 0	(Percent; Default value)
2. MiniCylPackLowCylProd	= 10	(free cylinders; Default value)
3. PermDBSize	= 254	(Sectors; Default value)
4. JournalDBSize	= 254	(Sectors; Default value)
5. DefragLowCylProd	= 100	(Free cylinders; Default value)
6. PermDBAllocUnit	= 1	(Sectors; Default value)
7. Cylinders Saved for PERM	= 10	(Cylinders; Default value)
8. DisableWALforDBs	= FALSE	(Enabled)
9. DisableWAL	= FALSE	(Enabled)
10. WAL Buffers	= 20	(WAL log buffers; Default value)
11. MaxSyncWALWrites	= 5	(MaxSyncWalWrites; Default value)
12. SmallDepotCylsPerPdsk	= 2	(Cylinders; Default value)
13. LargeDepotCylsPerPdsk	= 1	(Cylinders; Default value)
14. WAL Checkpoint Interval	= 60	(Seconds; Default value)
15. Free Cylinder Cache Size	= 100	(Number of cylinders; Default value)
16. Bkgrnd Age Cycle Interval	= 60	(Seconds; Default value)
17. DisableAutoCylpack	= FALSE	(Enabled)
18. AutoCylPackColdData	= FALSE	(Disabled)
19. AutoCylPackFSP	= 10	(Percent; Default value)
20. AutoCylPackStyle	= 2	(PACK / UNPACK)
21. AutoCylPackThresh	= 5	(Percent; Default value)
22. AutoCylPackFreecylThresh	= 20	(Cylinders; Default value)
23. AutoCylPackInterval	= 60	(Seconds; Default value)
24. AutoCylPackIOThresh	= 1	(Number; Default value)
25. (Reserved for future use)		
26. (Reserved for future use)		
27. MergeBlockRatio	= 60	(Percent; Default value)

28. DisableMergeBlocks	= FALSE	(Enabled)
29. DisableRestartableBLC	= FALSE	(Enabled)
30. SpoolBigDBAllocUnit	= 1	(Sectors; Default value)

## DBS Control Record - Performance Fields:

1. DictionaryCacheSize	= 3072	(KB; Default value)
2. DBSCacheCtrl	= TRUE	(Enabled)
3. DBSCacheThr	= 10	(Percent; Default value)
4. MaxParseTreeSegs	= 3000	
5. ReadAhead	= TRUE	(Enabled)
6. StepsSegmentSize	= 1024	(KB; Default value)
7. RedistBufSize	= 3832	(Bytes; Default value))
8. DisableSyncScan	= FALSE	(Enabled)
9. SyncScanCacheThr	= 10	(Percent; Default value)
10. HTMemAlloc	= 10	(Percent; Default value)
11. SkewAllowance	= 75	(Percent; Default value)
12. Read Ahead Count	= 1	(Data block; Default value)
13. PPICacheThrP	= 10	(PPI cache per AMP = 1% of 100MB: 1MB)
14. ReadLockOnly	= FALSE	
15. IAMaxWorkloadCache	= 32	(MB; Default value)
16. MaxRequestsSaved	= 600	(Default value)
17. UtilityReadAheadCount	= 10	(Blocks; Default value)
18. StandAloneReadAheadCount	= 20	(Blocks; Default value)
19. DisablePeekUsing	= FALSE	(Enabled)
20. IVMaxWorkloadCache	= 1	(MB; Default value)
21. RevertJoinPlanning	= FALSE	(Default value)
22. MaxJoinTables	= 128	
23. DBQLXMLPlanMemLimit	= 8192	(KB; Default value)
24. LimitInlistCVal	= 1048576	
25. NumStatisticsCacheSegs	= 4	
26. SHOW IN XML Memory Limit	= 64000	(KB; Default value)
27. DictionaryCacheSegmentSize	= 16	( * 64 KBs; Default value)
28. PRPDMaxSkewedVals	= 50	(Default value)
29. PRPDSkewPct	= 50	(Percent)
30. DBQLLOBCacheSize	= 2	(MB; Default:2, Min:2, Max:16)
31. HJ2IMHJ	= FALSE	
32. InListRewriteThreshold	= 512	(Default:512, Min:1, Max:32767)
33. MaxNumInListRewrite	= 2	(Default:2, Min:0, Max:100)
34. InListRewriteOption	= 1	(Default:1, Min:0, Max:7)
35. DBQLDefCacheSize	= 2	(Default:2, Min:0, Max:16)

## DBS Control Record - Disk I/O Integrity Fields:

1. System Tables	= OFF
2. System Journal Tables	= OFF
3. System Logging Tables	= OFF
4. User Tables	= OFF
5. Permanent Journal Tables	= OFF
6. Temporary Tables	= OFF

## CHECKSUM LEVEL DEFINITIONS

OFF - Do not calculate or validate checksums.  
 ON - Calculate and validate checksums using the entire disk block.

## DBS Control Record - Compression (Block Level) Fields:

1. BlockLevelCompression	= ON	
2. (Reserved for future use)		
3. CompressSpoolDBs	= NEVER	(ALWAYS, NEVER, IFNOTCACHED)
4. (Reserved for future use)		
5. CompressMloadWorkDBs	= NEVER	(ALWAYS,
UNLESSQBNO, NEVER, ONLYIFQBYES)		
6. CompressPJDBs	= NEVER	(ALWAYS, NEVER)
7. MinDBSectsToCompress	= 16	
8. MinPercentCompReduction	= 20%	
9. CompressionAlgorithm	= ZLIB	(ZLIB, ELZS-H, ELZS-S)
10. CompressionLevel	= 1	(1-9; Default value = 1)
11. (Reserved for future use)		

12. UncompressReservedSpace	= 5%	
13. (Reserved for future use)		
14. CompressPermPrimaryDBs	= ONLYIFQBYES	(ALWAYS,
UNLESSQBNO, NEVER, ONLYIFQBYES)		
15. CompressPermFallbackDBs	= ONLYIFQBYES	(ALWAYS,
UNLESSQBNO, NEVER, ONLYIFQBYES)		
16. CompressPermPrimaryCLOBDBs	= NEVER	(ALWAYS,
UNLESSQBNO, NEVER, ONLYIFQBYES)		
17. CompressPermFallbackCLOBDBs	= NEVER	(ALWAYS,
UNLESSQBNO, NEVER, ONLYIFQBYES)		
18. CompressGlobalTempPrimaryDBs	= NEVER	(ALWAYS,
UNLESSQBNO, NEVER, ONLYIFQBYES)		
19. CompressGlobalTempFallbackDBs	= NEVER	(ALWAYS,
UNLESSQBNO, NEVER, ONLYIFQBYES)		
20. CompressGlobalTempPrimaryCLOBDBs	= NEVER	(ALWAYS,
UNLESSQBNO, NEVER, ONLYIFQBYES)		
21. CompressGlobalTempFallbackCLOBDBs	= NEVER	(ALWAYS,
UNLESSQBNO, NEVER, ONLYIFQBYES)		
22 - 31 (Reserved for future use)		
32. EnableTempBLC	= TRUE	(TRUE, FALSE)
33. DefaultTableMode	= MANUAL	(AUTOTEMP, MANUAL, ALWAYS, NEVER)
34. TempBLCThresh	= COLD	(COLD, WARM, HOT, 0-100 )
35. TempBLCSpread	= 5%	(1-25%)
36. TempBLCInterval	= 10 minute(s)	(1-120)
37. TempBLCIOThresh	= 1	(1-1000)
38. TempBLCPriority	= MEDIUM/DEFAULT	(LOW/BOTTOM, MEDIUM/DEFAULT, HIGH, RUSH/TOP)
39. TempBLCRescanPeriod	= 7 day(s)	(1-90)
40. CompressionZLIBMethod	= IPPZLIB	(ZLIB, IPPZLIB)

## DBS Control Record - Storage Fields:

1. (Reserved for future use)		
2. SPOOL Temperature	= HOT	(HOT, WARM, COLD, DEFAULT)
3. (Reserved for future use)		
4. PERMANENT JOURNAL Temperature	= HOT	(HOT, WARM, COLD, DEFAULT)
5. WAL Temperature	= HOT	(HOT, WARM, COLD, DEFAULT)
6. DEPOT Temperature	= HOT	(HOT, WARM, COLD, DEFAULT)
7. All Other Temperature	= WARM	(HOT, WARM, COLD, DEFAULT)
8. PERM Primary Temperature	= WARM	(HOT, WARM, COLD, DEFAULT)
9. PERM Primary CLOB Temperature	= WARM	(HOT, WARM, COLD, DEFAULT)
10. PERM Fallback Temperature	= WARM	(HOT, WARM, COLD, DEFAULT)
11. PERM Fallback CLOB Temperature	= WARM	(HOT, WARM, COLD, DEFAULT)
12. GLOBAL TEMP Primary Temperature	= WARM	(HOT, WARM, COLD, DEFAULT)
13. GLOBAL TEMP Primary CLOB Temperature	= WARM	(HOT, WARM, COLD, DEFAULT)
14. GLOBAL TEMP Fallback Temperature	= WARM	(HOT, WARM, COLD, DEFAULT)
15. GLOBAL TEMP Fallback CLOB Temperature	= WARM	(HOT, WARM, COLD, DEFAULT)
16. TIM Cache Load Disabled	= FALSE	(TRUE, FALSE; Default value)
17. TIM Cache Load Throttle	= FALSE	(TRUE, FALSE; Default value)

## DBS Control Record - Optimizer Statistics Fields:

1. DefaultTimeThreshold	= 0 (Disabled; Default value)
2. DefaultUserChangeThreshold	= 0 (Disabled; Default value)
3. SysChangeThresholdOption	= 0 (Enabled with DBQL ObjectUsage; Default value)
4. SysSampleOption	= 0 (Enabled; Default value)
5. BLCStats	= 0 (Enabled; Default value)

## DBS Control Record - Native Object Store Fields:

1. EnableNOS	= TRUE (Enabled; Default value)
2. ConnectionBufferSize	= 16 (MB; Default value)
3. ConnectionTimeout	= 40 (Seconds; Default value)
4. RequestTimeout	= 60 (Seconds; Default value)
5. RequestDuration	= 300 (Seconds; Default value)
6. UserRetryLimit	= 1 (Default value)
7. DefaultRowGroupSize	= 16 (MB; Default value)

11. ForeignTableErrorsData	= 0 (Log errors and data; Default value)
12. ForeignTableErrorsLimit	= 0 (Limit = 10; Default value)

## HELP

Provides help information on the DBS Control utility.

### Syntax

```
{ HELP | H } [ { ALL | A } | keyword ]
```

### Syntax Elements

**ALL**

**A**

Displays all DBS Control help information, including help on DBS Control commands and help on all fields.

***keyword***

Displays help information for either a specific DBS Control command or for a group of DBS Control fields.

*keyword* can be either a DBS Control command name or a DBS Control field group name.

You can type the first few characters that serve to uniquely identify a *keyword*, rather than the entire word.

### Usage Notes

If the options are omitted (that is, only HELP is specified), a brief introduction to DBS Control is displayed. If a keyword is specified, a detailed help display is provided.

### Example: DBS Control online help

The following example shows detailed help for the DISPLAY command:

```
Enter a command, HELP, or QUIT:
```

```
help display
```

```
DISPLAY [ <GroupName> ]
```

```
o This command will display the values of the DBS Control
  Record fields. If a GroupName is provided, only the fields
  associated with the specified group will be displayed.
  If GroupName is omitted, the values of all fields will be
  displayed.
```

```
o Valid GroupNames are:
```

GENERAL, FILESYS, PERFORMANCE, CHECKSUM., STORAGE,  
STATISTICS and COMPRESSION

Enter "HELP <GroupName>" for additional information.

Enter a command, HELP, or QUIT:

## MODIFY

Modifies the value of the writable DBS Control Record fields.

---

### Note:

Use the `DISPLAY GroupName` command to return the field numbers for a particular group.

---

### Syntax

```
{ MODIFY | M } [ GroupName field# = value ]
```

### Syntax Elements

#### *GroupName*

Indicates which of the following groups the MODIFY command is directed towards:

- GENERAL (for general fields)
- FILESYS (for file system fields)
- PERFORMANCE (for performance fields)
- CHECKSUM (for checksum fields)
- STORAGE (for default storage temperatures)
- STATISTICS (for optimizer statistics fields)
- COMPRESSION (for block-level compression fields)

You can type the first few characters that uniquely identify each *GroupName* rather than the entire name.

#### *field*

Specifies the name of the field to receive the new value.

#### *value*

Specifies the new value of the specified field.

In most cases, this is a decimal value. For Boolean fields, use TRUE or T and FALSE or F. All values are checked against the valid range of values of the respective field.

### Usage Notes

- Use of the MODIFY command alone does not affect the current system GDO. The MODIFY command indicates when any changes made will become effective, either after the next database restart or after the DBS Control Record has been written.
- If you omit the options, DBS Control explicitly prompts for the group name, field number, and the new value of the field.



- Changes to the configuration settings are logged automatically. A history of the changes made to the GDO can be viewed using the gdomview command-line tool from the system console. The gdomview online documentation is accessible from a system console using the man and pdehelp commands.

**Example: Modifying a DBS Control field value**

The following example shows how to modify the Deadlock Timeout field:

```
Enter a command, HELP, or QUIT:  
MODIFY GENERAL 3 = 600
```

The DeadlockTimeout field has been modified from 240 to 600.

NOTE: This change will become effective after the next DBS restart.

## QUIT

Causes DBS Control to exit.

### Syntax

```
{ QUIT | QU }
```

### Usage Notes

If the DBS Control Record has been modified, but you did not issue a WRITE command, you are asked to write the changes in the system GDO or terminate without saving them.

### Example: Quitting DBS Control

The following example shows how to exit DBS Control and issue a WRITE command after modifying the DBS Control Record:

```
Enter a command, HELP, or QUIT:
```

```
QUIT
```

```
The DBS Control Record has been modified.
```

```
Enter 'W' to write to the DBS Control GDO or  
      'Q' to terminate with no update:
```

```
W
```

```
Locking the DBS Control GDO...
```

```
Updating the DBS Control GDO...
```

```
Exiting DBSControl...
```

## WRITE

Forces the DBS Control Record with all current settings to be written out to its system GDO counterpart.

### Syntax

```
{ WRITE | WR }
```

### Usage Notes

A system-wide lock is placed on the DBS Control GDO to write the changes.

DBS Control notifies you if the DBS Control GDO has been modified by someone else during your DBS Control session, and offers to overwrite their changes with yours or terminate without saving them.

### Example: Writing DBS Control changes to the dbscontrol GDO

The following example writes the current settings to the GDO without exiting DBS Control.

```
Enter a command, HELP, or QUIT:
```

```
write
```

```
Locking the DBS Control GDO ...
```

```
Updating the DBS Control GDO ...
```

```
Enter a command, HELP, or QUIT:
```

## DBS Control Fields

The following table summarizes the DBS Control fields and indicates their groups. The fields are described in more detail in the sections that follow.

**Note:**

The default settings documented here apply to the Teradata Active Enterprise Data Warehouse platform. Default settings on other purpose-built platforms, such as data appliances, may be different.

Field	Description	Group
<a href="#">AccessLockForUncomRead</a>	Determines whether SELECT statements embedded in INSERT ... SELECT, UPDATE ... SELECT, DELETE ... SELECT statements place READ or ACCESS locks on the source table.	General
All Other Temperature see <a href="#">Storage Group</a>	Determines the initial temperature assigned to data loaded or inserted into tables of a type not otherwise specified by Storage group fields.	Storage
<a href="#">AutoCylPackColdData</a>	Determines whether AutoCylPack operations are performed on cylinders storing cold (infrequently accessed) data, in addition to cylinders storing more frequently accessed data.  <b>Note:</b> This field is effective only if you have licensed Teradata Virtual Storage. For more information, see <i>Teradata Vantage™ - Teradata® Virtual Storage</i> , B035-1179.	File System
<a href="#">AutoCylPackFreeCylThresh</a>	The number of free cylinders below which AutoCylPack will not operate.	File System
<a href="#">AutoCylPackFSP</a>	The percentage of storage space within a cylinder that AutoCylPack operations leave free of data, to allow for future table growth.	File System
<a href="#">AutoCylPackInterval</a>	The amount of time that the AutoCylPack task waits between checks for new AutoCylPack work.	File System
<a href="#">AutoCylPackIOThresh</a>	The number of pending I/O operations above which the AutoCylPack task waits for I/Os to complete before continuing.	File System
<a href="#">AutoCylPackStyle</a>	Determines whether the AutoCylPack task is restricted to packing data more tightly on cylinders, unpacking data, or if it	File System

Field	Description	Group
	is unrestricted, and can both pack and unpack data to achieve the target FSP.	
<a href="#">AutoCylPackThresh</a>	The threshold difference between actual free space on a cylinder and the target AutoCylPack FSP, below which AutoCylPack will not process the cylinder.	File System
<a href="#">AutoProvision</a>	Determines whether unknown external logons are automatically added as new users (CREATE USER).	General
<a href="#">Bkgrnd Age Cycle Interval</a>	Determines the amount of time that elapses between background cycles to write a subset of modified segments in the cache to disk.	File System
<a href="#">BLCStats</a>	Determines whether block-level compression (BLC) compression ratio information is gathered by COLLECT STATISTICS statements.	Optimizer Statistics
<a href="#">BLCStatsForCDS</a>	Specifies whether the the COLLECT STATISTICS statement calculates the estimated BLC compression factor for tables.	Optimizer Statistics
<a href="#">BlockLevelCompression</a>	Indicates whether the block-level compression (BLC) feature is enabled.	Compression
<a href="#">CDCProtocol</a>	Determines whether the Change Data Capture (CDC) protocol used in database replication scenarios allows for user-defined replication groups or reserved system-defined replication groups for Teradata® Unity™.	General
<a href="#">Century Break</a>	Defines how to interpret character-format Vantage date input when the input format only has two digits representing the year.	General
<a href="#">ChecktableTableLockRetryLimit</a>	Specifies that in nonconcurrent mode, CheckTable retries a table check for a specified limit when the table is locked by another application.	General
<a href="#">ClientResetAction</a>	Controls what happens to queries when Teradata® Unity™ is disconnected or disabled.	General
<a href="#">Client Reset Timeout</a>	Specifies how long the Relay Services Gateway (RSG) waits for an intermediary to reconnect before taking action.	General

Field	Description	Group
<a href="#">CompressGlobalTempFallbackDBs</a>	Specifies conditions under which Global Temporary Table data blocks in fallback subtables will be compressed by default.	Compression
<a href="#">CompressGlobalTempFallbackCLOBDBs</a>	Specifies conditions under which Global Temporary Table CLOB data blocks in fallback tables will be compressed by default.	Compression
<a href="#">CompressGlobalTempPrimaryDBs</a>	Specifies conditions under which Global Temporary Table data blocks in primary data subtables will be compressed by default.	Compression
<a href="#">CompressGlobalTempPrimaryCLOBDBs</a>	Specifies conditions under which Global Temporary Table CLOB data blocks in primary data subtables will be compressed by default.	Compression
<a href="#">CompressionAlgorithm</a>	Specifies the algorithm to use for compressing DBs.	Compression
<a href="#">CompressionLevel</a>	Determines whether DB compression favors speed or degree of compression.	Compression
<a href="#">CompressionZLIBMethod</a>	Determines whether the ZLIB compression uses the Intel Integrated Performance Primitives library for better performance on Intel platforms.	Compression
<a href="#">CompressPermFallbackDBs</a>	Specifies conditions under which permanent data blocks in fallback subtables will be compressed by default.	Compression
<a href="#">CompressPermFallbackCLOBDBs</a>	Specifies conditions under which permanent CLOB data blocks in fallback subtables will be compressed by default.	Compression
<a href="#">CompressPermPrimaryDBs</a>	Specifies conditions under which permanent data blocks in primary data subtables will be compressed by default.	Compression
<a href="#">CompressPermPrimaryCLOBDBs</a>	Specifies conditions under which permanent CLOB data blocks in primary data subtables will be compressed by default.	Compression
<a href="#">CompressPJDBs</a>	Specifies whether Permanent Journal DBs will be compressed by default.	Compression
<a href="#">CompressSpoolDBs</a>	Specifies conditions under which spool and volatile data blocks will be compressed by default.	Compression

Field	Description	Group
<a href="#">ConnectionBufferSize</a>	Controls how much memory is allocated for buffering data between Vantage and external data sources.	Native Object Store
<a href="#">ConnectionTimeout</a>	Controls how long Vantage waits for a new connection to an external network to be established when creating or querying a foreign table.	Native Object Store
<a href="#">CompressMloadWorkDBs</a>	Specifies conditions under which work data blocks will be compressed by default.	Compression
<a href="#">CostProfileId</a>	Contains the system standard cost profile ID number, which defines the cost profile that the system will use.	General
<a href="#">COUNT_mode</a>	Determines the data type returned by COUNT operations.	General
<a href="#">CPUUpdate</a>	Determines how column partitions are updated.	General
<a href="#">CurHashBucketSize</a>	Indicates the number of bits used to identify a hash bucket in the current system configuration.	General
<a href="#">CylindersSavedforPERM</a>	Saves some number of cylinders for permanent data only.	File System
<a href="#">DATASETAttributeSize</a>	Denotes the size (number of characters) of attributes returned by the DATASET type dot notation.	General
<a href="#">DateForm</a>	Defines whether Integer Date or ANSI Date formatting is used for a session.	General
<a href="#">DBQL CPU/IO Collection</a>	Sets the algorithm used by DBQL to collect CPU and input/output statistics. DBQL CPU/IO collection captures performance data for the execution of a query at the request level and, optionally, at the step level.	General
<a href="#">DBQL_AWTDPS_CacheMaximum</a>	Limits the size to which the per-AMP AWT DBQL Performance Stats (DPS) cache can grow. This cache holds DBQL request and step data for active requests.	Performance
<a href="#">DBQLDefCacheSize</a>	Sets the DBQL cache size and determines whether DBQL uses 64K or 1MB data blocks.	Performance
<a href="#">DBQLFlushRate</a>	Controls the length of time between flushing the Database Query Log (DBQL) cache.	General

Field	Description	Group
<a href="#">DBQLLOBCacheSize</a>	Determines the amount of cache allocated for DBQL Data Dictionary tables with LOB columns.	Performance
<a href="#">DBQLLogLastResp</a>	Determines whether DBQL logs a pseudo step labeled “RESP” after the last response of a DBQL logged request completes.	General
<a href="#">DBQL Options</a>	This field is unused, but reserved for future DBQL use. It should not be changed.	General
<a href="#">DBQLXMLPlanMemLimit</a>	Specifies the amount of memory used by the XMLPLAN option of DBQL.	Performance
<a href="#">DBSCacheCtrl</a>	Enables or disables the performance enhancements associated with the Cache Control Page-Release Interface associated with the DBSCacheThr field.	Performance
<a href="#">DBSCacheThr</a>	Specifies the percentage value to use for calculating the cache threshold when the DBSCacheCtrl field is enabled.	Performance
<a href="#">DeadLockTimeout</a>	Used by the Dispatcher to determine the interval (in seconds) between deadlock timeout detection cycles.	General
<a href="#">DefaultCaseSpec</a>	Determines whether character string comparisons consider character case and whether character columns are considered case specific by default in Teradata session mode.	General
<a href="#">Default Character Set</a>	Specifies the default character set assigned to users if a default character set is not explicitly specified in the CREATE USER statement.	General
<a href="#">DefaultTableMode</a>	Determines how BLC is managed for Permanent storage tables that have the BLOCKCOMPRESSION option set to DEFAULT, or that do not have the BLOCKCOMPRESSION option explicitly specified when the table is created. When changed, causes TBBLC to immediately start a new cycle.	Compression
<a href="#">DefaultPermSkewLimitPercent</a>	Specifies the percentage of permanent space that an AMP is allowed to use over and above the per-AMP quota.	General
<a href="#">DefaultRowGroupSize</a>	If a value is not explicitly provided through user query, determines the size of RowGroups used while creating Parquet files.	Native Object Store



Field	Description	Group
<a href="#">DefaultSpoolSkewLimitPercent</a>	Specifies the percentage of spool space that an AMP is allowed to use over and above the per-AMP quota.	General
<a href="#">DefaultTimeThreshold</a>	Defines a default time threshold value in days for statistics collection.	Optimizer Statistics
<a href="#">DefaultTempSkewLimitPercent</a>	Specifies the percentage of temporary space that an AMP is allowed to use over and above the per-AMP quota.	General
<a href="#">DefaultUserChangeThreshold</a>	Defines a default data change percentage threshold value for statistics collection.	Optimizer Statistics
<a href="#">DefragLowCylProd</a>	Determines the number of free cylinders below which cylinder defragmentation can begin.	File System
DEPOT Temperature see <a href="#">Storage Group</a>	Determines the initial temperature assigned to data loaded or inserted into tables of this type.	Storage
<a href="#">DictionaryCacheSize</a>	Defines the size of the dictionary cache for each PE on the system.	Performance
<a href="#">DictionaryCacheSegmentSize</a>	Defines the size of the dictionary segment cache for each PE on the system.	Performance
<a href="#">DictionaryDBSize</a>	Specifies the maximum size for multirow data blocks in dictionary (database DBC) tables.	File System
<a href="#">DisableAutoCylPack</a>	Disables AutoCylPack automatic background cylinder packing. Does not affect MiniCylPack operations, which happen when the number of free cylinders is critically low.	File System
<a href="#">DisableDATASETValidation</a>	Determines whether validation is enabled or disabled for the native Teradata DATASET data type.	General
<a href="#">DisableJSONValidation</a>	Determines whether validation is enabled or disabled for the native Teradata JSON data type.	General
<a href="#">DisableMergeBlocks</a>	Determines whether the file system automatically merges data blocks together on storage cylinders during full table modify operations. Merging several small data blocks into a single larger block can improve system performance by reducing disk I/O.	File System

Field	Description	Group
<a href="#">DisablePeekUsing</a>	Enables or disables the performance enhancements associated with exposed USING values in parameterized queries.	Performance
<a href="#">DisableRecoverableNetProtocol</a>	Determines whether the Recoverable Network Protocol is enabled.	General
<a href="#">DisableRestartableBLC</a>	Determines whether BLC compression and decompression operations will resume after being interrupted by a system restart.	File System
<a href="#">DisableSyncScan</a>	Enables or disables the performance enhancements associated with synchronized full table scans.	Performance
<a href="#">DisableUDTImplCastForSysFuncOp</a>	Disables/enables implicit cast/conversion of UDT expressions passed to built-in system operators/functions.	General
<a href="#">DisableWAL</a>	Forces the writing of data blocks and cylinder indexes directly to disk rather than writing the changes to the WAL log.	File System
<a href="#">DisableWALforDBs</a>	Forces changed data blocks to be written directly to disk rather than having the changes written to the WAL log.	File System
<a href="#">DMLStatementShipping [Deprecated]</a>	Do not use.	General
<a href="#">DotNotationOnErrorCondition</a>	Determines how Vantage responds when queries using dot notation return a list of values.	General
<a href="#">EnableCostProfileTLE</a>	Determines whether Optimizer Cost Estimation Subsystem (OCES) diagnostics are enabled in combination with Target Level Emulation (TLE).	General
<a href="#">EnableDATASET</a>	Determines whether native Teradata DATASET data type support is enabled in Vantage.	General
<a href="#">EnableDATASETCSV</a>	Determines whether the DATASET CSV storage format is available for use by Vantage.	General
<a href="#">EnableDynamicParameterization</a>	Determines whether the Optimizer uses dynamic parameterization of literals in processing repeat requests that differ only in the literal values used in the predicates of WHERE and ON clauses.	General
<a href="#">EnableEON</a>	Confirms object naming rules are enforced. This field value cannot be changed.	General

Field	Description	Group
<a href="#">EnableJSON</a>	Determines whether native JSON support is enabled in Vantage.	General
<a href="#">EnableLogonsMsg</a>	Enables or disables the display of “Logons are enabled” message in /var/log/messages on all nodes of a multinode system.	General
<a href="#">EnableNOS</a>	Determines whether the Native Object Store (NOS) feature is enabled or disabled.	Native Object Store
<a href="#">EnableTempBLC</a>	Determines whether temperature-based block-level compression (TBBLC) is enabled.	Compression
<a href="#">EnabNonTempoOp</a>	Determines whether nontemporal operations such as NONTEMPORAL DELETE, NONTEMPORAL UPDATE, NONTEMPORAL INSERT, NONTEMPORAL CREATE TABLE AS, and NONTEMPORAL ALTER TABLE can be used.	General
<a href="#">EnableSetCostProfile</a>	Controls whether the system cost profile can be changed for purposes of system emulation and special-case system tuning.	General
<a href="#">ExportOrderBySegmentLimit</a>	Specifies the maximum number of segments that can be allocated for FastExport redistribution when data is to be sorted.	General
<a href="#">Export Width Table ID</a>	Controls the export width of a character in bytes.	General
<a href="#">ExternalAuthentication</a>	Controls whether Vantage users can be authenticated outside (external) of the Vantage software authentication system.	General
<a href="#">FastAlterDefault</a>	Allows automatic use of Fast Column Add (FCA) to add new columns to populated user tables without using USING FAST MODE ON in ALTER TABLE syntax.	General
<a href="#">FastAlterEnable</a>	Used to turn off Fast Mode in ALTER TABLE.	General
<a href="#">FixSRandomRange</a>	Fixes a rare issue with the RANDOM(lower_bound, upper_bound) function that causes it to return a value outside of the specified bounds under certain circumstances.	General
<a href="#">ForeignTableErrorsData</a>	Logs the data record that contains the error.	NOS
<a href="#">ForeignTableErrorsLimit</a>	Limits the number of errors logged.	NOS

Field	Description	Group
<a href="#">ForwardCredential</a>	Determines whether to forward authenticated user credential token to Vantage.	General
<a href="#">Free Cylinder Cache Size</a>	Determines how many cylinders are to be managed in File System cache for use as spool cylinders.	File System
	Used to turn off Fast Mode ALTER TABLE.	General
<a href="#">FreeSpacePercent</a>	Determines the percentage of free space reserved on cylinders during bulk load operations.	File System
<a href="#">GlobalSpaceSoftLimitPercent</a>	Specifies the percentage limit for permanent, spool, and temporary space above the maximum value that the system permits for any database or user.	General
GLOBAL TEMP Fallback Temperature see <a href="#">Storage Group</a>	Determines the initial temperature assigned to data loaded or inserted into tables of this type.	Storage
GLOBAL TEMP Fallback CLOB Temperature see <a href="#">Storage Group</a>	Determines the initial temperature assigned to data loaded or inserted into tables of this type.	Storage
GLOBAL TEMP Primary Temperature see <a href="#">Storage Group</a>	Determines the initial temperature assigned to data loaded or inserted into tables of this type.	Storage
GLOBAL TEMP Primary CLOB Temperature see <a href="#">Storage Group</a>	Determines the initial temperature assigned to data loaded or inserted into tables of this type.	Storage
<a href="#">HashFuncDBC</a>	Defines the hashing function that Vantage uses.	General
<a href="#">HJ2IMHJ</a>	Directs Vantage to preferentially use the In-Memory Hash Join method to perform hash joins and dynamic hash joins.	Performance
<a href="#">HTMemAlloc</a>	Specifies the percentage of memory to be allocated to a hash table for a hash join.	Performance
<a href="#">IAMaxWorkloadCache</a>	Defines the maximum size of the workload cache when performing analysis operations.	Performance
<a href="#">IdCol Batch Size</a>	Indicates the size of a pool of numbers reserved for generating numbers for rows to be inserted into a table with an identity column.	General
<a href="#">IncINTforGrntOrRevokAll</a>	Determines whether the NONTEMPORAL privilege is included when GRANT ALL and	General

Field	Description	Group
	REVOKE ALL statements specify a user, database, view, or table.	
<a href="#">InListRewriteOption</a>	Specifies the type of IN-list processing that will be used for rewrite processing of queries that have IN-lists.	Performance
<a href="#">InListRewriteThreshold</a>	Specifies the minimum number of values in an IN-list in order for it to qualify for IN-list rewrite query processing.	Performance
<a href="#">IVMaxWorkloadCache</a>	Defines the maximum size of the workload cache when performing validation operations.	Performance
<a href="#">JournalDBSize</a>	Determines the maximum size of permanent journal table multirow data blocks, in 512-byte sectors. Rows are written to this journal during INSERT, UPDATE, and DELETE processing.	File System
<a href="#">JSON_MaxDepth</a>	Determines the level of JSON value nesting allowed during JSON syntax validation.	General
<a href="#">JSON_AttributeSize</a>	Denotes the size of attributes returned by the JSONExtractValue method and dot notation.	General
<a href="#">LargeDepotCylsPerPdisk</a>	Determines the number of Depot cylinders the file system allocates to contain large slots (1 MB). A large slot can hold several data blocks during Depot operations.	File System
<a href="#">LegacySpaceAcctg</a>	Determines whether Vantage uses legacy or global space accounting.	General
<a href="#">LimitInlistCVal</a>	Limits the number of combined values in the IN list.	Performance
<a href="#">LockLogger [Deprecated]</a>	Defines the system default for the lock logger.	General
<a href="#">LockLogger Delay Filter [Deprecated]</a>	Controls whether or not to filter out blocked lock requests based on delay time.	General
<a href="#">LockLogger Delay Filter Time [Deprecated]</a>	Indicates the value at which blocked lock requests with a delay time greater than this value are logged.	General
<a href="#">LockLoggerFlushRate</a>	Determines the frequency for writing entries to the DBQLXMLLOCKTBL cache.	General
<a href="#">LockLogSegmentSize [Deprecated]</a>	Specifies the size of the Lock Logger segment.	General

Field	Description	Group
<a href="#">MaxDecimal</a>	Defines the maximum number of Decimal Digits in the default maximum value used in expression typing.	General
<a href="#">MaxDownRegions</a>	Determines the number of regions (ranges of rows) in a data or index subtable that can be marked as down before the entire subtable is marked down on all AMPs.	General
<a href="#">MaxJoinTables</a>	Influences the maximum number of tables that can be joined per query block.	Performance
<a href="#">MaxLoadAWT</a>	Works together with MaxLoadTasks to limit the number of load utility jobs that can run concurrently on the system.	General
<a href="#">MaxLoadTasks</a>	Works together with MaxLoadAWT to limit the number of load utility jobs that can run concurrently on the system.	General
<a href="#">MaxMLOADXAWT</a>	Works together with MaxMLOADXTasks to limit the number of MLOADX jobs that can run concurrently on the system.	General
<a href="#">MaxMLOADXTasks</a>	Works together with MaxMLOADXAWT to limit the number of MLOADX jobs that can run concurrently on the system.	General
<a href="#">MaxNumInListRewrite</a>	Specifies the maximum number of IN-lists in a single query that can use IN-list rewrite query processing.	Performance
<a href="#">MaxParseTreeSegs</a>	Defines the maximum number of tree segments that the parser allocates while parsing a request.	Performance
<a href="#">MaxRequestsSaved</a>	Specifies the number of request-to-step cache entries allowed on each PE on a Teradata system.	Performance
<a href="#">MaxRowHashBlocksPercent</a>	Specifies the proportion of available locks that can be used for rowhash locks by a transaction before the transaction is automatically aborted.	General
<a href="#">MaxSetQueryBandSize</a>	Defines the maximum query band size in kilobytes that can be set with the SET QUERY_BAND statement.	General
<a href="#">MaxSyncWALWrites</a>	Determines the maximum number of outstanding WAL log writes to allow before tasks requiring synchronous writes are delayed to achieve better buffering.	File System
<a href="#">Memory Limit Per Transaction</a>	Specifies the maximum amount of in-memory, temporary storage that the Relay	General

Field	Description	Group
	Services Gateway (RSG) can use to store the records for one transaction.	
<a href="#">MergeBlockRatio</a>	Specifies the maximum size of a new data block that results from the merging of several smaller blocks. The size is expressed as the ratio of the new block size to the maximum block size specified in the table definition or by the PermDBSize field in DBS Control.	File System
<a href="#">MetaDataCapture</a>	Specifies the maximum size of a new data block that results from the merging of several smaller	General
<a href="#">MinDBSectsToCompress</a>	Specifies the minimum size DB that will be compressed. DBs smaller than this size will not be compressed.	Compression
<a href="#">MiniCylPackHighWaterMark</a>	Determines the number of free cylinders above which MiniCylPack is not active.	File System
<a href="#">MiniCylPackLowCylProd</a>	Determines the number of free cylinders below which the File System will start to perform the MiniCylPack operation in anticipation of the need for additional free cylinders.	File System
<a href="#">MinPercentCompReduction</a>	Specifies the minimum percentage by which the DB size must be reduced by compression. If compression will not reduce the DB by this amount, the DB will not be compressed.	Compression
<a href="#">MLoadDiscardDupRowUPI</a>	Determines whether the DBS MultiLoad protocol should silently discard rows with UPI violations or log these rows to the application error table.	General
<a href="#">MLOADXUtilityLimits</a>	Determines whether MLOADX concurrent task and AWT limits are specific to MLOADX or are shared with other load utilities (FastLoad, MultiLoad, and FastExport).	General
<a href="#">MonSesCPUNormalization</a>	Determines whether CPU data in workload management (PM/API and Open API) calls is normalized. This affects workload rules defined in Teradata Viewpoint Workload Designer portlet.	General
<a href="#">MPS_IncludePEOnlyNodes</a>	Excludes PE-only (AMP-less) nodes from MONITOR PHYSICAL SUMMARY Workload Management API statistics calculations.	General

Field	Description	Group
<a href="#">NameValidationRule</a>	Determines the rules used by Vantage to validate object names.	General
<a href="#">NewHashBucketSize</a>	Specifies the number of bits used to identify hash buckets on the system after the next system initialization or reconfiguration.	General
<a href="#">NoDot0Backdown</a>	Enables several point release features which, if enabled, prevent a system from being backed down to the prior ("dot 0") release without a full system initialization (sysinit).	General
<a href="#">NumStatisticsCacheSegs</a>	Specifies the number of segments maintained for the Optimizer statistics cache on each PE vproc.	Performance
<a href="#">ObjectUseCountCollectRate</a>	Specifies the amount of time between collections of object use-count data.	General
<a href="#">OverrideARCBLC</a>	Determines whether database tables restored from archives are block-level compressed according to the system-level BLC defaults.	Compression
PERMANENT JOURNAL Temperature see <a href="#">Storage Group</a>	Determines the initial temperature assigned to data loaded or inserted into tables of this type.	Storage
<a href="#">PermDBAllocUnit</a>	Determines the allocation unit for multirow data blocks in units of 512-byte sectors for permanent tables.	File System
<a href="#">PermDBSize</a>	Determines the maximum size, in consecutive 512-byte sectors, of the multirow data blocks of a permanent table.	File System
PERM Fallback Temperature see <a href="#">Storage Group</a>	Determines the initial temperature assigned to data loaded or inserted into tables of this type.	Storage
PERM Fallback CLOB Temperature see <a href="#">Storage Group</a>	Determines the initial temperature assigned to data loaded or inserted into tables of this type.	Storage
PERM Primary Temperature see <a href="#">Storage Group</a>	Determines the initial temperature assigned to data loaded or inserted into tables of this type.	Storage
PERM Primary CLOB Temperature see <a href="#">Storage Group</a>	Determines the initial temperature assigned to data loaded or inserted into tables of this type.	Storage
<a href="#">PMPC_SessionRateThreshold</a>	Determines the threshold above which the session cache is refreshed on demand, based on the session rate and last	General



Field	Description	Group
	collection time. Only effective when TDWM Workloads are enabled.	
<a href="#">PMPC_TimeoutSecs</a>	Sets timeout interval for certain supported PM API commands.	General
<a href="#">PPICacheThrP</a>	Specifies the maximum amount of memory to be used for disk read operations that involve multiple partitions.	Performance
<a href="#">PrimaryIndexDefault</a>	Determines whether Vantage automatically creates primary indexes for tables and join indexes created by CREATE TABLE statements that do not include PRIMARY INDEX, PRIMARY AMP [INDEX], NO PRIMARY INDEX, PRIMARY KEY, UNIQUE, and PARTITION BY clauses.	General
<a href="#">ProratedSpaceDistPercent</a>	Specifies the percentage of newly added space that will be divided evenly and distributed to every database and user during a system reconfiguration.	General
<a href="#">PRPDMaxSkewedVals</a>	The maximum number of skewed values in a join that should be considered for a PRPD optimization plan.	Performance
<a href="#">PRPDSkewPct</a>	The percentage of rows in a join to be considered for skew by the optimizer, which determines whether PRPD optimization is used.	Performance
<a href="#">ReadAhead</a>	Enables or disables the performance enhancements associated with the Read-Ahead Sequential File Access Workload operation.	Performance
<a href="#">Read Ahead Count</a>	Specifies the number of data blocks that are preloaded in advance of the current file position while performing sequential scans.	Performance
<a href="#">ReadLockOnly</a>	Enables or disables the special read-or-access lock protocol on the DBC. AccessRights table during privileges validation and on other dictionary tables accessed by read-only queries during request parsing.	Performance
<a href="#">RedistBufSize</a>	Determines the size (in KB) of units of hashed row redistribution buffers for use by load utilities.	Performance
<a href="#">RedriveDefaultParticipation</a>	Determines whether or not the Redrive feature is used for all sessions by default.	General

Field	Description	Group
<a href="#">RedriveProtection</a>	Determines system-level support for the Redrive feature.	General
<a href="#">RepCacheSegSize [Deprecated]</a>	Do not use.	General
<a href="#">RequestDuration</a>	Controls how long Vantage waits for the completion of a request to an external file system after the request timeout or connection timeout has been exceeded and request retries have been commenced.	Native Object Store
<a href="#">RequestTimeout</a>	Controls how long Vantage waits for an external file system request response before aborting the request, re-establishing the connection, and issuing the request again.	Native Object Store
<a href="#">RevertJoinPlanning</a>	Determines whether the Teradata query Optimizer uses newer or older join planning techniques.	Performance
<a href="#">RollbackPriority</a>	Defines the default priority given to rollback operations.	General
<a href="#">RollForwardLock</a>	Defines the system default for the RollForward using Row Hash Locks option.	General
<a href="#">RoundHalfwayMagUp</a>	Indicates how rounding should be performed when computing values of DECIMAL types.	General
<a href="#">RoundNumberAsDec</a>	Determines whether NUMBER data type uses RoundHalfwaysMagUp rounding or will be rounded away from zero.	General
<a href="#">SessionMode</a>	Defines the system default transaction mode, case sensitivity, and character truncation rule for a session.	General
<a href="#">ShowAllUserNames</a>	Determines whether user names, including unauthenticated user names, are shown in Vantage logs	General
<a href="#">SHOW IN XML Formatted Output</a>	Controls the formatting of what is returned by the SHOW statement when the IN XML option is used.	General
<a href="#">SHOW IN XML Memory Limit</a>	Sets the maximum amount of memory in kilobytes used for internal processing of a request of SHOW statement when the IN XML option is used.	Performance
<a href="#">SkewAllowance</a>	Specifies a percentage factor used by the optimizer in deciding the size of each hash join partition.	Performance

Field	Description	Group
<a href="#">SmallDepotCylsPerPdisk</a>	Determines the number of Depot cylinders the file system allocates to contain small slots (1 MB). A small slot can hold a single data block during Depot operations.	File System
<a href="#">SnapBypassAggrCache</a>	Specifies whether snapshot dumps are suppressed if resource exhaustion is detected when reserving memory for aggregate processing.	General
<a href="#">Spill File Path</a>	Specifies a directory that the Relay Services Gateway (RSG) can use for spill files.	General
SPOOL Temperature see <a href="#">Storage Group</a>	Determines the initial temperature assigned to data loaded or inserted into tables of this type.	Storage
<a href="#">SpoolBigDBAllocUnit</a>	Determines the allocation unit in sectors for multirow big data blocks in spool tables.	File System
<a href="#">StandAloneReadAheadCount</a>	Specifies the number of data blocks the Teradata utilities will preload when the utilities or File System startup run as standalone tasks.	Performance
<a href="#">StepsSegmentSize</a>	Defines the maximum size (in KB) of the plastic steps segment (also known as OptSeg).	Performance
<a href="#">SyncScanCacheThr</a>	Specifies the percentage of File Segment (FSG) cache that is expected to be available for all synchronized full-file scans occurring simultaneously.	Performance
<a href="#">SysInit</a>	Ensures the system has been initialized properly using the System Initializer utility. Read only.	General
<a href="#">SysChangeThresholdOption</a>	Defines a change threshold option for statistics recollection.	Optimizer Statistics
<a href="#">SysSampleOption</a>	Defines a sample option for statistics recollection.	Optimizer Statistics
<a href="#">SysInitRelease</a>	Specifies the major release number on which the system was last initialized. Read only.	General
<a href="#">System TimeZone Hour</a>	Defines the System Time Zone Hour offset from Universal Coordinated Time (UTC).	General
<a href="#">System TimeZone Minute</a>	Defines the System Time Zone Minute offset from Universal Coordinated Time (UTC).	General

Field	Description	Group
<a href="#">System TimeZone String</a>	Displays the currently set System Time Zone String.	General
<a href="#">Target Level Emulation</a>	Allows a test engineer to set the costing parameters considered by the Optimizer.	General
<a href="#">TempBLCInterval</a>	Specifies the amount of time in minutes that the background TBBLC task waits after compressing or uncompressing data before checking for other data to process.	Compression
<a href="#">TempBLCIOThresh</a>	Specifies the maximum threshold number of outstanding I/O operations that can exist on a node above which the background TBBLC task pauses for a short period	Compression
<a href="#">TempBLCPriority</a>	Specifies the priority at which the background TBBLC compression task runs.	Compression
<a href="#">TempBLCRescanPeriod</a>	The number of days the background TBBLC task waits before revalidating the state of TBBLC data to reconfirm that data is appropriately compressed.	Compression
<a href="#">TempBLCThresh</a>	Specifies the temperature at and below which data blocks are automatically compressed if TBBLC is enabled. When changed, causes TBBLC to immediately start a new cycle.	Compression
<a href="#">TempBLCSpread</a>	Specifies the percentage above and below the TempBLCThresh field setting at which data should be compressed or uncompressed if TBBLC is enabled. When changed, causes TBBLC to immediately start a new cycle.	Compression
<a href="#">TempLargePageSize</a>	Specifies the size (in KB) of the large memory allocation storage page used for Relay Services Gateway (RSG) temporary storage.	General
<a href="#">Temporal Behavior</a>	Determines the default temporal behavior for temporal DDL, queries, and DML statements.	General
<a href="#">Temporary Storage Page Size</a>	Specifies the size (in KB) of the standard memory allocation storage page size used for Relay Services Gateway (RSG) temporary storage.	General
<a href="#">TIM Cache Load Disabled</a>	Determines whether the Teradata Intelligent Memory (TIM) cache preload is enabled. This feature loads the most frequently accessed data into cache during	Storage

Field	Description	Group
	system startup, if the FSG cache was not otherwise preserved.	
<a href="#">TIM Cache Load Throttle</a>	Throttles the Teradata Intelligent Memory (TIM) background cache loading task based on the current number of I/O operations.	Storage
<a href="#">TimeDateWZControl</a>	Determines whether the time and time zone values returned by built in current time and date functions reflect the session time and time zone.	General
<a href="#">TruncRoundReturnTimestamp</a>	Determines whether the TRUNC and ROUND system functions return a TIMESTAMP or DATE value when passed a TIMESTAMP value.	General
<a href="#">UseVirtualSysDefault</a>	This field is no longer used. For cost profiling information, see <a href="#">CostProfileId</a> .	General
<a href="#">UncompressReservedSpace</a>	Specifies the minimum percentage of storage space that must remain available while DBs are uncompressed using the Ferret UNCOMPRESS command.	Compression
<a href="#">UserRetryLimit</a>	Determines the number of times Vantage retries a request to an external file system after the request timeout or connection timeout has been exceeded and request retries have been commenced.	Native Object Store
<a href="#">UtilityReadAheadCount</a>	Specifies the number of data blocks the Teradata utilities preloads when performing sequential scans.	Performance
<a href="#">Version</a>	Indicates the version number of the DBS Control Record.	General
<a href="#">WAL Buffers</a>	Determines the number of WAL append buffers to allocate.	File System
<a href="#">WAL Checkpoint Interval</a>	Determines the amount of time that elapses between WAL checkpoints.	File System
WAL Temperature see <a href="#">Storage Group</a>	Determines the initial temperature assigned to data loaded or inserted into tables of this type.	Storage
<a href="#">WorkDBSize</a>	Specifies the maximum size for multirow data blocks in work tables.	File System
<a href="#">XML_Memory Limit</a>	Specifies the maximum amount of memory available for operations involving the XML data type.	General



## AccessLockForUncomRead

Determines whether SELECT requests embedded in INSERT ... SELECT, UPDATE ... SELECT, and DELETE ... SELECT requests place READ or ACCESS locks on the source table.

### Field Group

General

### Valid Settings

Setting	Default Locking for Outer SELECT and Ordinary SELECT Subqueries	Default Locking for SELECT Embedded In DELETE, INSERT, MERGE, or UPDATE Request
TRUE	READ	ACCESS
FALSE	READ	READ

### Default

FALSE

### Changes Take Effect

After the DBS Control Record has been written.

### Related Information

For more information on...	See...
Locks and concurrency control	<i>Teradata Vantage™ - SQL Request and Transaction Processing</i> , B035-1142.
Session isolation level	<i>Teradata Vantage™ - SQL Request and Transaction Processing</i> , B035-1142.

## 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 <i>Teradata Vantage™ - Database Utilities</i> , B035-1102.
Free Space Percent	<ul style="list-style-type: none"> <li>FreeSpacePercent DBS Control field in <i>Teradata Vantage™ - Database Utilities</i>, B035-1102.</li> <li>Ferret SHOWFSP command in <i>Teradata Vantage™ - Database Utilities</i>, B035-1102.</li> <li><i>Teradata Vantage™ - Database Administration</i>, B035-1093.</li> </ul>



## AutoCylPackFreeCylThresh

The number of free cylinders below which AutoCylPack (automatic background cylinder packing) will not operate to allocate new cylinders.

### Field Group

File System

### Valid Range

20 through 100

### Default

20

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

To avoid conflicts with the MiniCylPack task, which runs only when the number of cylinders is critically low, the value of AutoCylPackFreeCylThresh should be greater than the value of the MiniCylPackLowCylProd field.

### Related Information

For more information on...	See...
AutoCylPack	<a href="#">DisableAutoCylPack.</a>
MiniCylPack	<a href="#">MiniCylPackLowCylProd</a> , <i>Teradata Vantage™ - Database Administration</i> , B035-1093.
Free Space Percent	<a href="#">FreeSpacePercent</a> , <a href="#">SHOWFSP</a> , <i>Teradata Vantage™ - Database Administration</i> , B035-1093.

## AutoCylPackFSP

The percentage of storage space within a cylinder that AutoCylPack (automatic background cylinder packing) operations leave free of data, to allow for future table growth. This value for Free Space Percent (FSP) is used only for tables that do not have FSP otherwise specified.

### Field Group

File System

### Valid Range

10 through 75

### Default

10

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

*Free Space Percent (FSP)* is the amount of space on a cylinder that is unoccupied and available for storage, expressed as a proportion of the total space on the cylinder. For tables that are expected to grow, free space on cylinders allows for table growth without requiring the additional cost of allocating free cylinders.

Vantage can maintain different levels of free space on cylinders used to store different tables. Static tables can be designated with little or no FSP, while tables that change and grow can be designated to have FSP levels that allow for table growth.

AutoCylPack runs periodically as a background task to maintain the set levels of FSP on table cylinders. If there is less than the desired FSP available on the cylinder, AutoCylPack moves some data to other cylinders to free up some space. If there is more than the desired FSP available on the cylinder, AutoCylPack moves data to the cylinder, potentially freeing up other cylinders.

A desired target level for FSP can be specified for tables in several ways. The FSP that AutoCylPack maintains for cylinders of each table depends on how the FSP was specified:

- If the FSP was specified in a CREATE or ALTER TABLE statement, AutoCylPack maintains that FSP for the cylinders of the table.

---

#### Note:

If the FSP was specified in a CREATE or ALTER TABLE statement, and AutoCylPack is enabled, specifying a FREESPACEPERCENT value with a PACKDISK command is not recommended. In such cases, the FSP resulting from the PACKDISK operation would be temporary, and the table would be returned to the original table-level FSP setting by the background AutoCylPack task.

---

- If FSP was not specified with CREATE or ALTER TABLE, but PACKDISK was run on the table, AutoCylPack maintains the table cylinders at the FSP level that PACKDISK used.
- If an FSP was not specified in a CREATE or ALTER TABLE statement for the table, and PACKDISK was never run on the table, AutoCylPack uses the FSP specified by the AutoCylPackFSP field in DBS Control.

## Related Information

For more information on...	See...
AutoCylPackThresh	<a href="#">AutoCylPackThresh</a> .
CREATE TABLE and ALTER TABLE	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.
PACKDISK	<a href="#">PACKDISK</a> .
Free Space Percent	<a href="#">FreeSpacePercent</a> , <a href="#">SHOWFSP</a> , <i>Teradata Vantage™ - Database Administration</i> , B035-1093.

## AutoCylPackInterval

The amount of time in seconds that the AutoCylPack (automatic background cylinder packing) task waits between checking for new AutoCylPack work.

### Field Group

File System

### Valid Range

60 through 600

### Default

60

### Changes Take Effect

After the DBS Control Record has been written.

### Related Information

For more information on...	See...
AutoCylPack	<a href="#">DisableAutoCylPack</a> .
Free Space Percent	<a href="#">FreeSpacePercent</a> , <a href="#">SHOWFSP</a> , <i>Teradata Vantage™ - Database Administration</i> , B035-1093.

# AutoCylPackIOThresh

The number of pending I/O operations, per node, at or above which the AutoCylPack (automatic background cylinder packing) task will pause to allow I/Os to complete. Setting this value to a lower number minimizes the performance impact of AutoCylPack operations.

## Field Group

File System

## Valid Range

1 through 1000

## Default

1

## Changes Take Effect

After the DBS Control Record has been written.

## Related Information

For more information on...	See...
AutoCylPack	<a href="#">DisableAutoCylPack</a> .
Free Space Percent	<a href="#">FreeSpacePercent</a> , <a href="#">SHOWFSP</a> , <i>Teradata Vantage™ - Database Administration</i> , B035-1093.

## AutoCylPackStyle

Determines whether the AutoCylPack (automatic background cylinder packing) task is restricted to packing data, unpacking data, or if it is unrestricted, and can both pack and unpack data to achieve the target free space percentage (FSP) on a cylinder.

The AutoCylPack task can pack cylinder data more tightly, to store more data on fewer cylinders. This can improve the efficiency of performing sequential scans of tables and also frees up cylinders to be used elsewhere.

Alternatively, the AutoCylPack task can unpack cylinder data, to free more space on a cylinder. This can improve the speed of updates to tables occupying those cylinders, at the cost of consuming more cylinders for the given amount of data.

### Field Group

File System

### Valid Settings

Setting	Description
1	AutoCylPack is restricted to packing data more tightly on cylinders.
2	AutoCylPack is unrestricted, so can both pack and unpack data as necessary to achieve the target FSP.
3	AutoCylPack is restricted to unpacking data, arranging it more loosely on and between cylinders.

### Default

2

### Changes Take Effect

After the DBS Control Record has been written.

### Related Information

For more information on...	See...
AutoCylPack	<a href="#">DisableAutoCylPack</a> .
Free Space Percent	<a href="#">FreeSpacePercent</a> , <a href="#">SHOWFSP</a> , <i>Teradata Vantage™ - Database Administration</i> , B035-1093.

## AutoCylPackThresh

The difference in free space percentage between actual FSP on a cylinder and the target AutoCylPack FSP, below which AutoCylPack (automatic background cylinder packing) will not process the cylinder.

**Field Group**

File System

**Valid Range**

5% through 30%

**Default**

5%

**Changes Take Effect**

After the DBS Control Record has been written.

**Related Information**

For more information on...	See...
AutoCylPack	<a href="#">DisableAutoCylPack</a> .
Free Space Percent	<a href="#">FreeSpacePercent</a> , <a href="#">SHOWFSP</a> , <i>Teradata Vantage™ - Database Administration</i> , B035-1093.

## AutoProvision

Determines whether unknown external logons (logons with no existing entry in the DBC.Dbase table) are automatically added as new users (CREATE USER) prior to the logon completing.

### Field Group

General

### Valid Settings

Setting	Description
TRUE	External logons from unknown users cause creation of new users (CREATE USER).
FALSE	External logons from unknown users are logged on as user EXTUSER.

### Default

FALSE

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

All new users created by auto provisioning are created under the special user External\_AP, which logs on to the system only when required to run the CREATE USER command for new external logons.

### Related Information

For more information on auto provisioning of external logons, see *Teradata Vantage™ - Advanced SQL Engine Security Administration*, B035-1100.



## Bkgrnd Age Cycle Interval

Used by the File System to determine the amount of time that elapses between background cycles to write a subset of modified segments in the cache to disk. This background activity serves to reduce the size of the WAL log, and allows for improved disk space utilization in WAL modes.

### Field Group

File System

### Valid Range

1 through 240 seconds

### Default

60 seconds

### Changes Take Effect

After the DBS Control Record has been written.

## BLCStats

Determines whether block-level compression (BLC) compression ratio information is gathered by COLLECT STATISTICS statements, and can be displayed by the SHOW STATISTICS statement.

### Field Group

Optimizer Statistics

### Valid Settings

Setting	Effect
0	Indicates field is set or has been reset to the Teradata default (enabled). Equivalent to a setting of 1.
1	Enabled. BLC statistics are collected if BLC is enabled.
2	Disabled. BLC statistics are not collected.

### Default

0 (enabled)

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

If BLC is enabled but BLC statistics are not wanted, set this field to 2 to reduce overhead of collecting such statistics on tables eligible for BLC.

The Ferret utility SHOWBLOCKS command also provides information on estimated compression ratios for BLC.

### Related Information

For more information on...	See...
COLLECT STATISTICS and SHOW STATISTICS statements	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.
Enabling and disabling block-level compression	<a href="#">BlockLevelCompression</a> .
The Ferret Utility	<a href="#">Ferret Utility (ferret)</a>

## BLCStatsForCDS

Specifies whether the the COLLECT STATISTICS statement calculates the estimated BLC compression factor for tables. The compression factor is calculated as uncompressed size/compressed size.

The BLC compression factor is used to calculate the customer data space (CDS), which is the logical size of customer data: the size of the uncompressed primary data, excluding the size of fallback data.

### Field Group

Optimizer Statistics

### Valid Settings

Setting	Effect
0	Disable BLC compression factor calculation for COLLECT STATISTICS.
1	Enable BLC compression factor calculation for COLLECT STATISTICS.

### Default

0 (disabled)

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

To enable the CDS feature, set this field to 1, [BlockLevelCompression](#) to ON, and [BLCStats](#) to 0.

### Related Information

For more information on...	See...
CDS macros that let you find the CDS sizes of objects, databases, and systems	<i>Teradata Vantage™ - SystemFE Macros</i> , B035-1103.
Enabling and disabling block-level compression	<a href="#">BlockLevelCompression</a> .

## BlockLevelCompression

Indicates whether the block-level compression (BLC) feature is enabled. This feature allows data blocks (DBs) to be compressed for different categories of tables, subject to other settings in the Compression group of DBS Control fields.

### Field Group

Compression

### Valid Setting

Setting	Description
On	BLC feature is enabled. DBs from different categories of table may or may not be compressed, subject to the other settings in the Compression group.
Off	BLC feature, including temperature-based BLC, is disabled for all categories of tables. Newly created data blocks will not be compressed. Existing blocks that have already been compressed will remain compressed. They can be uncompressed using the Ferret utility UNCOMPRESS command. The Ferret COMPRESS command is disabled.

### Default

OFF

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

BLC enables data compression at the data block (DB) level of the Teradata file system. Compression reduces the amount of storage required for a given amount of data. The BlockLevelCompression field of DBS Control enables and disables BLC.

BLC increases CPU utilization for dynamic compression and decompression operations. This can cause certain operations on compressed tables to use considerably more CPU resources than similar operations on noncompressed tables. Hardware-based BLC (using a compression engine board in every system node) has only a minimal effect on CPU utilization. Unless the system is CPU-rich, these operations can impact other workloads, and could lengthen elapsed response times. However, in some situations, BLC can improve database performance due to reduced I/O bandwidth.

BlocklevelCompression also affects tables that were loaded using the BlockCompression query band to specify data block compression. If BlockLevelCompression is Off, data added to these tables will not be compressed.

The PermDBSize and JournalDBSize field settings in DBS Control apply to the uncompressed size of these types of DBs.

**Related Information**

For more information on...	See...
Block-Level Compression	<i>Teradata Vantage™ - Database Design</i> , B035-1094.
COMPRESS and UNCOMPRESS commands	<a href="#">Ferret Utility (ferret)</a> .
BLOCKCOMPRESSION query band	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.

## CDCProtocol

Determines whether the Change Data Capture (CDC) protocol used in database replication scenarios allows for user-defined replication groups or reserved system-defined replication groups for Teradata® Unity™.

### Field Group

General

### Valid Settings

Setting	Description
0	Legacy CDC Traditional replication with user-defined replication groups. This is the default.
1	Teradata® Unity™ CDC Replication controlled by Teradata® Unity™, using system-reserved replication groups created dynamically by Vantage.

### Default

0

### Changes Take Effect

After the next database restart.

### Usage Notes

Teradata Unity is an integrated product that enables users to orchestrate a multi-system Teradata environment with load balancing, high system availability, workload optimization, and disaster recovery.

### Related Information

- For more information on related DBS Control fields, see [ClientResetAction](#) and [MetaDataCapture](#).
- For more information on replication and Teradata® Unity™, see *Teradata® Unity™ User Guide*, B035-2520.

## Century Break

Defines how to interpret character data when converting to date. The data only has two digits representing the year, and the applicable format only has two digits representing the year. Century Break specifies which two-digit years, if any, are to be interpreted as 20th-century years and which two-digit years, if any, are to be interpreted as 21st-century years.

### Field Group

General

### Valid Range

0 through 100

### Default

0

### Changes Take Effect

When the DBS Control Record is written. For sessions logged on at the time of a change, the new setting becomes effective at the next logon, or after the next database restart.

### Usage Notes

The following two-year digit rules apply:

IF a two-digit year yy is...	THEN the year is...
less than Century Break	20yy and is considered to be in the 21st century.
greater than or equal to Century Break	19yy and is considered to be in the 20th century.

The following Century Break value rules apply:

IF Century Break is...	THEN all years yy are...
0	19yy.
100	20yy.

Century Break does not affect four-digit years.

The Century Break setting has no effect on Vantage dates input in numeric (as opposed to character) format.

If the character data specifies a four-digit year, and the format specifies a two-digit year, Century Break does not affect the conversion. The four digits in the character data are used as the year.

Teradata recommends you convert to four-digit years and corresponding four-digit-year formats. However, Century Break provides a transitional facility while you use two-digit years.

**Example: Using the Century Break DBS Control field**

If Century Break = 25, strings such as '00/01/01' and '24/01/01' are interpreted as years 2000 and 2024, respectively. A string inserted as '25/01/01' is interpreted as year 1925.

---

**Note:**

The choice of 25 for Century Break indicates that the installation wants a cushion of up to 25 years to handle input dates into the 21st century and does not have historic input data prior to the year 1925.

---

If Century Break = 100, two-digit years are inserted as years in the 21st century (that is, 2000, 2001, and continuing on in this pattern).



## ChecktableTableLockRetryLimit

Specifies the duration, in minutes, that CheckTable, in nonconcurrent mode, will retry a table check when the table is locked by another application.

### Field Group

General

### Valid Range

0 through 32,767 minutes

If the CheckTable Table Lock Retry Limit field is greater than 0, then CheckTable will retry a table check within the specified limit.

### Default

0, which indicates that in nonconcurrent mode, CheckTable will retry a table check until CheckTable can access the table.

### Changes Take Effect

After the DBS Control record has been written

## ClientResetAction

Controls what happens to queries when Teradata® Unity™ is disconnected or disabled.

### Note:

This field is effective only when the CDCProtocol field is set to enable Teradata® Unity™ Change Data Capture protocol.

### Field Group

General

### Valid Settings

Setting	Description
0	Continue Queries continue to run as normal, but are not replicated.
1	Abort Aborts queries that were in-process or queued.

### Default

0

### Changes Take Effect

After the next database restart.

### Usage Notes

Teradata Unity is an integrated product that enables users to orchestrate a multi-system Teradata environment with load balancing, high system availability, workload optimization, and disaster recovery.

### Related Information

- For more information on related DBS Control fields, see [CDCProtocol](#) and [MetaDataCapture](#).
- For more information on replication and Teradata® Unity™, see *Teradata® Unity™ User Guide*, B035-2520.

## Client Reset Timeout

Specifies how long the Relay Services Gateway (RSG) should wait for an intermediary to reconnect after one of the following before taking action:

- Communication failure
- Intermediary reset
- Server reset

### Field Group

General

### Valid Range

0 through 65535 seconds

### Default

300 seconds

### Changes Take Effect

After the next database restart.

## CompressGlobalTempFallbackDBs

Specifies conditions under which Global Temporary Table data blocks (DBs) in fallback subtables will be compressed by default.

### Note:

Compression is subject to the [BlockLevelCompression](#) field setting, and to the criteria specified by the [MinPercentCompReduction](#) and [MinDBSectsToCompress](#) DBS Control fields. DBs not meeting these minimum criteria will not be compressed.

### Field Group

Compression

### Valid Settings

Setting	Description
ALWAYS	The DBs of a new table will be compressed regardless of Query Band options specified at load time.
ONLYIFQBYES	The DBs of a new table will not be compressed unless a Query Band used at load time specified to compress them.
UNLESSQBNO	The DBs of a new table will be compressed unless a Query Band used at load time specified not to compress them.
NEVER	The DBs of a new table will not be compressed regardless of any Query Band used at load time. The Ferret utility COMPRESS command will have no effect.

### Default

NEVER

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

BLC enables data compression at the data block (DB) level of the Teradata file system. Compression reduces the amount of storage required for a given amount of data. The BlockLevelCompression field of DBS Control enables and disables BLC.

BLC can be enabled or disabled system-wide, or for various categories of tables. For Permanent data, these include subtables for primary, fallback, eligible LOB data, and join and hash indexes. Secondary indexes on primary rows and fallback copies of secondary indexes are never compressed.

**Related Information**

For more information on...	See...
BlockLevelCompression field	<a href="#">BlockLevelCompression</a> .
COMPRESS and UNCOMPRESS commands	<a href="#">Ferret Utility (ferret)</a> .
BLOCKCOMPRESSION query band	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.

## CompressGlobalTempFallbackCLOBDBs

Specifies conditions under which Global Temporary Table LOB data blocks (DBs) eligible for compression in fallback tables will be compressed by default

### Note:

Compression is subject to the [BlockLevelCompression](#) field setting, and to the criteria specified by the [MinPercentCompReduction](#) and [MinDBSectsToCompress](#) DBS Control fields. DBs not meeting these minimum criteria will not be compressed.

### Field Group

Compression

### Valid Settings

Setting	Description
ALWAYS	The DBs of a new table will be compressed regardless of Query Band options specified at load time.
ONLYIFQBYES	The DBs of a new table will not be compressed unless a Query Band used at load time specified to compress them.
UNLESSQBNO	The DBs of a new table will be compressed unless a Query Band used at load time specified not to compress them.
NEVER	The DBs of a new table will not be compressed regardless of any Query Band used at load time. The Ferret utility COMPRESS command will have no effect.

### Default

NEVER

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

BLC enables data compression at the data block (DB) level of the Teradata file system. Compression reduces the amount of storage required for a given amount of data. The BlockLevelCompression field of DBS Control enables and disables BLC.

BLC can be enabled or disabled system-wide, or for various categories of tables. For Permanent data, these include subtables for primary, fallback, eligible LOB data, and join and hash indexes. Secondary indexes on primary rows and fallback copies of secondary indexes are never compressed.

**Related Information**

For more information on...	See...
BlockLevelCompression field	<a href="#">BlockLevelCompression</a> .
COMPRESS and UNCOMPRESS commands	<a href="#">Ferret Utility (ferret)</a> .
BLOCKCOMPRESSION query band	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.

## CompressGlobalTempPrimaryDBs

Specifies conditions under which Global Temporary Table data blocks (DBs) in primary data subtables will be compressed by default.

### Note:

Compression is subject to the [BlockLevelCompression](#) field setting, and to the criteria specified by the [MinPercentCompReduction](#) and [MinDBSectsToCompress](#) DBS Control fields. DBs not meeting these minimum criteria will not be compressed.

### Field Group

Compression

### Valid Settings

Setting	Description
ALWAYS	The DBs of a new table will be compressed regardless of Query Band options specified at load time.
ONLYIFQBYES	The DBs of a new table will not be compressed unless a Query Band used at load time specified to compress them.
UNLESSQBNO	The DBs of a new table will be compressed unless a Query Band used at load time specified not to compress them.
NEVER	The DBs of a new table will not be compressed regardless of any Query Band used at load time. The Ferret utility COMPRESS command will have no effect.

### Default

NEVER

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

BLC enables data compression at the data block (DB) level of the Teradata file system. Compression reduces the amount of storage required for a given amount of data. The BlockLevelCompression field of DBS Control enables and disables BLC.

BLC can be enabled or disabled system-wide, or for various categories of tables. For Permanent data, these include subtables for primary, fallback, eligible LOB data, and join and hash indexes. Secondary indexes on primary rows and fallback copies of secondary indexes are never compressed.



**Related Information**

For more information on...	See...
BlockLevelCompression field	<a href="#">BlockLevelCompression</a> .
COMPRESS and UNCOMPRESS commands	<a href="#">Ferret Utility (ferret)</a> .
BLOCKCOMPRESSION query band	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.

## CompressGlobalTempPrimaryCLOBDBs

Specifies conditions under which Global Temporary Table LOB data blocks (DBs) eligible for compression in primary data subtables will be compressed by default.

### Note:

Compression is subject to the [BlockLevelCompression](#) field setting, and to the criteria specified by the [MinPercentCompReduction](#) and [MinDBSectsToCompress](#) DBS Control fields. DBs not meeting these minimum criteria will not be compressed.

### Field Group

Compression

### Valid Settings

Setting	Description
ALWAYS	The DBs of a new table will be compressed regardless of Query Band options specified at load time.
ONLYIFQBYES	The DBs of a new table will not be compressed unless a Query Band used at load time specified to compress them.
UNLESSQBNO	The DBs of a new table will be compressed unless a Query Band used at load time specified not to compress them.
NEVER	The DBs of a new table will not be compressed regardless of any Query Band used at load time. The Ferret utility COMPRESS command will have no effect.

### Default

NEVER

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

BLC enables data compression at the data block (DB) level of the Teradata file system. Compression reduces the amount of storage required for a given amount of data. The BlockLevelCompression field of DBS Control enables and disables BLC.

BLC can be enabled or disabled system-wide, or for various categories of tables. For Permanent data, these include subtables for primary, fallback, eligible LOB data, and join and hash indexes. Secondary indexes on primary rows and fallback copies of secondary indexes are never compressed.

**Related Information**

For more information on...	See...
BlockLevelCompression field	<a href="#">BlockLevelCompression</a> .
COMPRESS and UNCOMPRESS commands	<a href="#">Ferret Utility (ferret)</a> .
BLOCKCOMPRESSION query band	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.

## CompressionAlgorithm

Specifies the algorithm that is used to compress and decompress data in file system data blocks.

### Field Group

Compression

### Valid Settings

Setting	Description
ZLIB	ZLIB lossless data compression and decompression algorithms.
ELZS_H	Hardware-based Exar Lemple-Ziv-Stac lossless data compression and decompression algorithms. <b>Note:</b> This type of compression requires that compression engine hardware be installed on every system node.
ELZS_S	Software-based Exar Lemple-Ziv-Stac lossless data compression and decompression algorithms. This type of compression is used automatically if ELZS_H has been selected but problems are detected with the hardware compression engine board. <b>Note:</b> Explicitly setting this algorithm in DBS Control is not recommended, except when recovering from a hardware compression engine failure.

### Default

ZLIB

### Changes Take Effect

After the DBS Control Record has been written. Changed setting applies to new data added to tables. Compression for existing data is not changed until that data is modified.

### Usage Notes

BLC enables data compression at the data block (DB) level of the Teradata file system. Compression reduces the amount of storage required for a given amount of data. The BlockLevelCompression field of DBS Control enables and disables BLC.

After a data block has been compressed using a particular algorithm, the data retains that compression, even if the algorithm is subsequently changed. New data added to the table is compressed using the new algorithm. If previously compressed data is updated, which requires uncompressing and recompressing the data, the currently set algorithm is used for the recompression.

Consequently, a single subtable can exhibit a mixture of uncompressed data blocks and data blocks compressed using different compression algorithms, depending on the settings at the time the data was

added to the table. To make the data consistent with respect to compression, use the Ferret COMPRESS command to compress all the data using the currently set algorithm, or use the Ferret UNCOMPRESS command to uncompress all the table data.

## Related Information

For more information on...	See...
BlockLevelCompression field	<a href="#">BlockLevelCompression</a> .
ZLIB data compression algorithm	<a href="http://zlib.net">http://zlib.net</a> .
Lempel-Ziv-Stac data compression algorithm	<a href="http://en.wikipedia.org/wiki/Lempel-Ziv-Stac">http://en.wikipedia.org/wiki/Lempel-Ziv-Stac</a> .
COMPRESS and UNCOMPRESS commands	<a href="#">Ferret Utility (ferret)</a> .
BLOCKCOMPRESSION query band	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.

## CompressionLevel

Determines whether compression operations favor processing speed or degree of data compression.

### Field Group

Compression

### Valid Range

1 through 9

A setting of 1 maximally favors speed of compression operation over degree of data compression. A setting of 9 maximally favors degree of data compression over speed of compression operation.

### Default

1

The default provides a balance between speed of compression operation and degree of data compression.

### Changes Take Effect

After the DBS Control Record has been written. Changed setting applies to new data added to tables. Compression for existing data is not changed until that data is modified.

### Usage Notes

BLC enables data compression at the data block (DB) level of the Teradata file system. Compression reduces the amount of storage required for a given amount of data. The BlockLevelCompression field of DBS Control enables and disables BLC.

### Related Information

For more information on...	See...
BlockLevelCompression field	<a href="#">BlockLevelCompression</a> .
COMPRESS and UNCOMPRESS commands	<a href="#">Ferret Utility (ferret)</a> .
BLOCKCOMPRESSION query band	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.

## CompressionZLIBMethod

Specifies the specific implementation of the ZLIB algorithm that Vantage will use for block-level compression (BLC)

In addition to the standard ZLIB implementation, Vantage supports the Intel Integrated Performance Primitives library (Intel IPP) with ZLIB for BLC. The CompressionZLIBMethod field determines whether BLC takes advantage of advanced instructions for better compression performance on Intel platforms.

### Field Group

Compression

### Valid Settings

Setting	Description
ZLIB	Vantage uses the standard ZLIB library for BLC.
IPPZLIB	Vantage uses the Intel IPP library for ZLIB BLC. If the Intel IPP library is not installed, or an error is encountered, standard ZLIB is used instead.

### Default

IPPZLIB

### Changes Take Effect

After the DBS Control Record has been written. Changed setting applies to new data added to tables. Compression for existing data is not changed until that data is modified.

### Usage Notes

BLC enables data compression at the data block (DB) level of the Teradata file system. Compression reduces the amount of storage required for a given amount of data. The BlockLevelCompression field of DBS Control enables and disables BLC.

Data compressed with IPPZLIB can be uncompressed with either IPPZLIB or standard ZLIB.

### Related Information

For more information on...	See...
BlockLevelCompression field	<a href="#">BlockLevelCompression</a> .
COMPRESS and UNCOMPRESS commands	<a href="#">Ferret Utility (ferret)</a> .
BLOCKCOMPRESSION query band	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.

## CompressMloadWorkDBs

Specifies conditions under which MultiLoad work data blocks (DBs) will be compressed by default. MultiLoad work DBs are built during the acquisition phase of loading, and are used to store the changes that will be made to the permanent table.

### Note:

Compression is subject to the [BlockLevelCompression](#) field setting, and to the criteria specified by the [MinPercentCompReduction](#) and [MinDBSectsToCompress](#) DBS Control fields. DBs not meeting these minimum criteria will not be compressed.

### Field Group

Compression

### Valid Settings

Setting	Description
ALWAYS	The DBs of a new table will be compressed regardless of Query Band options specified at load time.
ONLYIFQBYES	The DBs of a new table will not be compressed unless a Query Band used at load time specified to compress them.
UNLESSQBNO	The DBs of a new table will be compressed unless a Query Band used at load time specified not to compress them.
NEVER	The DBs of a new table will not be compressed regardless of any Query Band used at load time. The Ferret utility COMPRESS command will have no effect.

### Default

NEVER

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

BLC enables data compression at the data block (DB) level of the Teradata file system. Compression reduces the amount of storage required for a given amount of data. The BlockLevelCompression field of DBS Control enables and disables BLC.

BLC can be enabled or disabled system-wide, or for various categories of tables. For Permanent data, these include subtables for primary, fallback, eligible LOB data, and join and hash indexes. Secondary indexes on primary rows and fallback copies of secondary indexes are never compressed.



**Related Information**

For more information on...	See...
BlockLevelCompression field	<a href="#">BlockLevelCompression</a> .
COMPRESS and UNCOMPRESS commands	<a href="#">Ferret Utility (ferret)</a> .
Query Banding	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.

## CompressPermFallbackDBs

Specifies conditions under which MANUAL tables have the data blocks (DBs) of permanent data in their fallback data subtables compressed by default.

MANUAL tables are those created or altered to have BLOCKCOMPRESSION=MANUAL, or those with BLOCKCOMPRESSION=DEFAULT when the DefaultTableMode DBS Control field is set to MANUAL.

### Note:

Compression is subject to the [BlockLevelCompression](#) field setting, and to the criteria specified by the [MinPercentCompReduction](#) and [MinDBSectsToCompress](#) DBS Control fields. DBs not meeting these minimum criteria will not be compressed.

### Field Group

Compression

### Valid Settings

Setting	Description
ALWAYS	The DBs of a new table will be compressed regardless of Query Band options specified at load time.
ONLYIFQBYES	The DBs of a new table will not be compressed unless a Query Band used at load time specified to compress them.
UNLESSQBNO	The DBs of a new table will be compressed unless a Query Band used at load time specified not to compress them.
NEVER	The DBs of a new table will not be compressed regardless of any Query Band used at load time. The Ferret utility COMPRESS command will have no effect.

### Default

ONLYIFQBYES

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

BLC enables data compression at the data block (DB) level of the Teradata file system. Compression reduces the amount of storage required for a given amount of data. The BlockLevelCompression field of DBS Control enables and disables BLC.

BLC can be enabled or disabled system-wide, or for various categories of tables. For Permanent data, these include subtables for primary, fallback, eligible LOB data, and join and hash indexes. Secondary indexes on primary rows and fallback copies of secondary indexes are never compressed.

**Related Information**

For more information on...	See...
BlockLevelCompression field	<a href="#">BlockLevelCompression</a> .
COMPRESS and UNCOMPRESS commands	<a href="#">Ferret Utility (ferret)</a> .
DefaultTableMode field	<a href="#">DefaultTableMode</a>
BLOCKCOMPRESSION query band	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.

## CompressPermFallbackCLOBDBs

Specifies conditions under which MANUAL tables have the data blocks (DBs) of LOB permanent data that is eligible for compression in their fallback data subtables compressed by default.

MANUAL tables are those created or altered to have BLOCKCOMPRESSION=MANUAL, or those with BLOCKCOMPRESSION=DEFAULT when the DefaultTableMode DBS Control field is set to MANUAL.

### Note:

Compression is subject to the [BlockLevelCompression](#) field setting, and to the criteria specified by the [MinPercentCompReduction](#) and [MinDBSectsToCompress](#) DBS Control fields. DBs not meeting these minimum criteria will not be compressed.

### Field Group

Compression

### Valid Settings

Setting	Description
ALWAYS	The DBs of a new table will be compressed regardless of Query Band options specified at load time.
ONLYIFQBYES	The DBs of a new table will not be compressed unless a Query Band used at load time specified to compress them.
UNLESSQBNO	The DBs of a new table will be compressed unless a Query Band used at load time specified not to compress them.
NEVER	The DBs of a new table will not be compressed regardless of any Query Band used at load time. The Ferret utility COMPRESS command will have no effect.

### Default

NEVER

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

BLC enables data compression at the data block (DB) level of the Teradata file system. Compression reduces the amount of storage required for a given amount of data. The BlockLevelCompression field of DBS Control enables and disables BLC.

BLC can be enabled or disabled system-wide, or for various categories of tables. For Permanent data, these include subtables for primary, fallback, eligible LOB data, and join and hash indexes. Secondary indexes on primary rows and fallback copies of secondary indexes are never compressed.

**Related Information**

For more information on...	See...
BlockLevelCompression field	<a href="#">BlockLevelCompression</a> .
COMPRESS and UNCOMPRESS commands	<a href="#">Ferret Utility (ferret)</a> .
DefaultTableMode field	<a href="#">DefaultTableMode</a>
BLOCKCOMPRESSION query band	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.

## CompressPermPrimaryDBs

Specifies conditions under which MANUAL tables have the data blocks (DBs) of permanent data in their primary data subtables compressed by default.

MANUAL tables are those created or altered to have BLOCKCOMPRESSION=MANUAL, or those with BLOCKCOMPRESSION=DEFAULT when the DefaultTableMode DBS Control field is set to MANUAL.

### Note:

Compression is subject to the [BlockLevelCompression](#) field setting, and to the criteria specified by the [MinPercentCompReduction](#) and [MinDBSectsToCompress](#) DBS Control fields. DBs not meeting these minimum criteria will not be compressed.

### Field Group

Compression

### Valid Settings

Setting	Description
ALWAYS	The DBs of a new table will be compressed regardless of Query Band options specified at load time.
ONLYIFQBYES	The DBs of a new table will not be compressed unless a Query Band used at load time specified to compress them.
UNLESSQBNO	The DBs of a new table will be compressed unless a Query Band used at load time specified not to compress them.
NEVER	The DBs of a new table will not be compressed regardless of any Query Band used at load time. The Ferret utility COMPRESS command will have no effect.

### Default

ONLYIFQBYES

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

BLC enables data compression at the data block (DB) level of the Teradata file system. Compression reduces the amount of storage required for a given amount of data. The BlockLevelCompression field of DBS Control enables and disables BLC.

BLC can be enabled or disabled system-wide, or for various categories of tables. For Permanent data, these include subtables for primary, fallback, eligible LOB data, and join and hash indexes. Secondary indexes on primary rows and fallback copies of secondary indexes are never compressed.

**Related Information**

For more information on...	See...
BlockLevelCompression field	<a href="#">BlockLevelCompression</a> .
COMPRESS and UNCOMPRESS commands	<a href="#">Ferret Utility (ferret)</a> .
DefaultTableMode field	<a href="#">DefaultTableMode</a>
BLOCKCOMPRESSION query band	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.

## CompressPermPrimaryCLOBDBs

Specifies conditions under which MANUAL tables have the data blocks (DBs) of LOB permanent data that is eligible for compression in their primary data subtables compressed by default.

MANUAL tables are those created or altered to have BLOCKCOMPRESSION=MANUAL, or those with BLOCKCOMPRESSION=DEFAULT when the DefaultTableMode DBS Control field is set to MANUAL.

### Note:

Compression is subject to the [BlockLevelCompression](#) field setting, and to the criteria specified by the [MinPercentCompReduction](#) and [MinDBSectsToCompress](#) DBS Control fields. DBs not meeting these minimum criteria will not be compressed.

### Field Group

Compression

### Valid Settings

Setting	Description
ALWAYS	The DBs of a new table will be compressed regardless of Query Band options specified at load time.
ONLYIFQBYES	The DBs of a new table will not be compressed unless a Query Band used at load time specified to compress them.
UNLESSQBNO	The DBs of a new table will be compressed unless a Query Band used at load time specified not to compress them.
NEVER	The DBs of a new table will not be compressed regardless of any Query Band used at load time. The Ferret utility COMPRESS command will have no effect.

### Default

NEVER

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

BLC enables data compression at the data block (DB) level of the Teradata file system. Compression reduces the amount of storage required for a given amount of data. The BlockLevelCompression field of DBS Control enables and disables BLC.

BLC can be enabled or disabled system-wide, or for various categories of tables. For Permanent data, these include subtables for primary, fallback, eligible LOB data, and join and hash indexes. Secondary indexes on primary rows and fallback copies of secondary indexes are never compressed.



**Related Information**

For more information on...	See...
BlockLevelCompression field	<a href="#">BlockLevelCompression</a> .
COMPRESS and UNCOMPRESS commands	<a href="#">Ferret Utility (ferret)</a> .
DefaultTableMode field	<a href="#">DefaultTableMode</a>
BLOCKCOMPRESSION query band	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.

## CompressPJDBs

Specifies whether Permanent Journal (PJ) data blocks (DBs) will be compressed by default.

### Note:

Compression is subject to the [BlockLevelCompression](#) field setting, and to the criteria specified by the [MinPercentCompReduction](#) and [MinDBSectsToCompress](#) DBS Control fields. DBs not meeting these minimum criteria will not be compressed.

### Field Group

Compression

### Valid Settings

Setting	Description
ALWAYS	The DBs of a new PJ table will be compressed.
NEVER	The DBs of a new PJ table will not be compressed. The Ferret utility COMPRESS command will have no effect.

### Default

NEVER

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

BLC enables data compression at the data block (DB) level of the Teradata file system. Compression reduces the amount of storage required for a given amount of data. The BlockLevelCompression field of DBS Control enables and disables BLC.

BLC can be enabled or disabled system-wide, or for various categories of tables. For Permanent data, these include subtables for primary, fallback, eligible LOB data, and join and hash indexes. Secondary indexes on primary rows and fallback copies of secondary indexes are never compressed.

### Related Information

For more information on...	See...
BlockLevelCompression field	<a href="#">BlockLevelCompression</a> .
COMPRESS and UNCOMPRESS commands	<a href="#">Ferret Utility (ferret)</a> .
Permanent Journals	<i>Teradata Vantage™ - Database Design</i> , B035-1094.



## CompressSpoolDBs

Specifies conditions under which data blocks (DBs) for spool and volatile tables will be compressed by default.

### Note:

Compression is subject to the [BlockLevelCompression](#) field setting, and to the criteria specified by the [MinPercentCompReduction](#) and [MinDBSectsToCompress](#) DBS Control fields. DBs not meeting these minimum criteria will not be compressed.

### Field Group

Compression

### Valid Settings

Setting	Description
ALWAYS	The DBs of a new Spool table will be compressed.
NEVER	The DBs of a new Spool table will not be compressed. The Ferret utility COMPRESS command will have no effect.
IFNOTCACHED	The DBs of a new table will be compressed if they are not being cached.

### Default

NEVER

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

BLC enables data compression at the data block (DB) level of the Teradata file system. Compression reduces the amount of storage required for a given amount of data. The BlockLevelCompression field of DBS Control enables and disables BLC.

Spool includes Persistent Spool used for the Redrive feature.

### Related Information

For more information on...	See...
BlockLevelCompression field	<a href="#">BlockLevelCompression</a> .
COMPRESS and UNCOMPRESS commands	<a href="#">Ferret Utility (ferret)</a> .

## ConnectionBufferSize

Controls how much memory is allocated for buffering data between Vantage and external data sources. This setting is effective only when the Native Object Store (NOS) feature is enabled to allow creation of foreign tables.

### Field Group

Native Object Store

### Valid Settings

1 to 16 MB

### Default

16 MB

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

*Foreign tables* allow Vantage to access data from external, cloud-based data storage, such as AWS S3, without requiring you to manually move the data into the database first from where it natively resides. Foreign tables are identified by a hostname, path, and other metadata that point to the external storage. Vantage can read and process semi-structured or unstructured external data in foreign tables using standard SQL. For example, you can use Teradata analytic functions to examine the data, join it to the relational data in the database, and issue queries against it as you can for other data in Vantage.

### Related Information

The following fields affect the NOS feature and foreign table access.

- [EnableNOS](#)
- [ConnectionTimeout](#)
- [DefaultRowGroupSize](#)
- [RequestDuration](#)
- [RequestTimeout](#)
- [UserRetryLimit](#)

## ConnectionTimeout

Controls how long Vantage waits for a new connection to an external network to be established when creating or querying a foreign table. This setting is effective only when the Native Object Store (NOS) feature is enabled.

### Field Group

Native Object Store

### Valid Settings

1 to 60 seconds

### Default

40 seconds

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

*Foreign tables* allow Vantage to access data from external, cloud-based data storage, such as AWS S3, without requiring you to manually move the data into the database first from where it natively resides. Foreign tables are identified by a hostname, path, and other metadata that point to the external storage. Vantage can read and process semi-structured or unstructured external data in foreign tables using standard SQL. For example, you can use Teradata analytic functions to examine the data, join it to the relational data in the database, and issue queries against it as you can for other data in Vantage.

### Related Information

The following fields affect the NOS feature and foreign table access.

- [EnableNOS](#)
- [ConnectionBufferSize](#)
- [DefaultRowGroupSize](#)
- [RequestDuration](#)
- [RequestTimeout](#)
- [UserRetryLimit](#)

## CostProfileId

Specifies the cost profile used by Vantage. A *cost profile* is a suite of system-specific values (*cost profile constants*) that Vantage uses for Optimizer costing coefficients, enabling and disabling certain features, and other system settings. A cost profile can apply system-wide or be associated with a PROFILE and apply only to users assigned that PROFILE.

### Note:

This field should be changed only under the direction of Teradata Support Center personnel. Cost profiles should be set up and modified only by Teradata Support Center personnel.

### Field Group

General

### Valid Range

0 through 32760

IF CostProfileId is...	THEN...
0	Vantage uses the default cost profile for the current combination of hardware and database release. The default may differ from release to release, so this field should remain set to zero under most circumstances.
greater than 0	Vantage uses the cost profile identified by the specified number.

### Default

0

### Changes Take Effect

For new sessions after the DBS Control Record has been written. Changing this setting does not cause any running sessions to change cost profiles.

### Usage Notes

The CostProfileId field is preserved across upgrades. If the field is set to anything other than zero, it can result in an inappropriate cost profile being used when the system is upgraded.

## COUNT\_mode

Determines the data type returned by COUNT operations.

### Field Group

General

### Valid Settings

Setting	Description
0	Data type depends on session mode: <ul style="list-style-type: none"> <li>For ANSI session mode, the data type is DECIMAL.</li> <li>For Teradata session mode, the data type is INTEGER.</li> </ul>
1	Data type is NUMBER.
2	Data type is BIGINT.

### Default

0

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

If you experience numeric overflow errors with COUNT operations, try changing this field value to 1 or 2. This can happen for tables with billions of rows, specifically in Teradata session mode if the base table of the COUNT aggregate function exceeds about two billion rows.

### Related Information

For more information about aggregate functions including COUNT, see *Teradata Vantage™ - SQL Functions, Expressions, and Predicates*, B035-1145.



## CPUdate

Determines how column partitions are updated.

### Field Group

General

### Valid Settings

Setting	Description
0	Vantage uses the default setting, currently equivalent to 3.
1	Disables in-place updates. Column partition updates are performed as complex updates using a write lock.
2	An in-place update using a write lock is used if only one ROW-format column partition is updated. Other column partition updates are performed as complex updates using a write lock.
3	In-place updates using a write lock are performed if only ROW-format column partitions are updated. Other column partition updates are performed as complex updates using a write lock. This is the default.
4	An in-place update using a write lock is used if only one ROW-format column partition is updated. In-place updates using an exclusive lock is used if there is more than one ROW-format column partition, but no COLUMN-format column partitions are updated. Other column partition updates are performed as complex updates using a write lock.

### Default

3

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

The default (setting 3) provides the most concurrency for in-place updates if only ROW-format partitions are updated. However, queries using an access lock may encounter partial updates to a row if more than ROW-format column partition is being updated. That is, an update may have been applied to some of the columns but some columns may have not yet been updated.

When querying using an access lock, if you want rows processed by the query to be either non-updated or completely updated (as would occur for a non-columnar table), use setting 2 or 4.

Setting 2 allows for more concurrency while providing in-place updates if only one column partitions is updated and that column partition has ROW format. However, a complex update is performed if multiple ROW-format column partitions are updated. A *complex update* is one where the original row is logically deleted, the updated row is inserted, but the space for the original row is not reclaimed.

Setting 4 allows in-place update if multiple ROW-format column partition are updated (to allow space to be reclaimed), but it is done under an exclusive lock if more than one ROW-format column partition is updated to prevent queries using an access lock from seeing partially updated rows.

Setting 1 can be used to disable in-place updates if there is an unexpected problem with in-place update logic. In this case, contact the Teradata Support Center.

For all settings, a query using an access lock may see the non-updated row or the updated row as can occur for update of a non-columnar table. And, as for a non-columnar table, if the query sees the updated row, that updated row may be rolled back to the original row if the update transaction is aborted.

### **Related Information**

For more information about column partitioning, see *Teradata Vantage™ - SQL Data Definition Language Syntax and Examples*, B035-1144 and *Teradata Vantage™ - Database Design*, B035-1094.

## CurHashBucketSize

Indicates the number of bits used to identify hash buckets in the current system configuration.

### Field Group

General

### Valid Settings

16 or 20

This field is informational only. It cannot be changed.

See also [NewHashBucketSize](#).

## CylindersSavedforPERM

Reserves a specified number of cylinders to be used only for permanent data storage.

### Field Group

File System

### Valid Range

1 through 524,287 cylinders

### Default

10 cylinders

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

Free cylinders are used for permanent, spool, and temporary space in Vantage.

- Permanent (“perm”) space is used for storing data rows of tables.
- Spool space is used to hold intermediate query results or formatted answer sets for queries. Once the query is complete, the spool space is released as free cylinders.
- Temporary (“temp”) space is used for global temporary tables.

All requests for new cylinders are satisfied from the pool of free cylinders, so requests for perm cylinders compete with requests for spool and temp cylinders. If not enough free cylinders are available to meet a request, the request fails with a disk full error.

When the failed request involves perm cylinders, a lengthy rollback can be required, which adversely affects system performance. However, if the failed request involves spool or temp cylinders, very little rollback is generally required. Therefore, it is preferable to run out of spool and temp cylinders before running out of perm cylinders.

CylindersSavedforPERM reserves a number of free cylinders to be used only to fulfill requests for perm cylinders. If this number or fewer free cylinders is available, requests for spool and temp cylinders will fail, but requests for new perm cylinders will not fail unless there are no free cylinders available.

## DATASETAttributeSize

Denotes the size (number of characters) of attributes returned by the DATASET type dot notation.

### Field Group

General

### Valid Range

0 through 32000 characters

### Default

4096

### Changes Take Effect

After the DBS Control Record has been written.

### Related Information

For More Information on...	See...
DATASET data type	<i>Teradata Vantage™ - DATASET Data Type</i> , B035-1198.
Related DBS Control fields	<a href="#">DisableDATASETValidation</a> and <a href="#">EnableDATASET</a> .

## DateForm

Defines whether IntegerDate or ANSIDate formatting is used for a session.

### Field Group

General

### Valid Settings

Setting	Description
0 (IntegerDate)	Columns with DATE and PERIOD(DATE) type values use the format specified in the SDF file. The default format is 'YY/MM/DD', however the SDF file can be customized by a system administrator to change this date format.
1 (ANSIDate)	Date columns are formatted for output as 'YYYY-MM-DD'.

### Default

0 (IntegerDate)

### Changes Take Effect

For new sessions begun after the DBS Control Record has been written. Existing sessions are not affected.

### Usage Notes

DateForm can be overridden at the user level and at the session level (at logon or during the session).

### Related Information

For more information on...	See...
The SDF file	<a href="#">Teradata Locale Definition Utility (tdlocaledef)</a> .
DateForm	<i>Teradata Vantage™ - SQL Fundamentals</i> , B035-1141.

## DBQL CPU/IO Collection

Sets the algorithm used by DBQL to collect CPU and input/output statistics.

DBQL CPU/IO collection captures performance data for the execution of a query at the request level and, optionally, at the step level. It includes several options for specifying the data that is collected. See also the information about the BEGIN QUERY LOGGING statement in *Teradata Vantage™ - SQL Data Definition Language Syntax and Examples*, B035-1144.

### Field Group

General

### Valid Settings

Setting	Description
0	Indicates DBQL CPU/IO Collection has not yet been explicitly set. In this case, DBQL uses the default collection algorithm.
1	DBQL uses Algorithm 1, the classic algorithm with step adjustments.
2	DBQL uses AMP Algorithm 2. For most situations, this algorithm provides improved collection over Algorithm 1. Algorithm 2 is used primarily for certain types of diagnostic analysis.
3	DBQL uses Algorithm 3. Captures data for aborted and parallel steps, step instances for iterative steps, and includes additional information and status about step and request execution over Algorithms 1 and 2. This is the default.

### Default

3

### Changes Take Effect

After the DBS Control Record has been written.

## DBQL\_AWTDPS\_CacheMaximum

Limits the size to which the per-AMP AWT DBQL Performance Stats (DPS) cache can grow. This cache holds DBQL request and step data for active requests.

### Field Group

Performance

### Valid Range

1024 to 16384 KB

1024 KB holds approximately 2,500 cache entries.

### Default

8192 KB

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

- When the cache has grown to the set maximum size, no further DBQL AMP data is cached, and the accuracy of query performance statistics can degrade.
- If you reduce the size of the cache maximum to something below the current actual size of the cache, the current cache will neither shrink nor grow, but space in the current cache can be dynamically reused. The memory currently allocated to the cache is not released until the next database restart.
- If you see a 9666 information event logged to /var/log/messages, you should increase the value of this field.
- Teradata recommends you increase the value of this field in 4096 KB (4 MB) increments.

### Related Information

For more information on...	See...
BEGIN QUERY LOGGING, END QUERY LOGGING, and FLUSH QUERY LOGGING statements	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.
tracking processing behavior with the DBQL	<i>Teradata Vantage™ - Database Administration</i> , B035-1093.



## DBQLDefCacheSize

Sets the DBQL cache size and determines whether DBQL uses 64K or 1MB data blocks.

### Note:

The size of the DBQLParamTbl cache, populated when you begin query logging with the PARAMINFO option, is controlled separately by the [DBQLLOBCacheSize](#) DBS Control field.

### Field Group

Performance

### Valid Range

0 through 16 MB

A value of zero sets the DBQL cache size to 64 K, and causes DBQL to use 64 K data blocks. Any other value causes DBQL to use 1 MB data blocks.

### Default

2

### Changes Take Effect

After the next database restart.

### Usage Notes

- Increasing the DBQL cache size can reduce the overhead associated with frequent cache flushing. This can be beneficial on extremely busy systems that experience resource contention or exhaustion.
- Changing DBQLDefCacheSize invalidates the contents of the DBQL caches at the next database restart. Teradata recommends that you flush all DBQL caches prior to restarting the database. Use the following SQL statement:

```
FLUSH QUERY LOGGING WITH ALL;
```

### Related Information

For more information on...	See...
BEGIN QUERY LOGGING, END QUERY LOGGING, and FLUSH QUERY LOGGING statements	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.
tracking processing behavior with the DBQL	<i>Teradata Vantage™ - Database Administration</i> , B035-1093.
DBQLLOBCacheSize field	<a href="#">DBQLLOBCacheSize</a>

## DBQLFlushRate

Defines the interval in seconds between flushings of the DBQL caches to the DBQL dictionary tables.

### Field Group

General

### Valid Range

1 through 3600 seconds

### Note:

Teradata does not recommend values less than 600 seconds, and DBS Control issues a warning if you set the value below 600 seconds.

### Default

600 seconds

### Changes Take Effect

After the DBS Control Record has been written. However, DBQL will not become aware of the new setting until the current timer expires (or 10 minutes passes). Therefore, the change could take up to 10 minutes to become effective.

### Usage Notes

If an END QUERY LOGGING statement is issued, all the caches (except the DBQLSummary cache, which is flushed at the selected Flush Rate) are flushed as part of the END QUERY LOGGING statement.

If a FLUSH QUERY LOGGING statement is issued, the caches requested are flushed. To flush all the caches, use the All option in FLUSH QUERY LOGGING.

### Example: Using the DBQLFlushRate DBS Control field

Assume that the DBQLFlushRate is 300 seconds. This means that the cache entries are written to the DBQL dictionary tables at least every 5 minutes. If a cache is filled up after 3 minutes, entries are written at 3 minutes and then again at the 5-minute interval.

### Related Information

For more information on...	See...
BEGIN QUERY LOGGING, END QUERY LOGGING, and FLUSH QUERY LOGGING statements	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.
tracking processing behavior with the DBQL	<i>Teradata Vantage™ - Database Administration</i> , B035-1093.

See also [LockLoggerFlushRate](#).

## DBQLLOBCacheSize

Determines the amount of cache (in megabytes) allocated for DBQL Data Dictionary tables that have LOB columns. This field also determines the size of the express message used to flush this cache.

### Note:

Teradata recommends leaving this field set to the default value.

### Field Group

Performance

### Valid Range

2 through 16 MB

### Default

2 MB

### Changes Take Effect

After the next database restart.

### Usage Notes

Increasing the cache size allows larger requests to be captured by DBQL, and reduces the number of cache flushes. However, it will increase the express request message size when flushing the cache.

### Related Information

For more information on...	See...
BEGIN QUERY LOGGING, END QUERY LOGGING, and FLUSH QUERY LOGGING statements	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.
tracking processing behavior with the DBQL	<i>Teradata Vantage™ - Database Administration</i> , B035-1093.

## DBQLLogLastResp

Determines whether DBQL logs a pseudo step labeled “RESP” in the DBQLStepTbl table when the last response to a DBQL logged request is completed. This can be used together with the FirstRespTime log entry in the DBQLLogTbl to calculate the approximate response time experienced by the client.

### Field Group

General

### Valid Settings

Setting	Description
TRUE	DBQL logs a pseudo step labeled RESP to the DBQLStepTbl when the last response of a DBQL logged request is complete.
FALSE	DBQL does not log a RESP pseudo step for completed requests.

### Default

FALSE

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

This field is effective only when DBQL logging is enabled. The RESP pseudo step entry is logged for every query that is logged to the DBQL default table, regardless of whether STEPINFO or other DBQL options are requested.

When enabled, the RESP pseudo step is logged even if the request aborts or experiences an error. It indicates when the logged request ends for any reason.

### Related Information

For more information on...	See...
BEGIN QUERY LOGGING and END QUERY LOGGING statements	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.
tracking processing behavior with the DBQL	<i>Teradata Vantage™ - Database Administration</i> , B035-1093.

## DBQL Options

This field is unused, but reserved for future DBQL use. It should not be changed.

## DBQLXMLPlanMemLimit

Limits the amount of memory used for generating an XML version of a query plan when query logging is enabled using the WITH XMLPLAN option of BEGIN QUERY LOGGING.

### Field Group

Performance

### Valid Range

256 through 65536 KB

### Default

8192 KB

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

The WITH XMLPLAN option of the BEGIN QUERY LOGGING statement causes the query plan for every non-cached DDL, DML, and DCL SQL statement to be logged as an XML document in the Database Query Log (DBQL). This document is stored in the DBC.DBQLXMLTbl system table.

DBQLXMLPlanMemLimit limits the amount of memory used to generate XML for each SQL request. Reducing the amount of memory used to generate XML plans can free up memory for other tasks, however, setting DBQLXMLPlanMemLimit to a value smaller than the default risks that generated XML plan documents will be incomplete.

EXPLAIN, INSERT EXPLAIN, and DUMP EXPLAIN statements are not logged.

### Related Information

For more information on query logging, see *Teradata Vantage™ - Database Administration*, B035-1093 and *Teradata Vantage™ - SQL Data Definition Language Syntax and Examples*, B035-1144.

## DBSCacheCtrl

Enables or disables preferential caching for data blocks from smaller tables. The threshold size for tables whose data blocks are to be preferentially cached is set using the DBSCacheThr field.

### Field Group

Performance

### Valid Settings

Setting	Description
TRUE	Enables preferential caching.
FALSE	Disables preferential caching.

### Default

TRUE

### Changes Take Effect

After the DBS Control Record has been written. Operations in progress at the time of the change are not affected.

### Related Information

See [DBSCacheThr](#).



## DBSCacheThr

Specifies the threshold table size that demarcates “small” from “large” tables for purposes of system caching decisions. DBSCacheThr is expressed as a percentage of FSG Cache.

- Tables that would occupy the DBSCacheThr percentage or less of the cache are considered small. Data blocks from these tables are preferentially cached.
- Tables that would occupy more than the DBSCacheThr percentage of the cache are considered large. Data blocks from these tables are preferentially excluded from the cache.

DBSCacheThr is effective only when DBSCacheCtrl is set to TRUE.

### Field Group

Performance

### Valid Range

0 through 100%

### Default

10%

### Changes Take Effect

After the DBS Control Record has been written. Any operations in progress at the time of the change are not affected.

### Usage Notes

Caching frequently accessed data blocks can improve system performance because reading from memory (cache) is much faster than reading from disk. Because the FSG Cache size is limited, older data blocks are removed from the cache (“aged out”) as space is required to cache more recently accessed data.

Reference tables are typically accessed frequently. Caching data from these tables can markedly improve system performance. Because reference tables are often relatively small, data from several reference tables will fit into the cache.

Full-table scans of large, non-reference tables, however, if cached, can quickly overwhelm the cache, displacing all existing cached data. If full table scans of large tables are infrequent occurrences, as is typical, there is little benefit to caching this data.

DBSCacheThr provides a way to exclude data from larger tables from the FSG Cache, helping to ensure that data from smaller tables is retained in the cache as long as possible.

DBSCacheThr specifies the proportion of the FSG Cache that may be occupied by a table so that the table data be cached. Because it is a threshold value, tables equal to or smaller than DBSCacheThr will be preferentially cached. Larger tables will not be cached, under most circumstances.

**Note:**

DBSCacheThr is one of a number of factors that influence whether a data block is cached. It does not solely determine which data blocks are cached.

**Considerations**

- DBSCacheThr affects the caching of spool tables in addition to permanent data tables. If typical system work produces large spool tables, setting DBSCacheThr to a small value might prevent spool tables from being cached. This would slow query performance.
- Large tables that would normally be excluded from the cache by DBSCacheThr may qualify for synchronized table scans if two or more queries perform a full table scan on the large table simultaneously. In these cases, data from large tables may be cached, regardless of the DBSCacheThr setting. For more information see [SyncScanCacheThr](#).

**Recommendations**

- Use DBSCacheThr to prevent large, sequentially read or written tables from pushing other data out of the cache. Set DBSCacheThr to a value that corresponds to the demarcation between smaller, more frequently access tables, and larger tables that are infrequently accessed. Ideally, there will be a jump in size between these types of tables which makes distinguishing them easy.
- If moderately sized tables are accessed frequently, setting DBSCacheThr to cache these tables might cause smaller, less frequently accessed tables to be cached, which could impact system performance. Carefully evaluate the performance impacts of any changes to DBSCacheThr before committing those changes on a production system.
- Because DBSCacheThr also affects caching of spool tables, set DBSCacheThr to the smallest possible value that will not adversely affect spool tables generated by the typical system workload. For most moderate to large systems today, this would be a DBSCacheThr setting of 1%. If the average spool table size per node or AMP is greater than 1% of FSG Cache per node or AMP, DBSCacheThr can be set to a higher value.
- To calculate a DBSCacheThr threshold value, determine the size of frequently accessed tables that should be preferentially cached. Assume the table is evenly distributed across all nodes and AMPs of the system, and determine the percentage of cache on each node or AMP that table would occupy. Set DBSCacheThr to this percentage.

For example: DBSCacheThr setting = Per-node table size/FSG Cache per node

**Example: DBSCacheThr field and small, frequently-accessed reference tables**

Assume a system with many small reference tables that are frequently accessed. The goal of the DBSCacheThr setting is to preferentially retain these tables in the cache for as long as possible. Base the DBSCacheThr setting on the size of a typical reference table, and the amount of space such a table occupies on each node of the system:

- Typical small, frequently accessed reference table = 100 MB (one million rows with 100 bytes per row)
- System has 10 nodes

- Vantage distributes the table rows evenly across all AMPs and nodes of the system: 100 MB / 10 nodes = 10 MB per node occupied by the table

RAM per Node	FSG Cache per Node	DBSCacheThr Setting
1GB	500MB	2%
2GB	1.5 GB	1%
4 GB	3.5 GB	1%

These DBSCacheThr settings influence system caching decisions in favor of caching these tables.

### Example: DBSCacheThr field and large tables with full table scans

Assume a system with a workload that requires full-table scans of large tables. The goal of the DBSCacheThr setting is to preferentially exclude these tables from the cache, so that smaller, more frequently access tables will stay in the cache longer. Base the DBSCacheThr setting on the size of a typical large table, and the amount of space such a table occupies on each node of the system:

- Typical large, infrequently accessed table = 1000 MB (10 million rows with 100 bytes per row)
- System has 10 nodes
- Vantage distributes the table rows evenly across all AMPs and nodes of the system: 1000 MB / 10 nodes = 100 MB per node

RAM per Node	FSG Cache per Node	DBSCacheThr Setting
1GB	500MB	less than 20%
2GB	1.5 GB	less than 6%
4 GB	3.5 GB	less than 2%

These DBSCacheThr settings influence system caching decisions in favor of excluding these tables from the cache.

### Related Information

- [DBSCacheCtrl](#)
- [SyncScanCacheThr](#)

## DeadLockTimeout

Determines the interval (in seconds) between deadlock timeout detection cycles. The value in DeadLockTimeout specifies the time-out value for requests that are locking each other out on different AMPs. When the system detects a deadlock, it aborts one of the jobs.

Proxy locks reduce deadlock situations for all-AMP requests that require write or exclusive locks. However, deadlocks still may be an issue on large systems with heavy concurrent usage. In batch operations, concurrent requests may contend for locks on Data Dictionary tables.

### Field Group

General

### Valid Range

10 through 3600 seconds

### Default

240 seconds

### Changes Take Effect

After the next database restart.

### Recommendation

Reduce the value in this field to cause more frequent retries with less time in a deadlock state. Faster CPUs reduce the system overhead for performing deadlock checks, so you can set the value much lower than the current default of 240 seconds.

IF your applications...	THEN you should...
incur some dictionary deadlocks	set the value to between 30 and 45 seconds.
incur few dictionary deadlocks	retain the default value of 240 seconds.
incur many true deadlocks	set the value as low as 10 seconds.
are mostly online transaction processing (tactical) applications	set the value as low as 10 seconds.

### Related Information

For more information on locks and deadlocks, see *Teradata Vantage™ - SQL Request and Transaction Processing*, B035-1142 and *Teradata Vantage™ - Database Administration*, B035-1093.

## DefaultCaseSpec

Determines whether the default behavior for character string comparisons is to consider the case of characters when the transaction semantics are set to Teradata session mode. Also determines whether character column data is treated as case specific by default in Teradata session mode.

### Note:

In ANSI session mode this field has no effect. Character comparisons consider character case, and character columns are treated as case specific by default in ANSI session mode.

### Field Group

General

### Valid Settings

Setting	Description
TRUE	In Teradata session mode, character data is treated as case specific by default. Character string comparisons take into account character case differences.
FALSE	In Teradata session mode, character data is treated as not case specific by default. Character string comparisons do not take into account character case differences.

### Default

FALSE

### Changes Take Effect

After the next database restart.

### Related Information

For more information on the CASESPECIFIC character data type attribute, see *Teradata Vantage™ - Data Types and Literals*, B035-1143.

## Default Character Set

Specifies the default character set assigned to users if a default character set is not explicitly specified in the CREATE USER statement.

### Field Group

General

### Valid Settings

Setting	Description
1	Specifies the Latin character set as the default.
0	Specifies the Unicode character set as the default.

### Default

1 on systems configured to use Standard Language Support mode.

0 on systems configured to use Japanese Language Support mode.

Upgraded systems that have not been reinitialized retain the default character set they had before being upgraded:

- Latin for systems using Standard Language Support Mode
- Unicode for systems using Japanese Language Support Mode

### Changes Take Effect

After the DBS Control Record has been written for CREATE USER statements issued after the change. Existing users retain the default character set they had before Default Character Set was changed.

## DefaultPermSkewLimitPercent

Specifies the percentage of permanent space that an AMP is allowed to use over and above the per-AMP quota (the global maximum space limit divided by the number of AMPs). This is used for databases and users that are created without a SKEW option, or that are created with SKEW limit specified as DEFAULT for permanent space.

### Note:

This field should be changed only under the direction of Teradata Support Center personnel.

### Field Group

General

### Valid Range

0 through 10000 percent

### Default

0

### Changes Take Effect

After the DBS Control record has been written.

### Usage Notes

For operations involving uniform load distribution across all the AMPs in the system, the default skew limit percentage may be set to lower values.

### Related Information

For more information on...	See...
the SKEW option for DDL, such as CREATE DATABASE and CREATE USER	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.
related DBS Control fields	<ul style="list-style-type: none"> <li>• <a href="#">DefaultSpoolSkewLimitPercent</a></li> <li>• <a href="#">DefaultTempSkewLimitPercent</a></li> <li>• <a href="#">GlobalSpaceSoftLimitPercent</a></li> <li>• <a href="#">LegacySpaceAcctg</a></li> </ul>

## DefaultRowGroupSize

### Purpose

Determines the size of RowGroups used while creating Parquet files if a value is not explicitly provided through user query.

### Field Group

Native Object Store

### Valid Settings

4 to 16 MB

### Default

16 MB

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

- There is a chance of higher main memory usage when the row group size is large.
- Main memory usage depends on the number of columns in the table.

### Related Topics

The following fields affect the NOS feature and foreign table access.

- [ConnectionBufferSize](#)
- [ConnectionTimeout](#)
- [EnableNOS](#)
- [RequestDuration](#)
- [RequestTimeout](#)
- [UserRetryLimit](#)



## DefaultSpoolSkewLimitPercent

Specifies the percentage of spool space that an AMP is allowed to use over and above the per-AMP quota (the global maximum space limit divided by the number of AMPs). This is used for databases and users that are created without a SKEW option, or that are created with SKEW limit specified as DEFAULT for spool space.

### Note:

This field should be changed only under the direction of Teradata Support Center personnel.

### Field Group

General

### Valid Range

0 through 10000 percent

### Default

0

### Changes Take Effect

After the DBS Control record has been written.

### Usage Notes

- For operations involving uniform load distribution across all the AMPs in the system, the default skew limit percentage may be set to lower values.
- If the system involves long running transactions with high space consumption, you can increase the value of the GlobalSpaceSoftLimitPercent DBS Control field to reduce the chances of transaction aborts due to a lack of available space.

### Related Information

For more information on...	See...
the SKEW option for DDL, such as CREATE DATABASE and CREATE USER	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.
related DBS Control fields	<ul style="list-style-type: none"> <li><a href="#">DefaultPermSkewLimitPercent</a></li> <li><a href="#">DefaultTempSkewLimitPercent</a></li> <li><a href="#">GlobalSpaceSoftLimitPercent</a></li> <li><a href="#">LegacySpaceAcctg</a></li> </ul>

## DefaultTableMode

Determines how block-level compression (BLC) is managed for Permanent storage tables that have the BLOCKCOMPRESSION option set to DEFAULT, or that do not have the BLOCKCOMPRESSION option explicitly specified when the table is created.

### Field Group

Compression

### Valid Settings

Setting	Description
MANUAL	<p>BLC is applied based on the settings of these DBS Control fields: <a href="#">CompressPermPrimaryDBs</a>, <a href="#">CompressPermFallbackDBs</a>, <a href="#">CompressPermPrimaryCLOBDBs</a>, <a href="#">CompressPermFallbackCLOBDBs</a> which determine the default BLC for these table types at the time the table is created. These defaults can be overridden using the BLOCKCOMPRESSION query band at load time.</p> <p>Tables can be compressed or uncompressed at any time after loading by using the Ferret COMPRESS and UNCOMPRESS commands.</p> <p>Data inserted into existing tables inherits the current compression status of the table at the time the data is inserted.</p>
AUTOTEMP	<p>BLC is automatically performed by Vantage based on the frequency of access to the table data (the data temperature). Temperature-based BLC (TBBLC) compresses infrequently accessed data, and uncompresses data that it is accessed frequently.</p> <p><b>Note:</b></p> <p>If temperature-based BLC (TBBLC) is disabled, tables that are set to use AUTOTEMP will behave like tables set to use MANUAL compression until TBBLC is enabled again. TBBLC is enabled and disabled using the <a href="#">EnableTempBLC</a> field.</p>
ALWAYS	Data in the table is always compressed at the data block level.
NEVER	Data in the table is never compressed at the data block level.

### Default

MANUAL

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

Data temperatures refers to frequency of access:

- Hot data is frequently accessed.
- Cold data is infrequently accessed.
- Warm data is accessed moderately frequently.

Data temperature is measured separately for each AMP at a cylinder level and is a relative measure because the temperature of cylinders can change even if the data it is not directly accessed. This is because it depends on how often other cylinders are accessed in comparison. TBBLC automatically compresses cold data to save storage space and decompresses Warm or Hot data if it was previously Cold (and thus compressed). At any time, different portions of a table, which are spread among many cylinders, may exist in different states of compression depending on their temperatures.

BLC enables data compression at the data block (DB) level of the file system. Compression reduces the amount of storage required for a given amount of data. The BlockLevelCompression field of DBS Control enables and disables BLC.

When changed, causes TBBLC to immediately start a new cycle.

### Related Information

For more information on...	See...
Block-level compression	<a href="#">BlockLevelCompression</a> and <i>Teradata Vantage™ - Database Design</i> , B035-1094.
BLOCKCOMPRESSION table attribute	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.
EnableTempBLC field	<a href="#">EnableTempBLC</a> .
Ferret COMPRESS, UNCOMPRESS, and FORCE commands	<a href="#">Ferret Utility (ferret)</a> .
Query bands for compression	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.
Storage group of DBS Control fields	<a href="#">Storage Group</a> .

## DefaultTempSkewLimitPercent

Specifies the percentage of temporary space that an AMP is allowed to use over and above the per-AMP quota (the global maximum space limit divided by the number of AMPs). This is used for databases and users that are created without a SKEW option, or that are created with SKEW limit specified as DEFAULT for temporary space.

### Note:

This field should be changed only under the direction of Teradata Support Center personnel.

### Field Group

General

### Valid Range

0 through 10000 percent

### Default

0

### Changes Take Effect

After the DBS Control record has been written.

### Usage Notes

- For operations involving uniform load distribution across all the AMPs in the system, the default skew limit percentage may be set to lower values.
- If the system involves long running transactions with high space consumption, you can increase the value of the GlobalSpaceSoftLimitPercent DBS Control field to reduce the chances of transaction aborts due to a lack of available space.

### Related Information

For more information on...	See...
the SKEW option for DDL, such as CREATE DATABASE and CREATE USER	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.
related DBS Control fields	<ul style="list-style-type: none"> <li><a href="#">DefaultPermSkewLimitPercent</a></li> <li><a href="#">DefaultSpoolSkewLimitPercent</a></li> <li><a href="#">GlobalSpaceSoftLimitPercent</a></li> <li><a href="#">LegacySpaceAcctg</a></li> </ul>

# DefaultTimeThreshold

Specifies the amount of time in days that must elapse before statistics will be recollected on a table. If fewer than the set number of days has passed since the previous statistics collection, no new statistics will be collected.

This field is used when a COLLECT STATISTICS SQL statement does not specify the USING THRESHOLD DAYS option.

## Field Group

Optimizer Statistics

## Valid Range

0 through 9999

## Default

0, indicating time threshold checks are disabled.

## Changes Take Effect

After the DBS Control Record has been written.

## Usage Notes

Statistics are used by the optimizer to improve query performance.

## Related Information

For more information on...	See...
Optimizer statistics fields	<ul style="list-style-type: none"><li>• <a href="#">DefaultUserChangeThreshold</a></li><li>• <a href="#">SysChangeThresholdOption</a></li><li>• <a href="#">SysSampleOption</a></li></ul>

# DefaultUserChangeThreshold

Specifies the percentage of rows that must be changed or added to a table before statistics will be recollected on that table. If less than the set percentage of the table rows has changed since the previous statistics collection, no new statistics will be collected.

This field is used when a COLLECT STATISTICS SQL statement does not specify the USING THRESHOLD PERCENT option.

## Field Group

Optimizer Statistics

## Valid Range

0 through 9999

## Default

0

## Changes Take Effect

After the DBS Control Record has been written.

## Usage Notes

Statistics are used by the optimizer to improve query performance.

## Related Information

For more information on...	See...
Optimizer statistics fields	<ul style="list-style-type: none"><li>• <a href="#">DefaultTimeThreshold</a></li><li>• <a href="#">SysChangeThresholdOption</a></li><li>• <a href="#">SysSampleOption</a></li></ul>

## DefragLowCylProd

Determines the number of free cylinders below which cylinder defragmentation can begin. The system dynamically keeps cylinders defragmented.

If the system has less than the specified number of free cylinders, defragmentation occurs on cylinders with at least 25% free space, but not enough contiguous sectors to allocate a data block.

### Field Group

File System

### Valid Range

0 through 100 free cylinders

### Default

100 free cylinders

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

A cylinder becomes fragmented as a result of modifications to table data. As data blocks (DBs) are created, deleted, or change size as a result of these modifications, what started out as a small number of large free spaces on the newly allocated cylinder becomes a large number of small free spaces. Although there may still be a large total number of free sectors on the cylinder, with time there are fewer ranges of contiguous free sectors that are large enough to store new DBs. When new DBs cannot be stored on the cylinder, new free cylinders must be found and allocated.

The defragmentation process moves the data blocks closer together, coalescing the free space into areas that are large enough that new DBs can be allocated on the cylinder. This can reduce the need to allocate free cylinders for new data storage. The defragmentation process actually moves DBs from the fragmented cylinder to a new cylinder, where the DBs can be packed together, leaving a single, large set of consecutive free sectors. The cylinder from which the DBs were moved is freed, and made available for new allocations.

To keep costs to a minimum, background defragmentation is enabled only when the free cylinder count falls below the current value in the DefragLowCylProd field.

Setting this field to a value of 0 disables defragmentation. In this case, the lack of enough adjacent sectors to allocate a data block causes a new cylinder to be allocated.

The disadvantage of this approach is that free cylinders are a critical resource, and when no free cylinders remain, MiniCylPacks begin.

The operation is performed in the background.

**Performance Implications**

A low value in this field reduces the performance impact of defragmentation. However, setting the value extremely low might cause cylinders to be consumed more quickly, which could cause more MiniCylPacks to run. Set DefragLowCylProd higher than MiniCylPackLowCylProd because defragmentation has a smaller performance impact than cylinder pack.



# DictionaryCacheSize

Defines the size in kilobytes of the dictionary cache for each PE on the system.

## Field Group

Performance

## Valid Range

0, and 64 KB through 65536 KB

Setting this field to 0 means to use the Teradata default value.

## Default

8192 KB

## Changes Take Effect

After the next database restart.

## Usage Notes

The value of DictionaryCacheSize should be a multiple of the value of [DictionaryCacheSegmentSize](#).

## DictionaryCacheSize Performance Implications

The default value allows more caching of table header and database object access rights information, and reduces the number of I/Os required. It is especially effective for workloads that access many tables (more than 200) and for those that generate many dictionary seeks.

Increase the size of the dictionary cache to allow the parser to cache additional data dictionary and table header information.

For tactical and Online Complex Processing (OLCP) type workloads, maintaining a consistently short, few-second response time is important. These workloads may benefit from a larger dictionary cache, particularly when their query plans have not been cached in the request cache. A larger dictionary cache will allow more dictionary detail, needed for parsing and optimizing, to remain in memory for a longer period of time. For query workloads with a response time of more than one minute, there may be no measurable difference when this field is set to a higher value.

## Related Information

For more information on...	See...
DictionaryCacheSegmentSize	<a href="#">DictionaryCacheSegmentSize</a> .

# DictionaryCacheSegmentSize

Sets the memory segment size used for the Data Dictionary cache in multiples of 64 KB.

## Field Group

Performance

## Valid Range

0 and 1 through 256 (representing 64 KB through 16 MB)

Setting this field to 0 means to use the Teradata default value.

## Default

64 (representing 4 MB)

## Changes Take Effect

After the next database restart.

## Related Information

For more information on...	See...
DictionaryCacheSize	<a href="#">DictionaryCacheSize.</a>

## DictionaryDBSize

Specifies the maximum size for multirow data blocks in dictionary (database DBC) tables. Rows that are larger than DictionaryDBSize are stored in single-row data blocks, which are not limited by DictionaryDBSize.

### Field Group

File System

### Valid Values

42 through 2047 sectors for systems that use large cylinders.

18 through 512 sectors for systems that use small cylinders.

You can set DictionaryDBSize to USEPERMDBSIZE for both small and large cylinder systems. This value causes Vantage to use the current PermDBSize field setting for DictionaryDBSize.

A sector is 512 bytes.

### Default

254 sectors

Setting this field to 0 causes Vantage to use the system default size for this type of data block.

### Changes Take Effect

After a system initialization (sysinit) or an upgrade to a major xx.00 release.

### Usage Notes

When database tables are initially populated, Vantage stores as many rows as possible into each data block, until the block reaches the size specified by the various DB size settings in DBS Control. As tables are subsequently modified, rows can grow such that the existing data blocks would exceed the maximum size. When this happens, the data block is split, and roughly half the rows are moved to a new data block, with the result that the original and new data blocks are each one half of the original size. The result of this type of growth and splitting is that data blocks for heavily modified tables tend to be about 75% of the maximum size defined in DBS Control.

If DBs are compressed, this setting applies to the size of the uncompressed DBs.

Systems initialized on Teradata Database 13.10 or later release use large cylinders. On large-cylinder systems, the minimum data block size is 42 sectors (21504 bytes).

## DisableAutoCylPack

Disables AutoCylPack automatic background cylinder packing.

### Field Group

File System

### Valid Settings

Setting	Description
TRUE	Disables AutoCylPack
FALSE	Enables AutoCylPack

### Default

FALSE

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

Free Space Percent (FSP) is the amount of space on a cylinder that is unoccupied and available for storage, expressed as a proportion of the total space on the cylinder. For tables that are expected to grow, free space on cylinders allows for table growth without requiring the additional cost of allocating free cylinders.

Vantage can maintain different levels of free space on cylinders used to store different tables. Static tables can be designated with little or no FSP, while tables that change and grow can be designated to have FSP levels that allow for table growth.

AutoCylPack runs periodically as a background task to maintain the set levels of FSP on table cylinders. If there is less than the desired FSP available on the cylinder, AutoCylPack moves some data to other cylinders to free up some space. If there is more than the desired FSP available on the cylinder, AutoCylPack moves data to the cylinder, potentially freeing up other cylinders.

A desired target level for FSP can be specified for tables in several ways. The FSP that AutoCylPack maintains for cylinders of each table depends on how the FSP was specified:

- If the FSP was specified in a CREATE or ALTER TABLE statement, AutoCylPack maintains that FSP for the cylinders of the table.

**Note:**

If the FSP was specified in a CREATE or ALTER TABLE statement, and AutoCylPack is enabled, specifying a FREESPACEPERCENT value with a PACKDISK command is not recommended. In such cases, the FSP resulting from the PACKDISK operation would be temporary, and the table would be returned to the original table-level FSP setting by the background AutoCylPack task.

- If FSP was not specified with CREATE or ALTER TABLE, but PACKDISK was run on the table, AutoCylPack maintains the table cylinders at the FSP level that PACKDISK used.
- If an FSP was not specified in a CREATE or ALTER TABLE statement for the table, and PACKDISK was never run on the table, AutoCylPack uses the FSP specified by the AutoCylPackFSP field in DBS Control.

**Related Information**

For more information on...	See...
AutoCylPackFSP	<a href="#">AutoCylPackFSP</a> .
Free Space Percent	<a href="#">FreeSpacePercent</a> , <a href="#">SHOWFSP</a> , <i>Teradata Vantage™ - Database Administration</i> , B035-1093.
CREATE TABLE and ALTER TABLE	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.
PACKDISK	<a href="#">PACKDISK</a> .
MiniCylPack	<a href="#">MiniCylPackLowCylProd</a> , <i>Teradata Vantage™ - Database Administration</i> , B035-1093.

## DisableDATASETValidation

Determines whether validation is enabled or disabled for the native Teradata DATASET data type.

### Field Group

General

### Valid Settings

Setting	Description
TRUE	DATASET validation is disabled.
FALSE	DATASET validation is enabled.

### Default

FALSE

### Changes Take Effect

After the next database restart.

### Related Information

For More Information on...	See...
DATASET data type	<i>Teradata Vantage™ - DATASET Data Type</i> , B035-1198.
Related DBS Control fields	<a href="#">DATASETAttributeSize</a> and <a href="#">EnableDATASET</a>

## DisableJSONValidation

Determines whether validation is enabled or disabled for the native Teradata JSON data type.

### Field Group

General

### Valid Settings

Setting	Description
TRUE	JSON validation is disabled.
FALSE	JSON validation is enabled.

### Default

FALSE

### Changes Take Effect

After the next database restart.

### Related Information

For More Information on...	See...
JSON data type	<i>Teradata Vantage™ - JSON Data Type</i> , B035-1150.
Related DBS Control fields	<a href="#">EnableJSON</a> , <a href="#">JSON_AttributeSize</a> , and <a href="#">JSON_MaxDepth</a> .

## DisableMergeBlocks

Determines whether the file system automatically merges data blocks together on storage cylinders during full table modify operations. Merging several small data blocks into a single larger block can improve system performance by reducing disk I/O.

### Field Group

File System

### Valid Settings

Setting	Description
TRUE	Automatic merging of data blocks is disabled.
FALSE	Automatic merging of data blocks is enabled.

### Default

FALSE

### Changes Take Effect

After the DBS Control record has been written.

### Usage Notes

During normal database operations, the data blocks that store table rows on cylinders can split and shrink, resulting in many blocks which are far smaller than the maximum allowed data block size. Full table modify operations for tables with several small data blocks require more disk I/O than would be required if the table rows were stored on fewer and larger data blocks. Vantage can merge the small data blocks of these tables automatically during full table modify operations, which can result in reduced I/O overhead and improved database performance.

Data block merging applies only to permanent and permanent journal tables.

If DisableMergeBlocks is TRUE, the MergeBlockRatio field is ignored.

### Related Information

For more information on...	See...
Setting the maximum size of merged data blocks	<a href="#">MergeBlockRatio</a> .
the MERGEBLOCKRATIO option of CREATE TABLE and ALTER TABLE	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.



## DisablePeekUsing

Enables or disables the performance enhancements associated with exposed USING values in parameterized queries.

### Note:

This field should be changed only under the direction of Teradata Support Center personnel.

### Field Group

Performance

### Valid Settings

Setting	Description
TRUE	Optimizer peek USING performance enhancements are disabled.
FALSE	Optimizer peek USING performance enhancements are enabled.

### Default

FALSE

### Changes Take Effect

As soon as the request cache is purged.

### Performance Implications

The Teradata Query Optimizer determines the most efficient way to execute an SQL request in the Teradata parallel environment. It generates several possible plans of action, which involve alternate methods of accessing and joining database tables to satisfy the request. The Optimizer evaluates the relative costs of each plan in terms of resource usage and speed of execution, then chooses the plan with the lowest cost. Plans that are sufficiently generic are cached for fast reuse by the Optimizer in similar situations.

For some parameterized queries, the Optimizer can generate better plans by "peeking" at the specific USING values (data parcels) in the queries. Because the plans are specific for the USING values, they are not cached, which in rare cases may have an adverse affect on performance. The DisablePeekUsing field allows you to disable this feature of the Optimizer if you suspect it is a problem.

### Related Information

For more information on...	See...
the request cache and query optimization	<i>Teradata Vantage™ - SQL Request and Transaction Processing</i> , B035-1142.

For more information on...	See...
parameterized queries	<i>Teradata Vantage™ - SQL Request and Transaction Processing</i> , B035-1142.

## DisableRecoverableNetProtocol

Determines whether the Recoverable Network Protocol (RNP) is enabled.

RNP prevents transactions from being aborted and database response data from being dropped due to network failures. Under RNP, if there is a network disconnect during request processing, the gateway will preserve response data buffers and will not abort transactions that were in progress at the time of the disconnect.

RNP allows clients to reconnect when the network connection is re-established, and continue processing the interrupted requests. Because transactions were not aborted, they need not be resubmitted, which can save a significant amount of time.

---

### Note:

RNP is used only for sessions that are enabled with Redrive protection. For details on Redrive protection, see Database Administration.

---

This protocol is supported by CLI and JDBC.

### Field Group

General

### Valid Settings

Setting	Description
TRUE	Disables RNP
FALSE	Enables RNP

### Default

FALSE

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

Logons must be disabled in order to change this field. Users with appropriate privileges can disable logons from the Teradata system console. For more information, see [Database Window \(xdbw\)](#). In addition, Redrive protection must also be disabled.

## DisableRestartableBLC

Determines whether a block-level compression (BLC) operation that is interrupted by a database reset will resume automatically when the system restarts.

### Field Group

General

### Valid Settings

Setting	Description
TRUE	Disables restartable BLC. If a BLC compress or uncompress operation is interrupted by a system reset, the operation will not continue when the system restarts, and tables may be left in partially compressed states. Additionally, BLC compress and uncompress operations are not allowed if there are any down AMPs in the current Ferret scope.
FALSE	Enables restartable BLC. If a BLC compress or uncompress operation is interrupted by a system reset, the operation continues automatically when the system restarts. Additionally, BLC compress and uncompress operations are allowed, even if there are down AMPs in the current Ferret scope.

### Default

FALSE

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

If the value of this field is TRUE, and tables become partially compressed, you can use the COMPRESS and UNCOMPRESS commands to make the table state consistent. For more information see [Ferret Utility \(ferret\)](#).

Enabling restartable BLC requires that the NoDot0Backdown field be TRUE, which prevents the system from being backed down to the previous release level.

## DisableSyncScan

Disables or enables caching portions of large tables for the purpose of synchronized full table scans. The amount of cache that can be used for synchronized scans is set using the [SyncScanCacheThr](#) field.

### Field Group

Performance

### Valid Settings

Setting	Description
TRUE	Disables synchronized full table scanning of large tables.
FALSE	Enables synchronized full table scans.

### Default

FALSE

### Changes Take Effect

After the next database restart.

### Usage Notes

The synchronized full-table scan feature of Teradata allows several table scanning tasks to simultaneously access the portion of a large table that is currently in the cache. Synchronized table scans happen only when full-table scans access similar areas of a large table. Synchronized table scans can improve database performance by reducing the required disk I/O. Synchronized table scans are available only to large tables undergoing full-table scans.

### Related Information

See [SyncScanCacheThr](#).

## DisableUDTImplCastForSysFuncOp

Disables/enables implicit cast/conversion of UDT expressions passed to built-in system operators/functions.

### Field Group

General

### Valid Settings

Setting	Description
TRUE	Disable implicit conversions.
FALSE	Enable implicit conversions.

### Default

FALSE

### Usage Notes

The system implicitly converts a UDT expression to a compatible pre-defined type when there is a CAST AS ASSIGNMENT definition whose target is a compatible data type.

---

#### Note:

This field only affects built-in system operators and functions. The system always invokes implicit casts for INSERT, UPDATE, and parameter passing (UDF, UDM, Stored Procedure) operations.

---

## DisableWAL

Determines whether Write-Ahead Logging (WAL) is enabled.

### Field Group

File System

### Valid Settings

Setting	Description
TRUE	The WAL feature is disabled. Changed data blocks and cylinder indexes are written directly to disk. The DisableWALforDBs field is ignored.
FALSE	<p>The WAL feature is enabled. The items written to the WAL log are determined by the setting of DisableWALforDBs:</p> <ul style="list-style-type: none"> <li>• If DisableWALforDBs is TRUE, changed data blocks are written to disk, and cylinder index changes are written to the WAL log.</li> <li>• If DisableWALforDBs is FALSE, changed data blocks are written to the WAL log and cached in memory, or may be written to disk, depending on the specific operation.</li> </ul>

### Default

FALSE

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

WAL is a log-based file system recovery scheme in which modifications to permanent data are written to a log file, the WAL log. The log file contains change records (Redo records) which represent the updates. At key moments, such as transaction commit, the WAL log is forced to disk. In the case of a reset or crash, Redo records can be used to transform the old copy of a permanent data block on disk into the version that existed at the time of the reset.

Data blocks inserted by the MultiLoad utility are always written to disk regardless of the DisableWAL field setting.

### Related Information

See [DisableWALforDBs](#).

## DisableWALforDBs

Determines whether changed data blocks are written directly to disk or to the Write-Ahead Logging (WAL) log.

### Note:

This field is effective only when WAL is enabled (DisableWAL is set to FALSE).

### Field Group

File System

### Valid Settings

Setting	Description
TRUE	Changed data blocks are written to disk.
FALSE	Changed data blocks are written to the WAL log and cached in memory, or may be written to disk, depending on the specific operation.

### Default

FALSE

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

WAL is a log-based file system recovery scheme in which modifications to permanent data are written to a log file, the WAL log. The log file contains change records (Redo records) which represent the updates. At key moments, such as transaction commit, the WAL log is forced to disk. In the case of a reset or crash, Redo records can be used to transform the old copy of a permanent data block on disk into the version that existed at the time of the reset.

### Related Information

See [DisableWAL](#).



## **DMLStatementShipping [Deprecated]**

Do not use.

## DotNotationOnErrorCondition

Determines how Vantage responds when queries using dot notation return a list of values.

### Field Group

General

### Valid Settings

Setting	Description
0	The system returns a warning and an error string when a query using pre-16.00 dot notation returns a list of values. For newer elements added in Teradata Database 16.00, the system returns the list of values.
1	The system returns the list of values in all cases when a dot-notation query returns a list.
2	The system returns NULL in all cases when a dot-notation query returns a list.
3	The system returns an error in all cases when a dot-notation query returns a list.

### Default

0

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

Dot notation versions prior to Teradata Database 16.00 do not natively support returning lists of data.

### Related Information

For More Information on...	See...
dot notation data types	<ul style="list-style-type: none"> <li>• <i>Teradata Vantage™ - DATASET Data Type</i>, B035-1198.</li> <li>• <i>Teradata Vantage™ - JSON Data Type</i>, B035-1150.</li> </ul>

## EnableCostProfileTLE

Enables/disables Optimizer Cost Estimation Subsystem (OCES) diagnostics for use in combination with Target Level Emulation (TLE).

### Note:

This field should be changed only under the direction of Teradata Support Center personnel.

### Field Group

General

### Valid Settings

Setting	Description
TRUE	Enables OCES diagnostics to be used in combination with TLE. When set to TRUE, all TLE diagnostics are also processed by the OCES logic.
FALSE	Disables combined OCES/TLE diagnostics.

### Note:

If the value of the EnableSetCostProfile field is equal to 0, the EnableCostProfileTLE field is set to FALSE.

### Default

FALSE

### Changes Take Effect

For new sessions after the DBS Control Record has been written.

### Related Information

For More Information on...	See...
the EnableSetCostProfile field	<a href="#">EnableSetCostProfile.</a>

## EnableDATASET

Determines whether native Teradata DATASET data type support is enabled in Vantage.

### Field Group

General

### Valid Settings

Setting	Description
TRUE	DATASET support is enabled. Native Teradata DATASET data type has been or can be created and is available for use.
FALSE	DATASET support is disabled.

### Default

TRUE if no user-defined DATASET data type existed on the system at the time it was upgraded to Teradata Database 16.00.

FALSE if a user-defined DATASET data type existed on the system at the time it was upgraded to Teradata Database 16.00.

### Changes Take Effect

After the next database restart.

### Usage Notes

The DIPDATASET DIP script creates the Teradata DATASET data type. If a system is being upgraded from a pre-16.00 release, and a user-defined data type named DATASET already exists on the system, the Teradata DATASET data type is not created, and the EnableDATASET field is set to FALSE.

If you want to use the Teradata DATASET data type on a system where EnableDATASET is set to FALSE, delete your DATASET data type, modify the value of EnableDATASET to TRUE, and run the DIPDATASET DIP script.

### Related Information

For More Information on...	See...
DATASET data type	<i>Teradata Vantage™ - DATASET Data Type</i> , B035-1198.
Related DBS Control fields	<a href="#">DATASETAttributeSize</a> , <a href="#">DisableDATASETValidation</a> and <a href="#">EnableDATASETCSV</a> .
DIP utility	<a href="#">Database Initialization Program (dip)</a>

## EnableDATASETCSV

Determines whether the DATASET CSV storage format is available for use by Vantage.

### Field Group

General

### Valid Settings

Setting	Description
TRUE	DATASET CSV is available for use.
FALSE	DATASET CSV is not available for use.

### Default

TRUE

### Changes Take Effect

After the next database restart.

### Usage Notes

The DATASET CSV storage format is available only when the EnableDATASET DBS Control field is set to TRUE to enable the DATASET data type.

### Related Information

For More Information on...	See...
DATASET data type	<i>Teradata Vantage™ - DATASET Data Type</i> , B035-1198.
Related DBS Control fields	<a href="#">EnableDATASET</a> , <a href="#">DATASETAttributeSize</a> , and <a href="#">DisableDATASETValidation</a> .

## EnableDynamicParameterization

Determines whether Vantage uses dynamic parameterization of literals (DPL) when processing repeated nonparameterized requests.

### Field Group

General

### Valid Settings

Setting	Description
TRUE	Vantage attempts to produce and reuse generic query execution plans for similar, nonparameterized queries that differ only in literal values in WHERE and ON clauses.
FALSE	Vantage does not reuse generic query execution plans for nonparameterized queries that differ only in literal values in WHERE and ON clauses.

### Default

FALSE

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

- When dynamic parameterization is enabled, and eligible nonparameterized requests differ only in the eligible literal values used in the predicates of WHERE and ON clauses, DPL logic in the Parser automatically parameterizes these literal. This allows the Parser to reuse a cached generic execution plan for such queries, and speed up processing.
- Teradata recommends enabling this feature if you have several closely similar nonparameterized requests that would benefit from this type of processing. DPL can improve database query performance in proportion to the number of similar database requests.

### Related Information

For More Information on...	See...
Dynamic parameterization of literal values during transaction processing and optimization	<i>Teradata Vantage™ - SQL Request and Transaction Processing</i> , B035-1142.

# EnableEON

This field is set to TRUE and cannot be changed. It indicates that Vantage supports and enforces database object names of up to 128 UNICODE characters.

Additional naming restrictions can be imposed using the [NameValidationRule](#) field.

## Field Group

General

## Related Information

For More Information on...	See...
Object Naming	<i>Teradata Vantage™ - SQL Fundamentals</i> , B035-1141.
NameValidationRule field settings	<a href="#">NameValidationRule</a> .

## EnableJSON

Determines whether native JSON support is enabled in Vantage.

### Field Group

General

### Valid Settings

Setting	Description
TRUE	JSON support is enabled. Native Teradata JSON data type is available.
FALSE	JSON support is disabled.

### Default

TRUE

### Changes Take Effect

After the next database restart.

### Related Information

For More Information on...	See...
JSON data type	<i>Teradata Vantage™ - JSON Data Type</i> , B035-1150.
Related DBS Control fields	<a href="#">DisableJSONValidation</a> , <a href="#">JSON_AttributeSize</a> , and <a href="#">JSON_MaxDepth</a> .



## EnableLogonsMsg

Enables or disables the display of “Logons are enabled” message in /var/log/messages on all nodes of a multinode (MPP) system.

### Field Group

General

### Valid Settings

Setting	Description
TRUE	The “Logons are enabled” message is broadcast to all system nodes and logged to /var/log/messages.
FALSE	The “Logons are enabled” message is logged only to the node containing the control AMP.

### Default

FALSE

### Changes Take Effect

After the DBS Control Record has been written.

### Related Information

For more information on enabling and disabling logons, see Database Window in *Teradata Vantage™ - Database Utilities*, B035-1102.

## EnabNonTempoOp

Determines whether NONTEMPORAL operations can be used.

NONTEMPORAL operations delete rows, modify history rows, or change Transaction Time column values on temporal tables that include a Transaction Time column. These operations include NONTEMPORAL DELETE, NONTEMPORAL UPDATE, NONTEMPORAL INSERT, NONTEMPORAL CREATE TABLE AS, and NONTEMPORAL ALTER TABLE.

The NONTEMPORAL privilege allows a user to circumvent the automatic history that is kept for tables with transaction time. It allows users to perform operations that would otherwise not be allowed on these tables. For purposes of data security and integrity, Teradata recommends that the EnabNonTempoOp setting remain unchanged from the default, so that NONTEMPORAL operations are blocked, unless absolutely necessary.

### Field Group

General

### Valid Settings

Setting	Description
TRUE	Nontemporal operations can be performed on temporal tables.
FALSE	Nontemporal operations cannot be performed on temporal tables.

### Default

FALSE

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

Leave this field set to FALSE to prevent inadvertent and unauthorized modifications to temporal tables that include transaction time. Enable this setting only when nontemporal operations are necessary, as for deleting very old history rows or fixing problems with temporal tables.

### Related Information

For more information on temporal operations, see *Teradata Vantage™ - Temporal Table Support*, B035-1182.

## EnableNOS

Determines whether the Native Object Store (NOS) feature is enabled or disabled.

## Field Group

Native Object Store

## Valid Settings

Setting	Description
TRUE	The NOS feature is enabled. Foreign tables can be created in Vantage.
FALSE	<p>The NOS feature is disabled. Foreign tables cannot be created in Vantage.</p> <p><b>Note:</b></p> <p>Foreign tables that exist in the database when this field is set to FALSE continue to exist. They can still be queried, and can be dropped.</p>

## Default

TRUE

## Changes Take Effect

After the next database restart.

## Usage Notes

*Foreign tables* allow Vantage to access data from external, cloud-based data storage, such as AWS S3, without requiring you to manually move the data into the database first from where it natively resides. Foreign tables are identified by a hostname, path, and other metadata that point to the external storage. Vantage can read and process semi-structured or unstructured external data in foreign tables using standard SQL. For example, you can use Teradata analytic functions to examine the data, join it to the relational data in the database, and issue queries against it as you can for other data in Vantage.

## Related Information

The following fields affect the NOS feature and foreign table access.

- [ConnectionBufferSize](#)
- [ConnectionTimeout](#)
- [DefaultRowGroupSize](#)
- [RequestDuration](#)
- [RequestTimeout](#)
- [UserRetryLimit](#)

## EnableSetCostProfile

Controls whether the system cost profile can be changed for purposes of system emulation and special-case system tuning. A *cost profile* is a suite of system-specific values (*cost profile constants*) that Vantage uses for Optimizer costing coefficients, enabling and disabling certain features, and other system settings. A cost profile can apply system-wide or be associated with a PROFILE and apply only to users assigned that PROFILE.

---

**Note:**

This field should be changed only under the direction of Teradata Support Center personnel.

---

### Field Group

General

### Valid Settings

Setting	Description
0	Disable changing system cost profile.
1	Enable changing system cost profile for SESSION and REQUEST levels.
2	Enable changing system cost profile at all levels.

### Default

0

### Changes Take Effect

For new sessions after the DBS Control Record has been written.

## EnableTempBLC

Determines whether temperature-based block-level compression (TBBLC) is enabled.

### Field Group

Compression

### Valid Settings

Setting	Description
TRUE	<p>TBBLC is enabled. Tables using AUTOTEMP compression (specified in CREATE TABLE or ALTER TABLE statements) will have infrequently accessed data blocks automatically compressed for storage.</p> <p><b>Note:</b> In order to use TBBLC, BlockLevelCompression must be set to On.</p>
FALSE	TBBLC is disabled. Data blocks are not compressed based on frequency of access.

### Default

FALSE

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

When TBBLC is enabled, [AutoCylPackColdData](#) should be set to TRUE, and [DisableAutoCylPack](#) should be set to FALSE.

If TBBLC is disabled, tables that are set to use automatic temperature-based compression will behave like tables set to use manual compression until TBBLC is enabled again. The system default for whether tables use manual or automatic compression is set with the [DefaultTableMode](#) setting.

BLC enables data compression at the data block (DB) level of the Teradata file system. Compression reduces the amount of storage required for a given amount of data. The BlockLevelCompression field of DBS Control enables and disables BLC.

Data temperature refers to the frequency of access. Hot data is frequently accessed. Cold data is infrequently accessed. Warm data is accessed moderately frequently. TBBLC automatically compresses colder data to save storage space, and decompresses the data if it becomes hotter. At any time, different portions of a table, such as different partitions, may exist in different states of compression depending on their temperatures.

**Related Information**

For more information on...	See...
Block-level compression	<i>Teradata Vantage™ - Database Design</i> , B035-1094.
BLOCKCOMPRESSION table attribute	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.
DefaultTableMode field	<a href="#">DefaultTableMode</a> .
Ferret COMPRESS, UNCOMPRESS, and FORCE commands	<a href="#">Ferret Utility (ferret)</a> .
Query bands for compression	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.
Storage group of DBS Control fields	<a href="#">Storage Group</a> .

## ExportOrderBySegmentLimit

Specifies the maximum number of segments that can be allocated for FastExport redistribution when data is to be sorted.

### Field Group

General

### Valid Range

500 through 2000

### Default

500

### Changes Take Effect

After the DBS Control record has been written.

### Usage Notes

Increase the value of this setting if a large, complex FastExport job generates error 2587: Segment overflow - FastExport request is too complex to process.

## Export Width Table ID

Allows you to set the system default for the export width of a character in bytes.

### Field Group

General

### Valid Settings

Setting	Description
0 - Expected defaults	Provides reasonable default widths for the character data type and client form of use.
1 - Compatibility defaults	Allows Unicode columns to work in a way compatible with applications that were written to use Latin or Kanji1 columns.
2 - Maximum defaults	Provides maximum default width of the character data type and client form of use.

### Default

0 (Expected defaults)

### Changes Take Effect

After the next database restart.

### Usage Notes

The system-level export width can be overridden at the user level by use of EXPORTWIDTH option to the CREATE USER and MODIFY USER SQL statements. For more information, see *Teradata Vantage™ - SQL Data Definition Language Syntax and Examples*, B035-1144.

### Expected Default Export Widths

The following table illustrates the number of bytes exported from the various server character sets to the various client character sets for the Expected Default export width table (Export Width Table ID = 0).

IF the client character set is...	AND the server character set is...	THEN the number of bytes exported for a CHARACTER(n) column is...
any single-byte character set	<ul style="list-style-type: none"> <li>LATIN</li> <li>UNICODE</li> <li>KANJISJIS</li> <li>KANJ11</li> </ul>	n



IF the client character set is...	AND the server character set is...	THEN the number of bytes exported for a CHARACTER(n) column is...
	<b>Note:</b> GRAPHIC is not supported.	
KANJIEUC_0U KANJISJIS_0S KANJI932_1S0	<ul style="list-style-type: none"> <li>LATIN</li> <li>KANJISJIS</li> <li>KANJI1</li> </ul>	n
	<ul style="list-style-type: none"> <li>UNICODE</li> <li>GRAPHIC</li> </ul>	2n
	<ul style="list-style-type: none"> <li>UNICODE</li> <li>GRAPHIC</li> </ul>	n
KANJIEBCDIC5026_0I KANJIEBCDIC5035_0I KATAKANAEBDIC	<ul style="list-style-type: none"> <li>LATIN</li> <li>KANJISJIS</li> <li>KANJI1</li> </ul>	n
	UNICODE	2n+2
	GRAPHIC	<ul style="list-style-type: none"> <li>2n (record and indicator modes)</li> <li>2n+2 (field mode)</li> </ul>
UTF-8	LATIN	2n
	UNICODE	3n
	<ul style="list-style-type: none"> <li>KANJISJIS</li> <li>KANJI1</li> </ul>	n
	<b>Note:</b> GRAPHIC is not supported.	
UTF-16	<ul style="list-style-type: none"> <li>LATIN</li> <li>UNICODE</li> <li>KANJISJIS</li> <li>KANJI1</li> <li>GRAPHIC</li> </ul>	2n
<ul style="list-style-type: none"> <li>HANGUL949_7R0</li> <li>HANGULKSC5601_2R4</li> <li>SCHGB2312_1T0</li> <li>SCHINESE936_6R0</li> <li>SDHANGULKSC5601_4R4</li> </ul>	<ul style="list-style-type: none"> <li>LATIN</li> <li>KANJISJIS</li> <li>KANJI1</li> </ul>	n
	<ul style="list-style-type: none"> <li>UNICODE</li> </ul>	2n

IF the client character set is...	AND the server character set is...	THEN the number of bytes exported for a CHARACTER(n) column is...
<ul style="list-style-type: none"> <li>• SDTCHBIG5_3R0</li> <li>• SDSCHGB2312_2T0</li> <li>• TCHBIG5_1R0</li> <li>• TCHINESE950_8R0</li> <li>• Extended site-defined multibyte client character sets that use one of the following encoding forms:               <ul style="list-style-type: none"> <li>◦ Single-byte characters in the range 0x00-0x81</li> <li>◦ double-byte characters with first byte in the range 0x82-0xFF</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• GRAPHIC</li> </ul>	
<ul style="list-style-type: none"> <li>• HANGULEBCDIC933_1II</li> <li>• SCHEBCDIC935_21J</li> <li>• SDHANGULEBCDIC933_5II</li> <li>• SDSCHEBCDIC935_6IJ</li> <li>• SDTCHEBCDIC937_7IB</li> <li>• TCHEBCDIC937_3IB</li> </ul>	<ul style="list-style-type: none"> <li>• LATIN</li> <li>• KANJISJIS</li> <li>• KANJI1</li> </ul>	n
	UNICODE	2n+2
	GRAPHIC	<ul style="list-style-type: none"> <li>• 2n (record and indicator modes)</li> <li>• 2n+2 (field mode)</li> </ul>

### Compatibility Default Export Widths

The following table illustrates the number of bytes exported from the various server character sets to the various client character sets for the Compatibility Default export width table (Export Width Table ID = 1).

IF the client character set is...	AND the server character set is...	THEN the number of bytes exported for a CHARACTER(n) column is...
any single-byte character set	<ul style="list-style-type: none"> <li>• LATIN</li> <li>• UNICODE</li> <li>• KANJISJIS</li> <li>• KANJI1</li> </ul>	n
	<b>Note:</b> GRAPHIC is not supported.	
<ul style="list-style-type: none"> <li>• KANJIEUC_0U</li> <li>• KANJISJIS_0S</li> <li>• KANJI932_1S0</li> </ul>	<ul style="list-style-type: none"> <li>• LATIN</li> <li>• UNICODE</li> <li>• KANJISJIS</li> <li>• KANJI1</li> </ul>	n
	GRAPHIC	2n

IF the client character set is...	AND the server character set is...	THEN the number of bytes exported for a CHARACTER(n) column is...
KANJI EBCDIC5026_OI KANJI EBCDIC5035_OI KATAKANA EBCDIC	<ul style="list-style-type: none"> <li>LATIN</li> <li>UNICODE</li> <li>KANJISJIS</li> <li>KANJI1</li> </ul>	n
	GRAPHIC	<ul style="list-style-type: none"> <li>2n (record and indicator modes)</li> <li>2n+2 (field mode)</li> </ul>
UTF-8	LATIN	2n
	<ul style="list-style-type: none"> <li>UNICODE</li> </ul>	3n
	<ul style="list-style-type: none"> <li>KANJISJIS</li> <li>KANJI1</li> </ul>	n
	<b>Note:</b> GRAPHIC is not supported.	
UTF-16	<ul style="list-style-type: none"> <li>LATIN</li> <li>UNICODE</li> <li>KANJISJIS</li> <li>KANJI1</li> <li>GRAPHIC</li> </ul>	2n
<ul style="list-style-type: none"> <li>HANGUL949_7R0</li> <li>HANGULKSC5601_2R4</li> <li>SCHGB2312_1T0</li> <li>SCHINESE936_6R0</li> <li>SDHANGULKSC5601_4R4</li> <li>SDTCHBIG5_3R0</li> <li>SDSCHGB2312_2T0</li> <li>TCHBIG5_1R0</li> <li>TCHINESE950_8R0</li> <li>Extended site-defined multibyte client character sets that use one of the following encoding forms:               <ul style="list-style-type: none"> <li>Single-byte characters in the range 0x00-0x81</li> <li>double-byte characters with first byte in the range 0x82-0xFF</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>LATIN</li> <li>UNICODE</li> <li>KANJISJIS</li> <li>KANJI1</li> </ul>	n
	<ul style="list-style-type: none"> <li>GRAPHIC</li> </ul>	2n

IF the client character set is...	AND the server character set is...	THEN the number of bytes exported for a CHARACTER(n) column is...
<ul style="list-style-type: none"> <li>HANGULEBCDIC933_1II</li> <li>SCHEBCDIC935_21J</li> <li>SDHANGULEBCDIC933_5II</li> <li>SDSCHEBCDIC935_6IJ</li> <li>SDTCHEBCDIC937_7IB</li> <li>TCHEBCDIC937_3IB</li> </ul>	<ul style="list-style-type: none"> <li>LATIN</li> <li>UNICODE</li> <li>KANJISJIS</li> <li>KANJI1</li> </ul>	n
	GRAPHIC	<ul style="list-style-type: none"> <li>2n (record and indicator modes)</li> <li>2n+2 (field mode)</li> </ul>

### Maximum Default Export Widths

The following table illustrates the number of bytes exported from the various server character sets to the various client character sets for the Maximum Default export width table (Export Width Table ID = 2).

IF the client character set is...	AND the server character set is...	THEN the number of bytes exported for a CHARACTER(n) column is...
any single-byte character set	<ul style="list-style-type: none"> <li>LATIN</li> <li>UNICODE</li> <li>KANJISJIS</li> <li>KANJI1</li> </ul>	n
	<b>Note:</b> GRAPHIC is not supported.	
KANJI EUC_0U	<ul style="list-style-type: none"> <li>LATIN</li> <li>UNICODE</li> <li>GRAPHIC</li> </ul>	3n
	<ul style="list-style-type: none"> <li>KANJISJIS</li> <li>KANJI1</li> </ul>	n
KANJI EBCDIC5026_0I KANJI EBCDIC5035_0I KATAKANA EBCDIC	<ul style="list-style-type: none"> <li>LATIN</li> <li>KANJI1</li> </ul>	n
	UNICODE	3n+1
	KANJISJIS	2n
	GRAPHIC	<ul style="list-style-type: none"> <li>2n (record and indicator modes)</li> <li>2n+2 (field mode)</li> </ul>

IF the client character set is...	AND the server character set is...	THEN the number of bytes exported for a CHARACTER(n) column is...
KANJI932_1S0 KANJI932_1S0	<ul style="list-style-type: none"> <li>LATIN</li> <li>UNICODE</li> <li>GRAPHIC</li> </ul>	2n
	<ul style="list-style-type: none"> <li>KANJISJIS</li> <li>KANJI1</li> </ul>	n
UTF-8	<ul style="list-style-type: none"> <li>LATIN</li> <li>UNICODE</li> <li>KANJISJIS</li> </ul>	3n
	KANJI1	n
	<b>Note:</b> GRAPHIC is not supported.	
UTF-16	<ul style="list-style-type: none"> <li>LATIN</li> <li>UNICODE</li> <li>KANJISJIS</li> <li>KANJI1</li> <li>GRAPHIC</li> </ul>	2n
<ul style="list-style-type: none"> <li>HANGUL949_7R0</li> <li>HANGULKSC5601_2R4</li> <li>SCHGB2312_1T0</li> <li>SCHINESE936_6R0</li> <li>SDHANGULKSC5601_4R4</li> <li>SDTCHBIG5_3R0</li> <li>SDSCHGB2312_2T0</li> <li>TCHBIG5_1R0</li> <li>TCHINESE950_8R0</li> <li>Extended site-defined multibyte client character sets that use one of the following encoding forms:               <ul style="list-style-type: none"> <li>Single-byte characters in the range 0x00-0x81</li> <li>double-byte characters with first byte in the range 0x82-0xFF</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>LATIN</li> <li>UNICODE</li> <li>GRAPHIC</li> </ul>	2n
	<ul style="list-style-type: none"> <li>KANJISJIS</li> <li>KANJI1</li> </ul>	n
<ul style="list-style-type: none"> <li>HANGULEBCDIC933_1II</li> <li>SCHEBCDIC935_21J</li> <li>SDHANGULEBCDIC933_5II</li> <li>SDSCHEBCDIC935_6IJ</li> <li>SDTCHEBCDIC937_7IB</li> <li>TCHEBCDIC937_3IB</li> </ul>	<ul style="list-style-type: none"> <li>KANJISJIS</li> <li>KANJI1</li> </ul>	n
	UNICODE	3n+1
	LATIN	2n

IF the client character set is...	AND the server character set is...	THEN the number of bytes exported for a CHARACTER(n) column is...
	GRAPHIC	<ul style="list-style-type: none"><li>• 2n (record and indicator modes)</li><li>• 2n+2 (field mode)</li></ul>

## ExternalAuthentication

Controls whether Vantage users can be authenticated outside (external) of the Vantage software authentication system.

### Field Group

General

### Valid Settings

Setting	Description
0 (ON)	External authentication sessions and traditional logons are accepted.
1 (OFF)	External authentication sessions are rejected; traditional logons are accepted.
2 (ONLY)	External authentication sessions are accepted; traditional logons are rejected.

### Default

0 (On)

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

The SET EXTAUTH database command duplicates the functionality of the ExternalAuthentication DBS Control field.

### Related Information

For more information on...	See...
the SET EXTAUTH command	<a href="#">Database Window (xdbw)</a> .
configuring how the network allows or disallows traditional and new external authentication logons	<a href="#">Gateway Control (gtwcontrol)</a> .

## FastAlterDefault

Allows automatic use of Fast Column Add (FCA) to add new columns to populated user tables without using USING FAST MODE ON in ALTER TABLE syntax. FastAlterDefault should be used carefully because once it is set to TRUE, every alter table that adds columns will switch to fast mode. Before setting FastAlterDefault to TRUE, FastAlterEnable must be set to TRUE.

**Field Group**

General

**Valid Settings**

Setting	Description
TRUE	Enables adding columns to populated user tables in fast mode.
FALSE	Default. Use slow mode to add columns.

**Default**

FALSE

**Changes Take Effect**

After the DBS Control Record has been written.

**Usage Notes**

- Use this DBS control flag to add new columns with non-null default values to tables with large number of rows, which would normally take a lot of time to complete.
- Does not add value for smaller tables that typically take little time to add new columns.

**Related Information**

- [FastAlterEnable](#)
- *Teradata Vantage™ - SQL Data Definition Language Syntax and Examples*

**FastAlterEnable**

Used to turn on Fast Mode ALTER TABLE.

**Field Group**

General

**Valid Settings**

Setting	Description
TRUE	Enables Fast Mode ALTER TABLE.
FALSE	Disables Fast Mode ALTER TABLE.

**Default**

FALSE



**Changes Take Effect**

After the DBS Control Record has been written.

**Usage Notes**

- Fast Column Add (FCA) cannot be used if this flag is disabled.

**Related Information**

- [FastAlterDefault](#)
- *Teradata Vantage™ - SQL Data Definition Language Syntax and Examples*

## FixSRandomRange

Allows RANDOM function compatibility between systems managed by Teradata® Unity™ that are running different database versions.

Prior database releases had an issue with the RANDOM function, `RANDOM(lower_bound,upper_bound)`, whereby, in rare instances, the function could return a value that was one greater than the specified `upper_bound`. Although this issue has been fixed, in Unity-paired systems, RANDOM function behavior must be consistent between systems. This field ensures that the RANDOM function behaves identically for systems managed by Teradata® Unity™, regardless of the database version each system is running.

### Field Group

General

### Valid Settings

Setting	Description
TRUE	Ensures consistent and correct results for RANDOM function operations when multiple systems managed by Teradata® Unity™ have the FixSRandomRange field available.
FALSE	Maintains consistency of RANDOM function results between multiple systems managed by Teradata® Unity™ when the FixSRandomRange field is not available on all systems.

### Default

TRUE on systems initialized (sysinited) on Teradata Database 15.0 or for which this issue with the RANDOM function has been fixed.

FALSE on systems upgraded from prior releases and not re-initialized, where this RANDOM issue still exists.

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

If all systems being managed by Teradata® Unity™ have FixSRandomRange available in DBS control, it should be set to TRUE on all systems. Otherwise, it should be set to FALSE on all systems where the field is available.

### Related Information

For more information on...	See...
the RANDOM function	<i>Teradata Vantage™ - SQL Functions, Expressions, and Predicates</i> , B035-1145

For more information on...	See...
Teradata® Unity™	<i>Teradata® Unity™ User Guide</i> , B035-2520

## ForeignTableErrorsData

Enables the logging of error or error with data. ForeignTableErrorsData also disables logging errors.

### Field Group

Native Object Store

### Valid Settings

Value	Description
0	Default. Log errors and data.
1	Disable logging.
2	Log errors and data.
3	Log errors. Do not log data.

### Default

0

### Changes Take Effect

After the next database restart.

### Usage Considerations

ForeignTableErrorsData data type is ByteInt.

Errors detected while accessing or parsing data in the object store are logged.

### Related Information

See [ForeignTableErrorsLimit](#).

## ForeignTableErrorsLimit

Limits the number of errors logged.

### Field Group

Native Object Store

**Valid Settings**

Value	Description
0	Use the default errors value, which is 10.
1 to 2,147,483,647	Limits the number of errors logged to the value set. Default value is 10.
-1	Unlimited. Logs errors and places no limit on the number of errors logged.

**Default**

0

**Changes Take Effect**

After the next database restart.

**Usage Considerations**

ForeignTableErrorsLimit data type is Integer.

**Related Information**See [ForeignTableErrorsData](#).**ForwardCredential**

Determines whether to forward authenticated user credential token to Vantage.

**Field Group**

General

**Valid Settings**

Setting	Description
TRUE	Enable forwarding of authenticated user credential token to Vantage.
FALSE	Disable forwarding of authenticated user credential token to Vantage.

**Default**

TRUE

**Changes Take Effect**

After the DBS Control record has been written.

**Usage Notes**

This field must be set to TRUE if you are using a single sign on mechanism to connect to an external server from Vantage. If you are not using a single sign on mechanism for logons, this field has no effect.

Forwarding of user credentials is available only for authentication methods that support credential delegation.

## Free Cylinder Cache Size

Determines the maximum number of cylinders that can be in the File System cache of free spool cylinders. Using such a cache reduces the number of times storage services must be called to allocate new spool cylinders.

### Field Group

File System

### Valid Range

1 through 1,000

### Default

100

### Changes Take Effect

After the next database restart.

## FreeSpacePercent

Specifies the amount of space on each cylinder that is to be left unused during the following operations:

Type	Operations
Utility or operation	<ul style="list-style-type: none"> <li>• FastLoad and MultiLoad</li> <li>• Table Rebuild</li> <li>• system reconfiguration</li> <li>• Ferret PACKDISK</li> <li>• MiniCylPack operations attempt to honor the FreeSpacePercent (FSP) setting, or the FSP value specified in CREATE TABLE and ALTER TABLE statements. However, if few cylinders are available, and storage space is limiting, MiniCylPack may not be able to honor that FSP.</li> </ul>
SQL	<ul style="list-style-type: none"> <li>• An ALTER TABLE that adds fallback protection.</li> <li>• A CREATE INDEX that defines or redefines any type of secondary index on a populated table.</li> <li>• Fallback creation during an INSERT ... SELECT into an empty table that is defined with fallback protection.</li> <li>• Index creation during an INSERT ... SELECT into an empty table that is defined with any type of secondary index.</li> </ul>

The reserved free space allows table data to expand on current table cylinders, preventing or delaying the need for additional table cylinders to be allocated, therefore preventing or delaying data migration operations associated with new cylinder allocations. Keeping new table data physically close to existing table data, and avoiding data migrations, can improve overall system performance.

### Field Group

File System

### Valid Range

0 through 75%

### Default

0%

### Changes Take Effect

After the DBS Control Record has been written, and during the next data load operation. Any operations in progress when the setting is changed are not affected.

**Note:**

After setting a non-zero value for the free space percentage, all subsequent operations listed above will respect that setting, and will continue to reserve free space beyond what table data requires. To have Vantage utilize the reserved free space for data storage and avoid data migrations, the free space percentage must be reduced after the initial data is loaded.

**Evaluating Free Space Percent Requirements**

Reserved free space allows tables to expand within their currently allocated cylinders. This can prevent or delay the need for additional cylinders to be allocated, which incurs the overhead of moving data to the new cylinders. Avoiding new cylinder allocations can improve overall system performance.

Choosing an appropriate FSP value for a table involves both performance and space considerations, and depends on the growth needs of the table:

- Reference tables that experience no modifications or growth require no FSP, so FSP can be zero for these types of tables. If the system is primarily these types of tables, set FreeSpacePercent to zero, and use the FREESPACE option of the CREATE TABLE and ALTER TABLE SQL statements to set a different FSP value for tables that will experience growth.
- Tables that are expected to experience large scale growth require higher FSP values than tables that grow to lesser degrees. However, larger FSP values consume more storage, so FSP choice should balance space and performance considerations.

A table that would require 100 cylinders of storage with 0% FSP, requires 134 cylinders when FSP is set to 25%. If FSP is 75%, that same table would require 400 cylinders. Ensure that the requisite cylinders are available to satisfy the storage overhead added by the FSP, or performance can suffer.

With time, the available free space may change, due to table modifications that do not honor the FreeSpacePercent setting. The AutoCylPack background task runs periodically to check and restore the FSP for tables. The Ferret PACKDISK command can be run manually to force FSP to be restored, or to temporarily set a table FSP to a different value. MiniCylPack may change the available free space if there is a shortage of cylinders on the system.

**Related Information**

For more information on...	See...
CREATE TABLE and ALTER TABLE	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.
Free Space Percent	<a href="#">SHOWFSP</a> , <i>Teradata Vantage™ - Database Administration</i> , B035-1093.
AutoCylPack	<a href="#">DisableAutoCylPack</a> .
MiniCylPack	<a href="#">MiniCylPackLowCylProd</a> .



## GlobalSpaceSoftLimitPercent

Specifies the percentage limit for permanent, spool, and temporary space above the maximum value that the system permits for any database or user.

### Note:

This field should be changed only under the direction of Teradata Support Center personnel.

### Field Group

General

### Valid Range

0 through 100 percent

### Default

0

### Changes Take Effect

After the DBS Control record has been written.

### Usage Notes

If the system involves long running transactions with high space consumption, you can increase the value of GlobalSpaceSoftLimitPercent to reduce the chances of transaction aborts due to a lack of available space.

Teradata recommends setting a non-zero soft limit, because that provides flexibility to modify limits without impacting data load applications that may otherwise report out-of-space errors.

The DBC space limit should be ample enough that there is enough soft limit cushion to handle the extra space usage without hitting space errors unexpectedly. For example, assume the soft limit is set to 10% system-wide. The DBA can ensure that the available space in DBC is around 10% using this setting.

### Related Information

For more information on...	See...
the SKEW option for DDL, such as CREATE DATABASE and CREATE USER	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.
related DBS Control fields	<ul style="list-style-type: none"> <li>• <a href="#">DefaultPermSkewLimitPercent</a></li> <li>• <a href="#">DefaultSpoolSkewLimitPercent</a></li> <li>• <a href="#">DefaultTempSkewLimitPercent</a></li> <li>• <a href="#">LegacySpaceAcctg</a></li> </ul>



## HashFuncDBC

---

**Note:**

This field can be modified only during system initialization (sysinit), performed only by Teradata Services personnel. System initialization destroys all user and dictionary data.

---

Defines the hashing function that the database uses.

**Field Group**

General

**Valid Settings**

Setting	Description
3	Kanji
4	International
5	pre-13.10 Universal
6	Universal (New Universal hash)

**Default**

6 (Universal)

Universal is the recommended hash.

## HJ2IMHJ

Determines whether the Optimizer preferentially chooses the In-Memory Hash Join method for performing hash joins and dynamic hash joins. Also determines whether the Optimizer evaluates the In-Memory Hash Join method as an alternative path for costing.

### Field Group

Performance

### Valid Settings

Setting	Description
TRUE	Hash join and dynamic hash joins are preferentially performed using the In-Memory Hash Join method whenever possible.
FALSE	All hash join and dynamic hash join methods, including In-Memory, are evaluated by the Optimizer, which chooses the most cost-effective method to use.

### Default

FALSE

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

When two tables are joined, the Optimizer selects the best method to accomplish the join by predicting and comparing the costs of all possible join methods (for example, merge join, hash join, in-memory hash join, product join, nested join) using various different potential data geographies (local, duplicated, and hash distributed) for the tables.

Setting this field to TRUE causes the Optimizer to use the In-Memory Hash Join method whenever possible, in preference to any other hash join method, regardless of costs. The In-Memory method offers generally superior performance over other hash-join methods. Setting this field to TRUE also avoids generation of cost-based plans for in-memory hash joins, and therefore can save time.

## HTMemAlloc

Specifies the percentage of cache memory to be allocated to a hash table that is used for a hash join. Hash joins are used as optimizations to a merge joins under specific conditions.

### Note:

This field should be changed only under the direction of the Teradata Support Center.

### Field Group

Performance

### Valid Range

0 through 10

A setting of zero prevents hash joins from being used as optimizations.

### Default

10

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

Hash joins can benefit performance when skew is low.

Larger values for HTMemAlloc allow hash join optimizations to be applied to larger tables.

The Optimizer uses the value of HTMemAlloc to determine the size of the hash table as follows:

Hash Table Size = (HTMemAlloc/100) \* 20 MB

20 MB is the maximum amount of memory set aside by Vantage for hash joins.

### Related Information

For more information on...	See...
SkewAllowance field	<a href="#">SkewAllowance</a> .
hash table size calculations and possible values	<i>Teradata Vantage™ - SQL Request and Transaction Processing</i> , B035-1142.

## IAMaxWorkloadCache

Defines the maximum size of the workload cache when performing analysis operations. This field is applicable to both the INITIATE INDEX ANALYSIS and INITIATE PARTITION ANALYSIS statements.

### Field Group

Performance

### Valid Range

32 through 187 MB

### Default

32 MB

### Changes Take Effect

After the DBS Control Record has been written.

## IdCol Batch Size

Indicates the size of a pool of numbers reserved by a vproc for assigning identity values to rows inserted into an identity column table.

Identity columns are used mainly to ensure row uniqueness by taking a system-generated unique value. They are valuable for generating simple unique indexes and primary and surrogate keys when composite indexes or keys are not desired. Identity columns are also useful for ensuring column uniqueness when merging several tables or to avoid significant preprocessing when loading and unloading tables. For more information on identity columns, see *Teradata Vantage™ - SQL Data Definition Language Syntax and Examples*, B035-1144.

### Field Group

General

### Valid Range

1 through 1,000,000

### Default

100,000

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

When the initial batch of rows for a bulk insert arrives on a PE/AMP vproc, the following occurs:

1. A range of numbers is reserved before processing the rows.
2. Each PE/AMP retrieves the next available value for the identity column from the IdCol table.
3. Each PE/AMP immediately updates this value with an increment equal to the IdCol Batch Size setting.

Bulk loads using MultiLoad, FastLoad, and INSERT... SELECT, have identity values assigned by the AMPs. For these types of loads, base the IdCol Batch Size setting on the number of AMPs in the system.

Bulk loads using TPump and iterated inserts have identity values assigned by the PEs. For these types of loads, base the setting on the number of PEs in the system.

### IdCol Batch Size Performance Implications

The IdCol Batch Size setting involves a trade-off between insert performance and potential gaps in the numbering of rows inserted into tables that have identity columns.

A larger setting results in fewer updates to DBC.IdCol in reserving batches of numbers for a load. This can improve the performance of bulk inserts into an identity column table. However, because the reserved numbers are kept in memory, unused numbers will be lost if a database restart occurs, resulting in a gap in the numbering of identity columns.





## InclNTforGrntOrRevokAll

Determines whether the NONTEMPORAL privilege is included when GRANT ALL and REVOKE ALL statements specify a user, database, view, or table. The NONTEMPORAL privilege allows use of NONTEMPORAL operations on temporal tables.

NONTEMPORAL operations delete rows, modify history rows, or change Transaction Time column values on temporal tables that include a Transaction Time column. These operations include NONTEMPORAL DELETE, NONTEMPORAL UPDATE, NONTEMPORAL INSERT, NONTEMPORAL CREATE TABLE AS, and NONTEMPORAL ALTER TABLE.

### Field Group

General

### Valid Settings

Setting	Description
TRUE	GRANT ALL and REVOKE ALL statements include the NONTEMPORAL privilege
FALSE	GRANT ALL and REVOKE ALL statements exclude the NONTEMPORAL privilege

### Default

FALSE

### Changes Take Effect

After the DBS Control Record has been written.

### Related Information

For more information on temporal operations, see *Teradata Vantage™ - Temporal Table Support*, B035-1182.

## InListRewriteOption

Specifies the type of IN-list processing that will be used for rewrite processing of queries that have IN-lists.

### Field Group

Performance

### Valid Range

0 through 4

Value	Description
0	Disable all IN-list rewrite processing
1	Rewrite IN-lists into subqueries. Vantage uses this type of processing for IN-lists that are not part of CASE expressions.
2	Rewrite IN-lists into outer joins. Vantage uses this type of processing for IN-lists that are part of CASE expressions. <b>Note:</b> Teradata recommends that you enable outer-join-based rewrite processing only when there are large IN-lists in CASE-expressions.
3	Options 1 and 2 are enabled
4	Reserved for use by Teradata Professional Services and Teradata Services personnel.

### Default

1

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

The default setting is appropriate for most situations, and generally provides the best query performance.

IN-list rewrite operations preclude predicate simplification and block optimization during query processing. So in rare cases, using an IN-list rewrite for a query with an IN-list can result in a performance degradation, compared to processing without the rewrite. In these cases, you should increase the value of InListRewriteThreshold to higher than the number values in the query.

### Related Information

- For more information on large IN-list processing, see *Teradata Vantage™ - SQL Request and Transaction Processing*, B035-1142.

- For information on other DBS Control fields related to IN-list processing, see [InListRewriteThreshold](#) and [MaxNumInListRewrite](#).

## InListRewriteThreshold

Specifies the minimum number of values in an IN-list in order for it to qualify for IN-list rewrite query processing.

### Field Group

Performance

### Valid Range

1 through 32767

### Default

512

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

The default setting is appropriate for most situations, and will give the best query performance.

IN-list rewrite processing precludes predicate simplification and block optimization during query processing. So in rare cases, using an IN-list rewrite for a query with an IN-list can result in a performance degradation, compared to processing without the rewrite. In these cases, you should increase the value of InListRewriteThreshold to a value greater than the number of values in the IN-list in the query.

### Related Information

- For more information on large IN-list processing, see *Teradata Vantage™ - SQL Request and Transaction Processing*, B035-1142.
- For information on other DBS Control fields related to IN-list processing, see [InListRewriteOption](#) and [MaxNumInListRewrite](#).

## IVMaxWorkloadCache

Defines the maximum size of the workload cache when performing validation operations.

### Field Group

Performance

### Valid Range

1 through 32 MB

### Default

1

### Changes Take Effect

After the DBS Control Record has been written.

## JSON\_AttributeSize

Denotes the size (number of characters) of attributes returned by the JSONExtractValue method and dot notation.

### Field Group

General

### Valid Range

0 through 32000 characters

### Default

4096

### Changes Take Effect

After the DBS Control Record has been written.

### Related Information

For More Information on...	See...
JSON data type	<i>Teradata Vantage™ - JSON Data Type</i> , B035-1150.
Related DBS Control fields	<a href="#">DisableJSONValidation</a> , <a href="#">EnableJSON</a> , <a href="#">JSON_MaxDepth</a> .

## JSON\_MaxDepth

Determines the level of JSON value nesting allowed during JSON syntax validation. If a JSON text exceeds this level of nesting, the system returns an error and does not create the JSON instance.

### Field Group

General

### Valid Range

0 through 32000, where 0 means no nesting is allowed.

### Default

512

### Changes Take Effect

After the DBS Control Record has been written.

### Related Information

For More Information on...	See...
JSON data type	<i>Teradata Vantage™ - JSON Data Type</i> , B035-1150.
Related DBS Control fields	<a href="#">DisableJSONValidation</a> , <a href="#">EnableJSON</a> , <a href="#">JSON_AttributeSize</a> .

## JournalDBSize

Determines the maximum size of permanent journal table multirow data blocks (DBs), in 512-byte sectors. For tables using journaling, rows are written to this journal during INSERT, UPDATE, and DELETE processing. Rows that are larger than JournalDBSize are stored in single-row data blocks, which are not limited by JournalDBSize.

### Field Group

File System

### Valid Range

18 through 512 sectors for systems that use small cylinders.

42 through 2047 sectors for systems that use large cylinders.

A sector is 512 bytes.

### Default

254 sectors

Setting this field to 0 causes Vantage to use the system default size for this type of data block.

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

When database tables are initially populated, Vantage stores as many rows as possible into each data block, until the block reaches the size specified by the various DB size settings in DBS Control. As tables are subsequently modified, rows can grow such that the existing data blocks would exceed the maximum size. When this happens, the data block is split, and roughly half the rows are moved to a new data block, with the result that the original and new data blocks are each one half of the original size. The result of this type of growth and splitting is that data blocks for heavily modified tables tend to be about 75% of the maximum size defined in DBS Control.

If DBs are compressed, this setting applies to the size of the uncompressed DBs.

If table rows are very long, or many rows are being manipulated, try increasing JournalDBSize. A larger size also can produce significant savings if the system is I/O bound.

In general, the maximum multirow data block size for journals should agree with the data row length. If the modified rows are short, the journal data block size can be small. If the modified rows are long, the journal data block size can be large.

If you base data block size on processing activity, the following rules are generally successful for good performance when the workload is mixed:

- PermDBSize should be a large number to optimize decision support, especially for queries involving full table scans. See [PermDBSize](#).



- JournalDBSize should be a low number to benefit analytic functions and High-Availability Transaction Processing (HATP) operations.

## LargeDepotCylsPerPdisk

Determines the number of Depot cylinders the file system allocates per pdisk (storage device) to contain large slots (1 MB). A large slot can hold several data blocks (DBs) during Depot operations.

The actual number of large-depot cylinders used per AMP is this value multiplied by the number of pdisks per AMP.

### Field Group

File System

### Valid Range

0 through 10

Setting this field to zero disables large Depot usage.

### Default

1 cylinder

### Changes Take Effect

After the next database restart.

### Usage Notes

The Depot is a set of transitional storage locations (a number of cylinders) used by the file system for performing in-place writes of DBs or WAL DBs (WDBs). An in-place write means that the changed DB is written back to exactly the same place on disk from which it was originally read. In-place writes are only performed for modifications to DBs that do not change the size of DBs, and therefore do not require any reallocation of space.

Writing the changed DB directly back to its original disk location can leave the data vulnerable to various hardware and system problems that can occur during system resets, such as a disk controller malfunctions or power failures. If such a problem occurred during the write operation, the data could be irretrievably lost.

The Depot protects against such data loss by allowing the file system to perform disk writes in two stages. First the changed DB (or WDB) is written to the Depot. After the data has been completely written to the Depot, it is written to its permanent location on the disk. If there is a problem while the data is being written to the Depot, the original data is still safe in its permanent disk location. If there is a problem while the data is being written to its permanent location, the changed data is still safe in the Depot.

During database startup, the Depot is examined to determine if any of the DBs or WDBs should be rewritten from the Depot to their permanent disk locations.

## LegacySpaceAcctg

Determines whether Vantage uses legacy or global space accounting. By default, Vantage uses global space accounting.

- Legacy space accounting makes cumulative space allocations to the AMPs that equal the maximum space value at allocation time, and the space limit for each AMP is set to the per-AMP quota. Vantage monitors the AMP space usage, and violations of space limitations are treated as exceptions, even if total use at the global database level is less than the maximum.
- Global space accounting allows AMPs to temporarily utilize space beyond per-AMP maximum space values on an as-needed basis, provided the space utilization does not exceed the database maximum. Space allocations and de-allocations for AMPs happen dynamically as needed in the background as modifications are made to database and user tables. The amount by which AMPs can utilize extra space is determined by the SKEW option in CREATE DATABASE and CREATE USER statements, or by the system default settings defined by DBS Control fields DefaultPermSkewLimitPercent, DefaultSpoolSkewLimitPercent, and DefaultTempSkewLimitPercent.

### Field Group

General

### Valid Settings

Setting	Description
TRUE	Vantage uses legacy space accounting, and prevents creation of databases and users with non-zero SKEW values. Additionally, DSA perm space errors during archive/restore operations cause the operations to abort.
FALSE	Vantage uses global space accounting, and allows creation of databases and users with non-zero SKEW values. Additionally, DSA perm space errors during archive/restore operations are ignored, allowing the operations to continue to run.

### Default

FALSE

### Changes Take Effect

After the DBS Control record has been written.

#### Note:

Changes to the value do not change the space accounting used for databases and users that were already created. If a database has been created with a non-zero SKEW value, it will use global space accounting and continue to do so, even if the value of LegacySpaceAcctg is subsequently changed to TRUE.

## Usage Notes

- Regardless of the field setting, databases and users created with a SKEW value of zero will use legacy space accounting.
- The legacy space accounting method does not have the overhead of dynamic need-based space allocations. It is best suited when the database is expected to have uniform data distribution across all the AMPs in the system, or the global limit is set high enough to allow for variability in data distribution due to skew in the data.
- The global space accounting method is best suited when the actual space usage is expected to be non-uniform across the AMPs.

## Related Information

For more information on...	See...
the SKEW option for DDL, such as CREATE DATABASE and CREATE USER	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.
related DBS Control fields	<ul style="list-style-type: none"> <li>• <a href="#">DefaultPermSkewLimitPercent</a></li> <li>• <a href="#">DefaultSpoolSkewLimitPercent</a></li> <li>• <a href="#">DefaultTempSkewLimitPercent</a></li> <li>• <a href="#">GlobalSpaceSoftLimitPercent</a></li> </ul>

## LegacyUnAuthBehavior

Controls whether some unauthenticated user names are logged in the DBC.EventLog table as they were entered or as "Non-existent User".

### Field Group

General

### Valid Settings

Setting	Description
TRUE	Preserve legacy Vantage behavior to log some unauthenticated user name values to DBC.EventLog.
FALSE	Log unauthenticated user name values as the literal "Non-existent User" in the DBC.EventLog table.

### Default

FALSE

### Changes Take Effect

After the DBS Control record has been written.

### Usage Notes

The name information is logged in the UserName and AuthUser columns of DBC.EventLog and in views that are based on this table, such as DBC.LogOnOffV.

In some cases, in database releases without the LegacyUnAuthBehavior field, user names that could not be authenticated were nevertheless logged and visible in the DBC.EventLog table. When a user logs in to the database and types a user name and password, a common mistake is to accidentally enter the password in the user name field. Because this would be considered an "unauthenticated user" by the database, legacy behavior can risk having these misplaced passwords logged to the database.

The default setting of FALSE for LegacyUnAuthBehavior prevents these user names from being visible in the log by recording them as "Non-existent User".

Change this field value to TRUE if you want to preserve the legacy behavior, and record some user name values in the log.

### Related Information

For more information about Data Dictionary tables and views, see *Teradata Vantage™ - Data Dictionary*, B035-1092.

## LimitInlistCVal

Limits the number of combined values in the IN list.

---

**Note:**

This setting should be changed only under the direction of the Teradata Support Center.

---

### Field Group

Performance

### Valid Range

1 through 1048576

### Default

1048576 values

### Changes Take Effect

After the DBS Control Record has been written. A system restart is not required.

## LockLogger [Deprecated]

Determines whether the Lock Logger feature is enabled or disabled. The Lock Logger logs delays caused by blocked database locks, and can help identify lock conflicts.

### Note:

The command-line Lock Logger utility (dumplocklog) and related DBS Control settings are deprecated, but remain available for compatibility with prior releases. Database locks should be logged using the Database Query Log (DBQL). This lock information can be accessed by means of the Teradata Viewpoint Lock Viewer portlet, or by querying the DBC.QrylockLogXMLV view.

- For more information on DBQL, see *Teradata Vantage™ - Database Administration*, B035-1093 and the BEGIN QUERY LOGGING statement in *Teradata Vantage™ - SQL Data Definition Language Syntax and Examples*, B035-1144.
- For more information on the Lock Viewer portlet, see *Teradata® Viewpoint User Guide*, B035-2206.

### Field Group

General

### Valid Settings

Setting	Description
TRUE	Enable the Lock Logger.
FALSE	Disable the Lock Logger. <b>Note:</b> If LockLogger is set to FALSE, the LockLogger Delay Filter, LockLogger Delay Filter Time, and LockLogSegmentSize DBS Control fields are ignored.

### Default

FALSE

### Changes Take Effect

After the next database restart.

### Usage Notes

Lock Logger runs as a background task, recording lock information in a table. Use the Lock Logger (dumplocklog) utility to create or designate a table to be used for storing lock log entries.

LockLogger is useful for troubleshooting problems such as determining whether locking conflicts are causing high overhead.

Some values in the lock log table represent internal IDs for the object on which the lock was requested. The lock log table defines the lock holder and the lock requester as transaction session numbers. The lock log table can be joined with the DBC.DBase, DBC.TVM, and DBC.EventLog tables to gain additional information about the object IDs and transaction session numbers.

**Related Information**

- [LockLogger Delay Filter \[Deprecated\]](#)
- [LockLogger Delay Filter Time \[Deprecated\]](#)
- [LockLoggerFlushRate](#)
- [LockLogSegmentSize \[Deprecated\]](#)



## LockLogger Delay Filter [Deprecated]

Enables or disables log filtering of blocked lock requests based on delay time.

### Note:

The command-line Lock Logger utility (dumplocklog) and related DBS Control settings are deprecated, but remain available for compatibility with prior releases. Database locks should be logged using the Database Query Log (DBQL). This lock information can be accessed by means of the Teradata Viewpoint Lock Viewer portlet, or by querying the DBC.QrylockLogXMLV view.

- For more information on DBQL, see *Teradata Vantage™ - Database Administration*, B035-1093 and the BEGIN QUERY LOGGING statement in *Teradata Vantage™ - SQL Data Definition Language Syntax and Examples*, B035-1144.
- For more information on the Lock Viewer portlet, see *Teradata® Viewpoint User Guide*, B035-2206.

### Field Group

General

### Valid Settings

Setting	Description
TRUE	Enable filtering
FALSE	Disable filtering

### Default

FALSE.

### Note:

If the LockLogger field is set to FALSE, then the settings for the LockLogger Delay Filter field and the LockLogger Delay Filter Time field are ineffective.

### Changes Take Effect

After the DBS Control Record has been written.

### Related Information

- [LockLogger \[Deprecated\]](#)
- [LockLogger Delay Filter Time \[Deprecated\]](#)
- [LockLoggerFlushRate](#)
- [LockLogSegmentSize \[Deprecated\]](#)



## LockLogger Delay Filter Time [Deprecated]

Specifies a threshold time delay value (in seconds). Lock requests that are blocked for less than this amount of time are not logged. This prevents the log table from being filled with data for insignificantly blocked locks.

### Note:

The command-line Lock Logger utility (dumplocklog) and related DBS Control settings are deprecated, but remain available for compatibility with prior releases. Database locks should be logged using the Database Query Log (DBQL). This lock information can be accessed by means of the Teradata Viewpoint Lock Viewer portlet, or by querying the DBC.QrylockLogXMLV view.

- For more information on DBQL, see *Teradata Vantage™ - Database Administration*, B035-1093 and the BEGIN QUERY LOGGING statement in *Teradata Vantage™ - SQL Data Definition Language Syntax and Examples*, B035-1144.
- For more information on the Lock Viewer portlet, see *Teradata® Viewpoint User Guide*, B035-2206.

### Field Group

General

### Valid Range

0 through 1,000,000 seconds

IF you set LockLogger Delay Filter Time to...	THEN blocked lock requests with a delay of...
0 seconds	0 seconds would not be logged.
10 seconds	10 seconds or less would not be logged.
1,000,000 seconds	1,000,000 seconds or less would not be logged.

### Default

0 seconds

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

The LockLogger Delay Filter Time field has a dependency on the Lock Logger Delay field and the LockLogger field.

IF the LockLogger field is...	THEN LockLogger Delay Filter Time field is...
FALSE, and the LockLogger Delay Filter field is TRUE	ineffective.
TRUE, and the LockLogger Delay Filter field is FALSE	ineffective.
TRUE, and the LockLogger Delay Filter field is TRUE	effective.

## Related Information

- [LockLogger \[Deprecated\]](#)
- [LockLogger Delay Filter \[Deprecated\]](#)
- [LockLoggerFlushRate](#)
- [LockLogSegmentSize \[Deprecated\]](#)

## LockLoggerFlushRate

Determines the frequency for writing entries to the system table DBC.DBQLXMLLOCKTbl.

the system table DBC.DBQLXMLLOCKTbl stores blocker/blocked transaction and lock information, and is also useful for getting deadlock information.

### Field Group

General

### Valid Range

600 through 3600 seconds

Recommended value is 1200 seconds or more.

### Default

1200 seconds

### Changes Take Effect

After the DBS Control Record has been written. However, DBQL will not become aware of the new setting until the current timer expires.

### Usage Notes

You must monitor the system table DBC.DBQLXMLLOCKTbl for performance and data loss and adjust LockLoggerFlushRate accordingly, in increments of 600.

You can access DBC.DBQLXMLLOCKTbl Through the Viewpoint lock viewer portlet or query the system table DBC.DBQLXMLLOCKTbl or the view DBC.QrylockLogXMLV.

**Related Information**

- [DBQLFlushRate](#)
- [LockLogger \[Deprecated\]](#)
- [LockLogger Delay Filter \[Deprecated\]](#)
- [LockLogger Delay Filter Time \[Deprecated\]](#)
- [LockLogSegmentSize \[Deprecated\]](#)

## LockLogSegmentSize [Deprecated]

Specifies the size of the Lock Logger segment. This field allows you to control the size of the buffer that is used to store lock information.

### Note:

The command-line Lock Logger utility (dumplocklog) and related DBS Control settings are deprecated, but remain available for compatibility with prior releases. Database locks should be logged using the Database Query Log (DBQL). This lock information can be accessed by means of the Teradata Viewpoint Lock Viewer portlet, or by querying the DBC.QrylockLogXMLV view.

- For more information on DBQL, see *Teradata Vantage™ - Database Administration*, B035-1093 and the BEGIN QUERY LOGGING statement in *Teradata Vantage™ - SQL Data Definition Language Syntax and Examples*, B035-1144.
- For more information on the Lock Viewer portlet, see *Teradata® Viewpoint User Guide*, B035-2206.

### Field Group

General

### Valid Range

64 KB through 1024 KB

### Default

64 KB

### Changes Take Effect

After the next database restart.

### Usage Notes

If the LockLogger field is set to FALSE, then the setting for the LockLogSegmentSize field is ineffective.

### Related Information

- [LockLogger \[Deprecated\]](#)
- [LockLogger Delay Filter \[Deprecated\]](#)
- [LockLogger Delay Filter Time \[Deprecated\]](#)
- [LockLoggerFlushRate](#)

# MaxDecimal

Defines the maximum number of decimal digits in the default maximum value used in expression typing.

## Field Group

General

## Default

15 decimal digits

## Changes Take Effect

After the next database restart.

**Note:**

After you perform a system initialization (sysinit) that re-initializes DBS Control, MaxDecimal is set to zero.

## Usage Notes

This MaxDecimal value...	Sets a default DECIMAL maximum size for expression evaluation to...
0	15.
15	15.
18	18.
38	38.

For more information on the DECIMAL data type, see *Teradata Vantage™ - Data Types and Literals*, B035-1143.

## MaxDownRegions

Vantage can isolate some file system errors to a specific data or index subtable, or to a contiguous range of rows ("region") in a data or index subtable. In these cases, Vantage marks only the affected subtable or region down. This improves system performance and availability by allowing transactions that do not require access to the down subtable or rows to proceed, without causing a database crash that would require a system restart.

However, if several regions in a subtable are marked down, it could indicate a fundamental problem with the subtable itself. Therefore, when a threshold number of down regions is exceeded per AMP, the entire subtable is marked down on all AMPs, making it unavailable to most SQL queries. This threshold can be adjusted by means of the MaxDownRegions field in the General fields group of the DBS Control utility.

The MaxDownRegions field determines the value of this threshold. If the number of down regions per AMP in a subtable exceeds MaxDownRegions, the entire subtable will be marked down and become inaccessible.

### Field Group

General

### Valid Range

0 through 12

When MaxDownRegions is set to 0, any fatal file system error in a data subtable causes the subtable to be marked down on all AMPs.

### Default

6

### Changes Take Effect

After the next database restart.

### Usage Notes

If a data or index subtable is marked down on all AMPs, it is highly recommended that the associated table be rebuilt using the Table Rebuild utility after the source of the problem has been corrected. Alternatively, the table can be dropped and restored from a backup copy. If the problem is with an index subtable, the index can be dropped and recreated.

Tables that are marked down at the time they are rebuilt will remain down after the rebuild. To clear the down status, use the ALTER TABLE ... RESET DOWN statement after the table has been rebuilt.

For more information on the Table Rebuild utility, see [Table Rebuild \(rebuild\)](#).



## MaxJoinTables

Influences the maximum number of tables that can be joined per query block.

### Field Group

Performance

### Valid Settings

0, and 64 through 128

Zero means MaxJoinTables will use the system default upper limit of 128.

### Default

0

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

MaxJoinTables sets a system-wide upper bound on the MaxJoinTables cost parameter used during optimization. The value 0 is mapped internally to 128.

If the RevertJoinPlanning DBS Control field is set to TRUE, the maximum join tables is 64, regardless of the MaxJoinTables setting.

### Related Information

For more information on...	See...
RevertJoinPlanning	<a href="#">RevertJoinPlanning.</a>
join planning and optimization	<i>Teradata Vantage™ - SQL Request and Transaction Processing</i> , B035-1142.
database limits	<i>Teradata Vantage™ - SQL Fundamentals</i> , B035-1141.

## MaxLoadAWT

Specifies the combined number of AMP Worker Tasks (AWTs) that can be used by FastLoad and MultiLoad at any time. It allows more FastLoad, MultiLoad, and FastExport tasks (jobs) to run concurrently, and sets a limit on AWT usage to prevent excessive consumption or exhaustion of AWT resources.

This field also acts as a “switch” on the function of MaxLoadTasks field:

- When MaxLoadAWT set to zero, the number of load utilities that can run concurrently is controlled entirely by the MaxLoadTasks field. In this case, MaxLoadTasks specifies the maximum number of combined FastLoad, MutiLoad, and FastExport jobs that can run concurrently.
- When MaxLoadAWT is set to an integer greater than zero, MaxLoadTasks applies only to the combined number of FastLoad and MultiLoad jobs, which are also limited by the MaxLoadAWT setting. In this case, the number of FastExport jobs that can run is always 60 minus the number of combined FastLoad and MultiLoad jobs currently running.

---

### Note:

Throttle rules for load utility concurrency set in the Teradata Viewpoint Workload Designer portlet override the MaxLoadAWT setting.

---

## Field Group

General

## Valid Settings

If MaxLoadAWT is set to a non-zero value, it should be a value greater than or equal to five, which allows at least one FastLoad and one MultiLoad job to run concurrently.

The maximum allowable value is 60% of the total AWTs per AMP. By default, the maximum number of AWTs started for each AMP vproc is 80, so the default maximum value for MaxLoadAWT is 48.

## Default

0

## Changes Take Effect

After the DBS Control Record has been written.

## Usage Notes

Consider using MultiLoad, rather than FastLoad, especially in cases of many small load jobs. MultiLoad generally consumes fewer AWTs per job than FastLoad.

The MaxLoadAWT field works together with the MaxLoadTasks field to limit the number of concurrent load utilities allowed to run:

- If MaxLoadAWT is zero (the default):
  - MaxLoadTasks can be an integer from zero through 15.

- The MaxLoadTasks field specifies the maximum number of combined FastLoad, MultiLoad, and FastExport jobs that can run concurrently.
- The system does not consider the number of available AWTs when limiting the number of load utilities that can run concurrently.
- If MaxLoadAWT is greater than zero:
  - MaxLoadTasks can be an integer from zero through 30.
  - The MaxLoadTasks field sets the maximum number of combined FastLoad and MultiLoad jobs that can run concurrently. MaxLoadTasks does not directly limit the number of FastExport jobs that can run.
  - The number of combined FastLoad and MultiLoad jobs that can run concurrently is limited by the values of both the MaxLoadTasks field and the MaxLoadAWT field. When either limit is met, no further FastLoad or MultiLoad jobs are allowed to start until the limiting factor is reduced.
  - The maximum number of load utility jobs of any type—FastLoad, MultiLoad, or FastExport—that can run concurrently is 60. Consequently, the number of FastExport jobs allowed to run at any time is 60 minus the number of combined FastLoad and MultiLoad jobs that are running.

For example, if the sum of currently running FastLoad and MultiLoad jobs is 29, the number of FastExport jobs that can be started is 31 (60 minus 29), regardless of the MaxLoadAWT and MaxLoadTasks settings.

  - If MaxLoadAWT is set to anything greater than zero, it can only be reset to zero if MaxLoadTasks is 15 or less.

Because load utilities share system resources with other system work, such as tactical and DSS queries, limiting the number of load utility jobs can help ensure sufficient system resources are available for other work.

## About AWTs

AMP Worker Tasks (AWTs) are processes dedicated to servicing the Vantage work requests. A fixed number of AWTs are pre-allocated during Teradata system initialization for each AMP vproc. Each AWT looks for a work request to arrive in the Teradata system, services the request, and then looks for another. An AWT can process requests of any work type.

The number of AWTs required by FastLoad and MultiLoad changes as their jobs run. More AWTs are required in the early phases of the jobs than in the later phases. Vantage dynamically calculates the total AWTs required by active jobs, and allows more jobs to start as AWTs become available. If MaxLoadAWT is greater than zero, new FastLoad and MultiLoad jobs are rejected when the MaxLoadAWT limit is reached, regardless of the MaxLoadTasks setting. Therefore, FastLoad and MultiLoad jobs may be rejected before MaxLoadTasks limit is reached.

For example, FastLoad and MultiLoad require different numbers of AWTs at different phases of execution. The following table shows how many AWTs are used at each phase.

Load Utility and Phase	Number of AWTs Required
FastLoad: Loading	3

Load Utility and Phase	Number of AWTs Required
FastLoad: End Loading	1
MultiLoad: Acquisition	2
MultiLoad: Application	1 per target table

Assume that MaxLoadAWT = 48 and MaxLoadTasks = 30. The list below shows some permitted combinations of load utility jobs. The limiting conditions for each combination is shown in bold:

- 16 FastLoads in Loading phase  
16 concurrent load tasks  
**48 AWTs in use:**  $(16 \times 3)$
- 9 FastLoads in Loading phase and 21 FastLoads in End Loading phase  
**30 concurrent load tasks**  
**48 AWTs in use:**  $(9 \times 3) + (21 \times 1)$
- 24 MultiLoads in Acquisition phase  
24 concurrent load tasks  
**48 AWTs in use:**  $24 \times 2$
- 5 MultiLoads in Acquisition phase and 25 MultiLoads in Application phase  
**30 concurrent load tasks**  
35 AWTs in use:  $(5 \times 2) + (25 \times 1)$
- 6 FastLoads in Loading phase and 15 MultiLoads in Acquisition phase  
21 concurrent load tasks  
**48 AWTs in use:**  $(6 \times 3) + (15 \times 2)$

## Related Information

For more information on...	See...
Load utilities (MultiLoad, FastLoad, FastExport, and Teradata Parallel Transporter)	<ul style="list-style-type: none"> <li>• <i>Teradata® MultiLoad Reference</i>, B035-2409.</li> <li>• <i>Teradata® FastLoad Reference</i>, B035-2411.</li> <li>• <i>Teradata® FastExport Reference</i>, B035-2410.</li> <li>• <i>Teradata® Parallel Transporter User Guide</i>, B035-2445</li> </ul>

## MaxLoadTasks

Specifies the combined number of FastLoad, MultiLoad, and FastExport tasks (jobs) that are allowed to run concurrently on Vantage.

---

### Note:

This field is ignored if the throttles category set in the Teradata Viewpoint Workload Designer portlet is enabled.

---

### Field Group

General

### Default

5 tasks

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

If MaxLoadTasks is set to 0, no load utilities can be started.

The MaxLoadTasks field works together with the MaxLoadAWT field to limit the number of concurrent load utilities allowed to run:

- If MaxLoadAWT is zero (the default):
  - MaxLoadTasks can be an integer from zero through 15.
  - The MaxLoadTasks field specifies the maximum number of combined FastLoad, MultiLoad, and FastExport jobs that can run concurrently.
  - The system does not consider the number of available AWTs when limiting the number of load utilities that can run concurrently.
- If MaxLoadAWT is greater than zero:
  - MaxLoadTasks is an integer from zero through 30.
  - The MaxLoadTasks field sets the maximum number of combined FastLoad and MultiLoad jobs that can run concurrently. MaxLoadTasks does not directly limit the number of FastExport jobs that can run.
  - The number of combined FastLoad and MultiLoad jobs that can run concurrently is limited by the values of both the MaxLoadTasks field and the MaxLoadAWT field. When either limit is met, no further FastLoad or MultiLoad jobs are allowed to start until the limiting factor is reduced.
  - The maximum number of load utility jobs of any type—FastLoad, MultiLoad, or FastExport—that can run concurrently is 60. Consequently, the number of FastExport jobs allowed to run at any time is 60 minus the number of combined FastLoad and MultiLoad jobs that are running.

For example, if the sum of currently running FastLoad and MultiLoad jobs is 29, the number of FastExport jobs that can be started is 31 (60 minus 29), regardless of the MaxLoadAWT and MaxLoadTasks settings.

- If MaxLoadAWT is set to anything greater than zero, it can only be reset to zero if MaxLoadTasks is 15 or less.

## About AWTs

AMP Worker Tasks (AWTs) are processes dedicated to servicing the Vantage work requests. A fixed number of AWTs are pre-allocated during Teradata system initialization for each AMP vproc. Each AWT looks for a work request to arrive in the Teradata system, services the request, and then looks for another. An AWT can process requests of any work type.

The number of AWTs required by FastLoad and MultiLoad changes as their jobs run. More AWTs are required in the early phases of the jobs than in the later phases. Vantage dynamically calculates the total AWTs required by active jobs, and allows more jobs to start as AWTs become available. If MaxLoadAWT is greater than zero, new FastLoad and MultiLoad jobs are rejected when the MaxLoadAWT limit is reached, regardless of the MaxLoadTasks setting. Therefore, FastLoad and MultiLoad jobs may be rejected before MaxLoadTasks limit is reached.

For example, FastLoad and MultiLoad require different numbers of AWTs at different phases of execution. The following table shows how many AWTs are used at each phase.

Load Utility and Phase	Number of AWTs Required
FastLoad: Loading	3
FastLoad: End Loading	1
MultiLoad: Acquisition	2
MultiLoad: Application	1 per target table

Assume that MaxLoadAWT = 48 and MaxLoadTasks = 30. The list below shows some permitted combinations of load utility jobs. The limiting conditions for each combination is shown in bold:

- 16 FastLoads in Loading phase  
16 concurrent load tasks  
**48 AWTs in use:** (16 x 3)
- 9 FastLoads in Loading phase and 21 FastLoads in End Loading phase  
**30 concurrent load tasks**  
**48 AWTs in use:** (9 x 3) + (21 x 1)
- 24 MultiLoads in Acquisition phase  
24 concurrent load tasks  
**48 AWTs in use:** 24 x 2

- 5 MultiLoads in Acquisition phase and 25 MultiLoads in Application phase

**30 concurrent load tasks**

35 AWTs in use:  $(5 \times 2) + (25 \times 1)$

- 6 FastLoads in Loading phase and 15 MultiLoads in Acquisition phase

21 concurrent load tasks

**48 AWTs in use:**  $(6 \times 3) + (15 \times 2)$

## Related Information

For more information on...	See...
Load utilities (MultiLoad, FastLoad, FastExport, and Teradata Parallel Transporter)	<ul style="list-style-type: none"> <li>• <i>Teradata® MultiLoad Reference</i>, B035-2409.</li> <li>• <i>Teradata® FastLoad Reference</i>, B035-2411.</li> <li>• <i>Teradata® FastExport Reference</i>, B035-2410.</li> <li>• <i>Teradata® Parallel Transporter User Guide</i>, B035-2445.</li> </ul>

## MaxMLOADXAWT

Specifies the number of AMP Worker Tasks (AWTs) that can be used by the MultiLoad Extended Protocol (MLOADX) at any time. It allows more tasks (jobs) to run concurrently, and sets a limit on AWT usage to prevent excessive consumption or exhaustion of AWT resources.

### Note:

AWT Resource Limit rules set in the Teradata Viewpoint Workload Designer portlet override the MaxMLOADXAWT setting.

### Field Group

General

### Valid Settings

0 through 70% of the number of AWTs started per AMP on your system.

The number of AWTs started per AMP is normally 80, however this can be modified by Teradata Services personnel for special circumstances.

### Default

60% of the number of AWTs started per AMP.

Because there are usually 80 AWTs started per AMP, the default value for MaxLoadAWT is typically 48.

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

This field is enforced only when MLOADXUtilityLimits is set to TRUE. In these cases, the setting of MaxLoadAWT does not apply to MLOADX jobs.

This field is ignored if the throttles category set in the Teradata Viewpoint Workload Designer portlet is enabled.

### Related Information

For more information on...	See...
Load utilities (MultiLoad, FastLoad, FastExport, and Teradata Parallel Transporter)	<ul style="list-style-type: none"> <li>• <i>Teradata® MultiLoad Reference</i>, B035-2409.</li> <li>• <i>Teradata® FastLoad Reference</i>, B035-2411.</li> <li>• <i>Teradata® FastExport Reference</i>, B035-2410.</li> <li>• <i>Teradata® Parallel Transporter User Guide</i>, B035-2445.</li> </ul>



## MaxMLOADXTasks

Specifies the number of MultiLoad Extended Protocol (MLOADX) tasks (jobs) that are allowed to run concurrently on Vantage.

### Note:

Throttle rules for load utility concurrency set in the Teradata Viewpoint Workload Designer portlet override the MaxMLOADXTasks setting.

### Field Group

General

### Valid Range

0 through 120

### Default

30 tasks

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

This field is enforced only when MLOADXUtilityLimits is set to TRUE. In these cases, the setting of MaxLoadTasks does not apply to MLOADX jobs.

This field is ignored if the throttles category set in the Teradata Viewpoint Workload Designer portlet is enabled.

### Related Information

For more information on...	See...
Load utilities (MultiLoad, FastLoad, FastExport, and Teradata Parallel Transporter)	<ul style="list-style-type: none"> <li>• <i>Teradata® MultiLoad Reference</i>, B035-2409.</li> <li>• <i>Teradata® FastLoad Reference</i>, B035-2411.</li> <li>• <i>Teradata® FastExport Reference</i>, B035-2410.</li> <li>• <i>Teradata® Parallel Transporter User Guide</i>, B035-2445.</li> </ul>

## MaxNumInListRewrite

Specifies the maximum number of IN-lists in a single query that can use IN-list rewrite query processing.

### Field Group

Performance

### Valid Range

0 through 100

### Default

2

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

Increasing the number of per-query IN-lists in rewrite processing requires more search space for join planning. Large values for MaxNumInListRewrite can lead to an increase in query parsing time. Teradata recommends that you keep this value under 10.

Setting this field to 0 effectively disables IN-list rewrite processing.

### Related Information

- For more information on large IN-list processing, see *Teradata Vantage™ - SQL Request and Transaction Processing*, B035-1142.
- For information on other DBS Control fields related to IN-list processing, see [InListRewriteOption](#) and [InListRewriteThreshold](#).

## MaxParseTreeSegs

Defines the maximum number of parse tree segments that the parser allocates while parsing a request.

### Field Group

Performance

### Valid Range

12 through 12,000

### Default

3,000

### Changes Take Effect

For new sessions after the DBS Control Record has been written. Any sessions in progress are not affected.

### Usage Notes

This field value should be increased if large, complex queries generate insufficient memory errors (error numbers 3710 and 3711). The value can also be increased to provide additional memory, if needed, for parser activities.

## MaxRequestsSaved

Specifies the maximum number of request-to-step cache entries that can be saved per PE.

### Field Group

Performance

### Valid Range

The valid range depends on the amount of memory on the node that runs the control AMP. The control AMP is the lowest numbered ONLINE AMP on the system.

Control AMP Node Memory	Valid Range
250 GB or greater	450 through 6000
Less than 250 GB	300 through 2000

The value must be a multiple of 10.

### Default

Control AMP Node Memory	Default
250 GB or greater	6000
Less than 250 GB	600

### Changes Take Effect

After the next database restart.

## MaxRowHashBlocksPercent

Specifies the proportion of available locks a transaction can use for rowhash locks before the transaction is automatically aborted. This setting protects Vantage against misbehaving database transactions.

To run, every query must acquire a lock on the objects upon which it will act. There are different types of locks: rowhash, rowrange, table, and database. A special table is used to keep track of the locks that have been applied. This lock table is maintained in memory for each AMP.

If a transaction performs a large number of single-row updates without closing the transaction, the lock table can fill up with rowhash locks on one or more AMPs. This prevents other transactions from running, and no further work can be accomplished until the problematic transaction is aborted to make space in the lock tables. This generally requires a database restart, unless the system returns an error, in which case the locks held by the query will be freed.

The Lock Manager Fault Isolation feature of Teradata monitors the number of locks each transaction uses. If the proportion of rowhash locks used reaches the threshold defined by MaxRowHashBlocksPercent, the transaction is automatically aborted. Other transactions are not affected, and the system need not be restarted.

### Field Group

General

### Default

50%, which means that a transaction will be stopped if its rowhash locks require more than 50% of the maximum space allowed in the lock table on any AMP.

To allow transactions to acquire more locks, set MaxRowHashBlocksPercent to a greater value.

### Changes Take Effect

After the DBS Control record has been written.

## MaxSetQueryBandSize

Defines the maximum query band size in kilobytes that can be set with the SET QUERY\_BAND statement.

### Field Group

General

### Valid Range

2 - 4 KB

### Default

3 KB

### Changes Take Effect

When the DBS Control record is written. Changes are effective for new sessions that begin after the change.

### Usage Notes

HELP SESSION displays several fields relating to query bands: QueryBand, Transaction QueryBand, Session QueryBand, and Profile QueryBand. The QueryBand field concatenates the values of Transaction QueryBand and Session QueryBand fields. Setting MaxSetQueryBandSize to a value greater than 2 KB can result in truncation of the concatenated QueryBand field in the output of HELP SESSION. To find the full concatenated query band string for the current transaction, session, and profile, use the GetQueryBand function.

### Related Information

- For more information about query bands, see *Teradata Vantage™ - Database Administration*, B035-1093 and *Teradata Vantage™ - SQL Data Definition Language Syntax and Examples*, B035-1144.
- For more information about the GetQueryBand function, see *Teradata Vantage™ - Application Programming Reference*, B035-1090.

## MaxSyncWALWrites

Determines the maximum number of outstanding WAL log writes to allow before tasks requiring synchronous writes are delayed to achieve better buffering.

### Field Group

File System

### Valid Range

1 through 40

### Default

5

### Changes Take Effect

After the next database restart.

### Usage Notes

If the number of outstanding WAL log writes is less than or equal to the MaxSyncWALWrites value, requests for synchronous operations will append the record to the current buffer, force a new buffer, and issue a synchronous write on the current buffer.

If the number of outstanding writes is greater than the MaxSyncWALWrites value, the record will be appended, but the write will be delayed until one of the outstanding write requests is completed.

## Memory Limit Per Transaction

Specifies the maximum amount of in-memory, temporary storage that the Relay Services Gateway (RSG) can use to store the records for one transaction. If the transaction exhausts this amount, the transaction data moves to a disk (or spill) file. This limit prevents a few, large transaction from swamping the memory pool.

### Field Group

General

### Valid Range

0 through 127 pages

### Default

2 pages

### Changes Take Effect

After the next database restart.



## MergeBlockRatio

Specifies the maximum size of a new data block that results from the merging of several smaller blocks. The size is expressed as the ratio of the new block size to the maximum block size specified in the table definition or, if no maximum size is specified in the table definition, by the PermDBSize setting in DBS Control.

---

### Note:

Data blocks are never merged for spool, global temporary, or volatile tables.

---

### Field Group

File System

### Valid Settings

Setting	Description
0	Resets MergeBlockRatio to the Teradata default, currently 60%.
1 through 100%	The maximum size of a data block that results from merged smaller blocks, as a percentage of the maximum data block size for the table.

### Default

60%

### Changes Take Effect

When the DBS Control record is written.

### Usage Notes

During normal database operations, the data blocks that store table rows on cylinders can split and shrink, resulting in many blocks which are far smaller than the maximum allowed data block size. Full table modify operations for tables with several small data blocks require more disk I/O than would be required if the table rows were stored on fewer and larger data blocks. Vantage can merge the small data blocks of these tables automatically during full table modify operations, which can result in reduced I/O overhead and improved database performance.

The maximum size for a multirow data block can be defined on a per-table basis with the DATABLOCKSIZE option of CREATE TABLE and ALTER TABLE statements. The maximum size for data blocks in tables that do not have a specified DATABLOCKSIZE is defined by the PermDBSize field in DBS Control.

Data block merging applies only to permanent and permanent journal tables.

The MergeBlockRatio is the largest size to which Vantage will merge small data blocks, expressed as a proportion of this maximum data block size.

The merge block ratio also can be defined on a per-table basis with the MERGEBLOCKRATIO option of CREATE TABLE and ALTER TABLE statements. The ratio used for tables that do not have a specified MERGEBLOCKRATIO is defined by the MergeBlockRatio field in DBS Control.

If DisableMergeBlocks is TRUE, the MergeBlockRatio field setting is ignored.

### Related Information

For more information on...	See...
the DATABLOCKSIZE and MERGEBLOCKRATIO options of CREATE TABLE and ALTER TABLE	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.
PermDBSize field	<a href="#">PermDBSize</a> .
Disabling automatic data block merging	<a href="#">DisableMergeBlocks</a> .

## MetaDataCapture

Controls which types of database requests include metadata used by Teradata® Unity™.

### Note:

This field is effective only when the CDCProtocol field is set to enable Teradata® Unity™ Change Data Capture protocol.

### Field Group

General

### Valid Settings

Setting	Description
0	Never include Teradata® Unity™ metadata. This is the default.
1	Include metadata with data definition language (DDL) and data manipulation language (DML) requests.
2	Include metadata with all database requests.

### Default

0

### Changes Take Effect

After the next database restart.

### Usage Notes

Teradata Unity is an integrated product that enables users to orchestrate a multi-system Teradata environment with load balancing, high system availability, workload optimization, and disaster recovery.

### Related Information

- For more information on related DBS Control fields, see [CDCProtocol](#) and [ClientResetAction](#).
- For more information on replication and Teradata® Unity™, see *Teradata® Unity™ User Guide*, B035-2520.

## MinDBSectsToCompress

Specifies the minimum size DB that will be compressed. DBs smaller than this size will not be compressed, because the overhead of having to compress and uncompress the blocks during DML operations is not worth the minimal space savings that might be achieved.

### Field Group

Compression

### Valid Range

2 through 255 sectors

### Default

16

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

BLC enables data compression at the data block (DB) level of the Teradata file system. Compression reduces the amount of storage required for a given amount of data. The BlockLevelCompression field of DBS Control enables and disables BLC.

Compression is also subject to the criteria specified by the [MinPercentCompReduction](#) DBS Control field. DBs not meeting these minimum criteria will not be compressed, even if they are larger than the size specified by MinDBSectsToCompress.

The PermDBSize and JournalDBSize field settings in DBS Control apply to the uncompressed size of these types of DBs.

### Related Information

For more information on...	See...
BlockLevelCompression field	<a href="#">BlockLevelCompression.</a>
COMPRESS and UNCOMPRESS commands	<a href="#">Ferret Utility (ferret).</a>

## MiniCylPackHighWaterMark

Determines the number of free cylinders above which MiniCylPack is not active.

### Field Group

File System

**Valid Range**

1 - 65535 cylinders

**Default**

100 cylinders

**Changes Take Effect**

After the DBS Control Record has been written.

**Usage Notes**

The value of this field should always be higher than MiniCylPackLowCylProd.

**Related Information**

For more information on...	See...
MiniCylPack	<i>Teradata Vantage™ - Database Administration</i> , B035-1093, and the DBS Control field: <a href="#">MiniCylPackLowCylProd</a>

## MiniCylPackLowCylProd

Determines the number of free cylinders below which the File System will perform a MiniCylPack operation. MiniCylPack attempts to free additional cylinders by packing cylinders that are currently in use.

### Field Group

File System

### Valid Range

0 through 65,535 free cylinders

Setting	Result
0	MiniCylPack does not run until a task needs a cylinder and none are available. In these cases, the requesting task is forced to wait until MiniCylPack has completed. When MiniCylPack runs while a task is waiting, the number of cylinders that can be scanned is unlimited. If necessary, MiniCylPack scans the entire disk in an attempt to free a cylinder.
a nonzero value	<p>MiniCylPack runs in the background. When running in this mode, each MiniCylPack scan packs a maximum of 20 cylinders.</p> <p>If MiniCylPack cannot free a cylinder, but free cylinders still exist, it does not run until another cylinder allocation request notices that the number of free cylinders has fallen below MiniCylPackLowCylProd.</p> <p>Setting this field to a low value reduces the impact of MiniCylPack on performance. However, there is a risk that free cylinders will not be available for tasks that require them. This would cause MiniCylPack to run while tasks are waiting, seriously impacting the response time of such tasks.</p>

### Default

10 free cylinders

### Recommendation

Set this value to no more than 20 free cylinders.

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

MiniCylPack is a background task that runs when the available number of free cylinders is low, as determined by the value of MiniCylPackLowCylProd. Because it runs before free cylinders are completely exhausted, MiniCylPack attempts to free cylinders before they are actually needed. MiniCylPack continues to work until the number of free cylinders defined by MiniCylPackLowCylProd is achieved, or until it can free no more cylinders.

Because MiniCylPack runs proactively, before the system is completely out of free cylinders, requests by foreground tasks for new cylinders can continue to be satisfied, and the work of freeing up new cylinders is

spread evenly across the active transactions as background work. No individual transaction needs to halt and wait for free cylinders, if needed.

MiniCylPack operates in the background as follows:

1. MiniCylPack scans the Master Index, a memory-resident structure with one entry per cylinder, looking for the best candidate set of logically adjacent cylinders where a good amount of free space exists.

To determine the best candidate, MiniCylPack considers such factors as how much free space is currently available on the cylinders, and how much free space would be left after they are packed. MiniCylPack attempts to keep cylinders that were allocated with specific Free Space Percent (FSP) values in compliance with those values.

2. MiniCylPack packs these logically adjacent cylinders to use one cylinder less than is currently being used. For example, MiniCylPack packs four cylinders with an FSP of 0% that are each 75% full into three cylinders that are 100% full.
3. If MiniCylPack cannot free up a permanent storage cylinder by honoring the applicable FSPs, it decreases the FSP uniformly across the set of cylinders, allowing less free space to be preserved on each, until it can free a cylinder, or until it reaches an FSP of 0%.

If MiniCylPack was initiated due to a request for spool cylinders, it only decreases the FSP to a minimum of 10%.

4. The process continues until either:
  - The number of free cylinders reaches the value in MiniCylPackLowCylProd, either because MiniCylPack freed them, or because other data was deleted.
  - A complete pass of all the cylinders in the Master Index was made using an FSP value of 0%, and no additional cylinders were freed.

Over time, MiniCylPack may not be able to keep up with demand, due to insufficient free CPU and I/O bandwidth, or to the increasing cost of freeing up cylinders as the demand for free cylinders continues.

## Related Information

For more information on...	See...
CREATE TABLE and ALTER TABLE	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.
Free Space Percent	<a href="#">FreeSpacePercent</a> , <a href="#">SHOWFSP</a> , <i>Teradata Vantage™ - Database Administration</i> , B035-1093.

# MinPercentCompReduction

Specifies the minimum percentage by which the data block (DB) size must be reduced by compression. If compression will not reduce the DB by at least this amount, the DB will not be compressed, because the overhead of having to compress and uncompress the blocks during DML operations is not worth the minimal space savings that might be achieved.

## Field Group

Compression

## Valid Range

0 through 99 percent

## Default

20 percent

## Changes Take Effect

After the DBS Control Record has been written.

## Usage Notes

BLC enables data compression at the data block (DB) level of the Teradata file system. Compression reduces the amount of storage required for a given amount of data. The BlockLevelCompression field of DBS Control enables and disables BLC.

Compression is also subject to the criteria specified by the [MinDBSectsToCompress](#) DBS Control setting. DBs not meeting these minimum criteria will not be compressed, even if they are larger than the size specified by MinPercentCompReduction.

## Related Information

For more information on...	See...
COMPRESS and UNCOMPRESS commands	<a href="#">Ferret Utility (ferret)</a> .



## MLoadDiscardDupRowUPI

Determines whether the DBS MultiLoad protocol will silently discard rows with UPI violations, or log these rows to the MultiLoad application error table.

### Field Group

General

### Valid Settings

Setting	Effect
TRUE	MultiLoad silently discards rows with UPI violations, and does not log these rows to the application error table.
FALSE	MultiLoad logs rows with UPI violations to the application error table.

### Default

FALSE

### Changes Take Effect

After the DBS Control record has been written.

### Usage Notes

This setting is effective only when IGNORE DUPLICATE ROWS is specified for the MultiLoad DML LABEL command.

The MultiLoad application error table normally receives information about errors detected during the application phase of the MultiLoad import or delete task. It provides information about:

- Uniqueness violations
- Field overflow on columns other than primary index fields
- Constraint errors.

### Related Information

For more information about Teradata MultiLoad, see *Teradata® MultiLoad Reference*, B035-2409.

## MLOADXUtilityLimits

Determines whether MultiLoad Extended Protocol (MLOADX) concurrent task and AWT limits are specific to MLOADX jobs or are shared with other load utilities (FastLoad, MultiLoad, and FastExport).

### Field Group

General

### Valid Settings

Setting	Effect
TRUE	MLOADX concurrent task and AWT limits are determined by the values of DBS Control fields MaxMLOADXTasks and MaxMLOADXAWT, instead of by the values of MaxLoadTasks and MaxLoadAWT.
FALSE	MLOADX shares the concurrent task and AWT limits of FastLoad, MultiLoad, and FastExport, which are determined by the values of DBS Control fields MaxLoadTasks and MaxLoadAWT.

### Default

FALSE

### Changes Take Effect

After the DBS Control record has been written.

### Usage Notes

This field is ignored if the throttles category set in the Teradata Viewpoint Workload Designer portlet is enabled.

### Related Information

For more information on...	See...
Load utilities (MultiLoad, FastLoad, FastExport, and Teradata Parallel Transporter)	<ul style="list-style-type: none"> <li>• <i>Teradata® MultiLoad Reference</i>, B035-2409.</li> <li>• <i>Teradata® FastLoad Reference</i>, B035-2411.</li> <li>• <i>Teradata® FastExport Reference</i>, B035-2410.</li> <li>• <i>Teradata® Parallel Transporter User Guide</i>, B035-2445.</li> </ul>

## MonSesCPUNormalization

Controls whether normalized or non-normalized statistical CPU data is reported in the responses to workload management API calls. API calls that return CPU data include MONITOR SESSION (PM/API), MonitorSession (open API), and MonitorMySessions (open API).

“Coexistent” Teradata systems combine different types of node hardware that might use different types of CPUs running at different speeds. CPU normalization adjusts for these differences when calculating statistics across the system. Although normalized CPU data is used in other areas of the system, such as DBQL and AMPUsage, this field affects only the CPU data reported by PM/API and open API calls.

### Field Group

General

### Valid Settings

Setting	Effect
TRUE	<p>CPU statistical data is normalized.</p> <p>If dynamic workload management is enabled, asynchronous and synchronous exception detection on CPU thresholds are performed using normalized CPU values.</p> <p>Workload management can be enabled from the Viewpoint Workload Designer portlet.</p> <p>There is no measurable overhead associated with setting this field to TRUE.</p>
FALSE	<p>CPU statistical data is not normalized.</p> <p>If dynamic workload management is enabled, asynchronous and synchronous exception detection on CPU thresholds are performed using normalized CPU values.</p> <p>Workload management can be enabled from the Viewpoint Workload Designer portlet.</p>

### Default

FALSE

### Changes Take Effect

After the DBS Control Record has been written.

### Changing the Setting

1. Disable Teradata dynamic workload management software Workload Definition (Category 3) rules, if they are enabled, using the Teradata Viewpoint Workload Designer portlet.  
 Teradata dynamic workload management workloads are affected by MonSesCPUNormalization because they use the MONITOR SESSION PM/API call to obtain CPU information.
2. Adjust any CPU-related exception thresholds associated with the workloads to account for the change in CPU data normalization.
3. Disable session monitoring from Database Window, using the SET SESSION COLLECTION command.

4. Change the MonSesCPUNormalization field in DBS Control.
5. Re-enable session monitoring.
6. Re-enable workload definition rules in the Teradata Viewpoint Workload Designer portlet.

The AMPCPUSec field in the response to workload management API calls contains accumulated CPU seconds for all requests in the session. If the value of MonSesCPUNormalization changes in the middle of a session, AMPCPUSec will no longer be valid during the current session, and will return -1 in record mode, or NULL in indicator mode. All other CPU fields are valid, even if MonSesCPUNormalization changes in the middle of a session.

### **Related Information**

- *Teradata Vantage™ - Application Programming Reference*, B035-1090.
- *Teradata Vantage™ - Database Administration*, B035-1093.

## MPS\_IncludePEOnlyNodes

Excludes PE-only (AMP-less) nodes from MONITOR PHYSICAL SUMMARY Workload Management API statistics calculations.

### Field Group

General

### Valid Settings

Setting	Effect
TRUE	PE-only nodes are included in calculations.
FALSE	PE-only nodes are excluded from calculations.

### Default

FALSE

### Changes Take Effect

After the DBS Control Record has been written.

### Related Information

MONITOR PHYSICAL SUMMARY collects statistical system information which includes average, high, and low CPU and disk usage per node. It can be used to determine node-level system skew.

Some sites use PE-only nodes to aid in balancing the AMP workload. Because PE-only nodes are likely to experience substantially lower CPU and disk usage than nodes running AMPs, they can cause the node statistics to appear as if there is a data skew condition, when no such condition exists. In these cases, excluding PE-only nodes provides statistics that more accurately represent the true system workload conditions.

For more information about Teradata APIs, see *Teradata Vantage™ - Application Programming Reference*, B035-1090.

## NameValidationRule

Determines whether additional rules are used by Vantage to validate object names.

### Field Group

General

### Valid Settings

Setting	Description
0	No additional name validation is performed beyond basic validation rules.
1	Name validation rules used in pre-14.10 releases.
2	Name validation using a minimal character set: 'A' - 'Z', 'a' - 'z', '0' - '9', '#', '\$', and '_'
62	Name validation using the UTF16 client character set.
63	Name validation using the UTF8 client character set.
64	Name validation using the EBCDIC client character set.
65 through 126	Name validation using a user-defined character set.
127	Name validation using the ASCII client character set.

### Default

0

### Changes Take Effect

After the next database restart for new objects. Objects created prior to changing this setting are unaffected by the change.

### Related Information

For more information on...	See...
Basic object naming rules	<i>Teradata Vantage™ - SQL Fundamentals</i> , B035-1141.
User-defined character sets	<i>Teradata Vantage™ - Advanced SQL Engine International Character Set Support</i> , B035-1125.

## NewHashBucketSize

Specifies the number of bits that will be used by the system to identify hash buckets after the next system initialization or reconfiguration. This setting determines how many hash buckets the system can create. A setting of 16 bits gives Vantage 65,536 hash buckets; a setting of 20 bits gives Vantage 1,048,576 hash buckets.

One goal of the Vantage parallel system is to distribute work evenly among the system resources (nodes, virtual processes, and storage). The number of hash buckets an AMP vproc uses is directly related to the amount of work an AMP must do. AMPs with more hash buckets manage more data, and therefore do more work than those with fewer hash buckets.

On many systems, the number of AMPs is not evenly divisible into the number of available hash buckets. Consequently, some AMPs have one more hash bucket than other AMPs. If the number of hash buckets per AMP is relatively high, the imbalance is proportionately low, and the difference in the amount of work the AMPs must do is relatively small.

However, as the number of AMPs is increased on the system, the hash buckets available to each AMP decreases. With fewer hash buckets per AMP, the effect of any imbalance in the number of hash buckets per AMP becomes proportionately greater. This results in the system operating less efficiently.

For example, an AMP using 656 hash buckets must do 1/656 or 0.15% more work than an AMP using 655 hash buckets, but an AMP using only 66 hash buckets must do 1/66 or 1.52% more work than an AMP using 65 hash buckets.

Making more hash buckets available to the system and to each AMP reduces the effects of the imbalance when some AMPs have one more hash bucket than others.

### Field Group

General

### Valid Settings

16 and 20 bits

### Default

Default	Description
20 bits	New installations of Vantage.
16 bits	Systems upgraded from earlier versions of Vantage.

### Changes Take Effect

After the next system initialization or reconfiguration.

### Related Information

See [CurHashBucketSize](#).





## NoDot0Backdown

Commits Vantage to a point release (for example, xx.10), allowing certain release-specific features to be enabled. This prevents the system from being backed down to the corresponding major release (xx.0) without re-initializing the system on the major release.

### Field Group

General

### Valid Settings

Setting	Description
TRUE	Allows certain point-release-specific features to be enabled which will preclude backing down the system to the previous major release of Vantage.  <b>Note:</b> When this setting is TRUE, it cannot be changed to FALSE.
FALSE	Prevents certain point release features from being enabled, but allows the system to be backed down from the currently-installed point release to the corresponding major release, without requiring a system initialization (sysinit).

### Default

On systems that have been initialized on a major release, the default setting is FALSE, and cannot be changed.

On systems that have been upgraded from a major release to a point release without a new system initialization, the default setting is FALSE, and can be changed. After this setting has been changed to TRUE, it cannot be reset to FALSE.

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

Certain new features, if enabled in a point release, would preclude the system from being backed down to the corresponding major release. These feature cannot be enabled unless NoDot0Backdown is explicitly set to TRUE. After NoDot0Backdown has been set to TRUE, it cannot be modified, and the system cannot be backed down without a system re-initialization. For information on the features that require NoDot0Backdown to be TRUE, see the *Teradata Vantage™ - Advanced SQL Engine Release Definition*, B035-1725 for this release of Vantage.

## NumStatisticsCacheSegs

Specifies the number of segments maintained for the Optimizer statistics cache on each PE vproc.

### Field Group

Performance

### Valid Range

2 through 200

### Default

4

### Changes Take Effect

After the next database restart.

### Usage Notes

The default size of each segment is 1024 KB.

The following situations may require a change from the default setting:

- If parsing time increases for tables that have many statistics, increasing the number of cache segments may improve performance.
- Systems with large amounts of memory can dedicate more memory to parsing when the number of cache segments is increased. This is useful for large memory systems.
- If event logging on the statistics cache shows a low hit ratio, increasing the size of the cache can help.

## ObjectUseCountCollectRate

Specifies the amount of time (in minutes) between collections of object use-count data. The system updates the Data Dictionary table DBC.ObjectUsage with this data. This table allows you to find the use count and last access timestamps of any of the following database objects:

- Columns
- Databases
- Indexes
- Macros
- Stored Procedures
- Tables
- Triggers
- UDFs
- Users
- Views

---

### Note:

Object use-count data is not recorded for EXPLAIN, INSERT EXPLAIN, or DUMP EXPLAIN statements, or for Data Dictionary objects.

---

## Field Group

General

## Valid Settings

IF you set this field to...	THEN...
an integer between 1 and 32767	<p>the Data Dictionary is updated with collected object use count data at an interval (in minutes) based on the set value.</p> <p>If you specify a decimal value, DBS Control ignores the fractional part and uses only the integer part. For example, if you specify 12.34, DBS Control uses 12 for the field value and ignores .34.</p> <p><b>Note:</b></p> <p>The recommended minimum value is 10 minutes. Any rate below 10 minutes severely impacts performance.</p>
a value higher than 32767	DBS Control displays a warning message.
a negative value	DBS Control displays an error message.

**Default**

10 minutes

**Changes Take Effect**

After the DBS Control Record has been written.

**Usage Notes**

If you are performing an archive/recovery operation for Data Dictionary tables, the system may disable ObjectUseCountCollectRate if it experiences any contention or locking.

Use the DBQL BEGIN/END QUERY LOGGING statements to control object use count logging on an individual database basis.

**Related Information**

For more information on...	See...
the object use count Data Dictionary columns	<i>Teradata Vantage™ - Data Dictionary</i> , B035-1092.
resetting the object use count Data Dictionary columns	<i>Teradata Vantage™ - Database Administration</i> , B035-1093.
the DBQL BEGIN/END QUERY LOGGING statements	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.

## OverrideARCBLC

Determines whether database tables restored from archives are block-level compressed according to the effective BLOCKCOMPRESSION setting of each table at the time of archiving or according to the system-level BLC defaults at the time the tables are restored.

### Note:

This field does not affect tables with effective BLOCKCOMPRESSION values of ALWAYS or NEVER.

### Field Group

Compression

### Valid Settings

Setting	Description
TRUE	The system default settings for block-level compression are used to determine whether restored tables are block-level compressed.
FALSE	The effective table BLOCKCOMPRESSION value at the time of archiving determines whether restored tables are block-level compressed.

### Default

FALSE

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

This field does not affect tables with effective BLOCKCOMPRESSION values of ALWAYS or NEVER. This includes tables that were created or altered to have their BLOCKCOMPRESSION option:

- set to ALWAYS or NEVER
- set to DEFAULT, and the DefaultTableMode DBS Control field is set to ALWAYS or NEVER
- unspecified, and the DefaultTableMode DBS Control field is set to ALWAYS or NEVER

This field does not affect subtables for which the effective BLOCKCOMPRESSION is set to ALWAYS or NEVER on the target system to which the data is restored.

For permanent tables and global temporary tables, different subtables hold primary data, fallback data, and LOB data. BLC for these categories of data can be independently controlled using DBS Control Compression fields 14 - 21 (Compress...DBs fields). These subtable settings are effective only when the BLOCKCOMPRESSION setting for the corresponding base table is set to MANUAL, either explicitly, or as a result of being set to DEFAULT (or left unset) while the DefaultTableMode DBS Control field is set to MANUAL.

Block compression that is controlled by the BLOCKCOMPRESSION query band takes precedence over the OverrideARCBLC field setting.

BLC enables data compression at the data block (DB) level of the Teradata file system. Compression reduces the amount of storage required for a given amount of data. The BlockLevelCompression field of DBS Control enables and disables BLC.

## Related Information

For More Information on...	See...
Block-level compression	<i>Teradata Vantage™ - Database Design</i> , B035-1094
DBS Control fields that determine effective BLOCKCOMPRESSION value for different types of subtable data	<ul style="list-style-type: none"> <li>• <a href="#">CompressGlobalTempFallbackDBs</a></li> <li>• <a href="#">CompressGlobalTempFallbackCLOBDBs</a></li> <li>• <a href="#">CompressGlobalTempPrimaryDBs</a></li> <li>• <a href="#">CompressGlobalTempPrimaryCLOBDBs</a></li> <li>• <a href="#">CompressMloadWorkDBs</a></li> <li>• <a href="#">CompressPermFallbackDBs</a></li> <li>• <a href="#">CompressPermFallbackCLOBDBs</a></li> <li>• <a href="#">CompressPermPrimaryDBs</a></li> <li>• <a href="#">CompressPermPrimaryCLOBDBs</a></li> <li>• <a href="#">CompressPJDBs</a></li> <li>• <a href="#">CompressSpoolDBs</a></li> <li>• <a href="#">DefaultTableMode</a></li> </ul>
BlockLevelCompression field	<a href="#">BlockLevelCompression</a> .
BLOCKCOMPRESSION table option and BLOCKCOMPRESSION query band	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144

## PermDBAllocUnit

Specifies the size of the storage allocation unit used for data blocks in permanent tables.

### Field Group

File System

### Valid Range

1 to 127 sectors

A sector is 512 bytes.

On cliques with 4 KB aligned devices, the setting is rounded up to the next 8-sector multiple.

### Default

1 sector

### Changes Take Effect

After the DBS Control Record has been written.

### PermDBAllocUnit and System Performance

As tables are modified, rows are added, deleted, and changed. Data blocks grow and shrink dynamically to accommodate their current contents. However, data block sizes can change only in units of PermDBAllocUnit. This means there will nearly always be some unused space left at the end of the data block. If table modifications are relatively even, such incremental changes in data block size result in an average of approximately half an allocation unit of space wasted for every data block. (This is a rough approximation, and will depend on many factors that differ from database to database.)

In environments where new rows are added frequently to tables, or where tables with variable length rows are frequently growing, system performance might be improved slightly by increasing the allocation unit size. With a larger allocation unit, data blocks will not need to be enlarged as frequently, because there will already be room for additional changes. However, in environments where new rows are not added frequently, the additional space in each block can degrade performance by increasing the average I/O size.

Make only small changes to this setting at a time, and carefully evaluate the results before committing the change on a production system. Set the allocation unit to a multiple of the average row size of tables that change frequently, rounded up to the nearest sector.

Because the benefit of larger allocation units is often offset by the consequent increase in average wasted space, Teradata recommends that PermDBAllocUnit be left at the default setting.

### Maximum Multirow Data Block Size

The PermDBAllocUnit and PermDBSize fields together determine the maximum size of multirow data blocks. Because data blocks can grow only in steps (allocation units) defined by PermDBAllocUnit, the size of a data block at any time will always be an integer multiple of PermDBAllocUnit, regardless of the PermDBSize setting.

Consequently, if the PermDBAllocUnit setting is not an integer factor of PermDBSize, then the largest multirow data blocks will be smaller than PermDBSize. For example, if PermDBAllocUnit is set to 4 sectors, even if PermDBSize is set to 255, the largest multirow data blocks can be only 252 sectors (the greatest multiple of 4 that is less than or equal to 255). Similarly, if PermDBAllocUnit is set to 16, the largest multirow data blocks can be only 240 sectors.



## PermDBSize

Specifies the maximum size for multirow data blocks in permanent tables. Rows that are larger than PermDBSize are stored in single-row data blocks, which are not limited by PermDBSize.

### Field Group

File System

### Valid Range

42 through 2047 sectors for systems that use large cylinders

18 through 512 sectors for systems that use small cylinders

A sector is 512 bytes.

On cliques with 4 KB aligned devices, the setting is rounded up to the next 8-sector multiple.

### Default

254 sectors

Setting this field to 0 causes Vantage to use the system default size for this type of data block.

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

When database tables are initially populated, Vantage stores as many rows as possible into each data block, until the block reaches the size specified by the various DB size settings in DBS Control. As tables are subsequently modified, rows can grow such that the existing data blocks would exceed the maximum size. When this happens, the data block is split, and roughly half the rows are moved to a new data block, with the result that the original and new data blocks are each one half of the original size. The result of this type of growth and splitting is that data blocks for heavily modified tables tend to be about 75% of the maximum size defined in DBS Control.

If DBs are compressed, this setting applies to the size of the uncompressed DBs.

### PermDBSize and System Performance

Database performance can be affected by the relationship of data block size to the type of work typically performed by the database:

- When database queries are tactical in nature, involving one or a few table rows, it is advantageous to have fewer rows stored per data block to speed data access. Online transaction processing (OLTP) is an example of this type of work.
- Alternatively, when database queries are strategic in nature, involving complex queries that involve many table rows per table, it is advantageous to have many rows stored in each data block, to

minimize costly data I/O operations. Decision support software (DSS) and complex report generation are examples of this type of work.

PermDBSize sets the default maximum size used by the system for multirow data blocks in permanent tables. Use a larger value if the database is used primarily for strategic work, and a smaller value if the database is used primarily for tactical work.

In a mixed-work environment, determine a value for PermDBSize based on the kind of work typically performed by the database. For tables involved in other types of work, PermDBSize can be overridden on a table-by-table basis using the DATABLOCKSIZE option of the CREATE TABLE and ALTER TABLE SQL statements.

## PMPC\_SessionRateThreshold

Specifies the threshold above which the session cache is refreshed on demand, based on the session rate and last collection time.

---

**Note:**

PMPC\_SessionRateThreshold should be changed only under the direction of Teradata Support Center personnel.

---

**Field Group**

General

**Valid Range**

0 through 600 seconds

**Default**

60 seconds

**Changes Take Effect**

After the DBS Control Record has been written.

**Usage Notes**

This setting is effective only when workloads are enabled from the Viewpoint Workload Designer portlet.

The session cache is refreshed on demand under these conditions:

- Workloads are enabled, and the Workload Exception Interval is greater than PMPC\_SessionRateThreshold
- Workloads are not enabled

If neither of the conditions are met, the session cache is refreshed automatically, once during every exception interval period.

Typically the Exception Interval is less than 60 seconds, so the cache is refreshed automatically.

For best performance and the lowest system overhead, Teradata recommends that the session cache be refreshed automatically.

## PMPC\_TimeoutSecs

Specifies the timeout interval for certain supported PM/API commands.

### Field Group

General

### Valid Range

0 through 3600 seconds

### Default

60 seconds

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

Setting PMPC\_TimeoutSecs to zero means there is no timeout. Vantage waits for internal processing of PM/API requests to complete before responding to the requesting client application.

### Related Information

For more information on PM/API, see *Teradata Vantage™ - Application Programming Reference*, B035-1090.

## PPICacheThrP

Specifies the maximum amount of memory to be used for disk read operations that involve multiple partitions.

---

### Note:

This field affects operations on tables and join indexes with all types of partitioning, including row partitioning, column partitioning, and any combinations of the two. It is not limited to operations on tables and join indexes with partitioned primary indexes (PPIs). It affects partitioned tables and join indexes with primary indexes (PI), primary AMP indexes (PA), and without primary indexes (NoPI).

---

### Field Group

Performance

### Valid Range

0 through 500

The value is in units of 1/10th of a percent, so the valid range represents 0% through 50.0%.

### Default

The default is 10 (1.0%).

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

For some partitioning operations, Vantage processes a subset of the nonempty, noneliminated partitions together, rather than processing one partition at a time. A context is kept for each nonempty partition to be processed. The context defines the current position within the corresponding partition.

The PPICacheThrP value can be used to reduce swapping and avoid running out of memory by limiting the amount of memory used for these contexts and associated data blocks during partitioning operations. However, larger values for PPICacheThrP may improve the performance of these partitioning operations by allowing them to use more memory.

For a multilevel partition, a context is associated with a nonempty combined partition. In the following discussions, partition means combined partition.

PPICacheThrP also specifies the percentage value to use for calculating the number of memory segments that can be allocated to buffer the appending of column partition values to column partitions. The sum of the sizes of these memory segments (minus some overhead) divided by the size of a column partition context determines the number of available column partition contexts. If there are more column partitions in a target table than available column partition contexts, multiple passes over the source rows are required to process a set of column partitions, where the number of column partitions in each set is up to the number

of available column partition contexts. Note that, in this case, there is only one file context open, but each column partition context allocates buffers in memory.

## Partitioning Cache Threshold (PCT)

PCT is the amount of memory to be made available for partitioning operations.

On systems where the file system cache per AMP is less than 100 MB:

On systems where the file system cache per AMP is greater than 100 MB:

---

### Note:

The output of the DBS Control DISPLAY command includes some additional information for PPICacheThrP. This information can help DBAs determine the actual amount of memory available for multiple-context operations on partitioned tables.

---

The value PCT is used in various operations for a partitioned table, as the amount of memory to use for data blocks or AMP buffers associated with the multiple partition contexts, which allow reading from, or writing to, a set of combined partitions at the same time.

The data block size also affects the number of contexts. A smaller data block size allows for more contexts, therefore more partitions can be processed together. If the data block size is 128 KB or less, up to eight contexts may be used, regardless of the PPICacheThrP setting. For larger data block sizes, this may scale down to as low as two contexts.

No more than 256 contexts are ever used, regardless of the PPICacheThrP setting.

## PPICacheThrP Performance Implications

Under most use cases, the default value for PPICacheThrP is adequate, and should not be changed. However, if there are performance issues that might be addressed by adjusting this value, consider the information in this section.

The current data block for the corresponding partition (or buffer, in the case of inserts to column-partitioned tables) is associated with each context. The current set of data blocks or buffers (one for each context) are kept in memory, if possible, to improve the performance of processing the set of partitions at the same time. If there is a shortage of memory, these blocks or buffers may need to be swapped to disk. Excessive swapping, however, can degrade system performance.

Larger values may improve the performance of partitioning operations, as long as the following occur:

- Data blocks or AMP buffers for each context can be kept in memory. When they can no longer be kept in memory and must be swapped to disk, performance may degrade.
- The number of contexts does not exceed the number of nonempty, noneliminated partitions for partitioning operations. (If they do, performance will not improve because each partition can have a context, and additional contexts would be unused.)

In some cases, increasing the value of PPICacheThrP above the default value can provide a performance improvement for individual queries that do these partitioning operations. However, be aware of the potential

for memory contention and running out of memory if too many of these queries are running at the same time.

The default setting of 10 is conservative, and intended to avoid such memory problems. With 80 AMP Worker Tasks (AWTs) per AMP on a system with the default setting of 10, the maximum amount of FSG cache that could be used for these partitioning operations is 80% of FSG cache memory, if all AMPs are simultaneously executing partitioning operations, such as sliding-window joins for 80 separate requests. For configurations that have more than 80 AWTs defined as the maximum, the setting is scaled to the number AWTs. For example, at the default setting of 10, a cap of 80% of FSG cache memory per AMP would still be in effect on such systems.

For many sites, the default may be too conservative. All 80 AWTs might not be running partitioning operations at the same time. If, at most, 60 partitioning operations are expected to occur at the same time, the value of PPICacheThrP could possibly be raised to 15. If at most 40 are expected, the value could possibly be raised to 20, and so on. The best value for this parameter is dependent on the maximum concurrent users expected to be on the system and their workload. No one value is appropriate for all systems.

Also, consider that the number of concurrent partitioning operations, such as sliding-window joins, may increase as partition usage is increased. Increasing the value may increase performance now without memory contention or running out of memory but, in the future, as more partitioning operations run concurrently, performance may decrease, or out of memory situations may occur.

If less than 80 concurrent partitioning operations are expected for your site, and you think that better performance may be possible with an increased value for PPICacheThrP, you can experiment with PPICacheThrP settings to determine an increased PPICacheThrP setting that is optimal for your site and safe for your workloads. Measure pre- and post-change performance and degree of memory contention under expected current and future workloads to evaluate the effects of the change. If increasing the value to one that is reasonably safe for your site does not yield adequate performance for partitioning operations such as sliding-window joins, consider defining partitions with larger granularity, so fewer partitions are involved in the sliding-window join.

## Related Information

For more information on partitioning, including row and column partitioning for tables and join indexes, see *Teradata Vantage™ - Database Design*, B035-1094 and *Teradata Vantage™ - SQL Data Definition Language Syntax and Examples*, B035-1144.

## PrimaryIndexDefault

The CREATE TABLE and CREATE JOIN INDEX statements optionally can include several modifying clauses that determine how a table or join index is created. The PRIMARY INDEX, PRIMARY AMP [INDEX], NO PRIMARY INDEX, PRIMARY KEY, UNIQUE, and PARTITION BY modifiers affect whether the table or join index includes a primary index or primary AMP index. The PrimaryIndexDefault field determines whether a table or join index that is created without any of these modifiers will have a primary index created automatically by Vantage, or will be created as a NoPI table, lacking a primary index.

### Field Group

General

### Valid Settings

Setting	Description
D	Sets or resets this field to the Teradata default. This is the same as specifying P.
P	Vantage automatically creates primary indexes for tables and join indexes created with CREATE TABLE and CREATE JOIN INDEX statements that lack PRIMARY INDEX, PRIMARY AMP [INDEX], NO PRIMARY INDEX, PRIMARY KEY, UNIQUE, and PARTITION BY modifiers. The first column of the table serves as a nonunique primary index (NUPI).  <b>Note:</b> Using the first column may cause significant skew in the distribution of rows to the AMPs if this column is not a good choice as a primary index.
N	Vantage does not create primary indexes for tables and join indexes created without PRIMARY INDEX, PRIMARY AMP [INDEX], NO PRIMARY INDEX, PRIMARY KEY, UNIQUE, and PARTITION BY modifiers. Defaulting to no primary index (NoPI) avoids skew that might occur with using the first column as a primary index.

### Default

D

### Changes Take Effect

After the DBS Control record has been written.

### Usage Notes

If a CREATE TABLE statement specifies a PARTITION BY clause but no PRIMARY INDEX, PRIMARY AMP [INDEX], or NO PRIMARY INDEX clause, the default is NO PRIMARY INDEX, regardless of the PrimaryIndexDefault setting, and regardless of whether the CREATE TABLE includes a PRIMARY KEY or UNIQUE constraint.



The best practice is to explicitly specify PRIMARY INDEX, PRIMARY AMP [INDEX], or NO PRIMARY INDEX in CREATE TABLE statements, rather than relying on the PrimaryIndexDefault field setting.

**Related Information**

For the rules governing how system-defined primary indexes are created, see *Teradata Vantage™ - SQL Data Definition Language Syntax and Examples*, B035-1144.

## ProratedSpaceDistPercent

Specifies the percentage of newly added space that will be divided evenly and distributed to every database and user during a system reconfiguration. New space that is not distributed to databases and users remains owned by database DBC.

---

**Note:**

This field should be changed only under the direction of Teradata Support Center personnel, especially prior to a system reconfiguration.

---

**Field Group**

General

**Valid Range**

0 through 100 percent

**Default**

0

**Changes Take Effect**

After the DBS Control record has been written.

## PRPDMaxSkewedVals

Determines the maximum number of skewed values in a join that should be considered by the optimizer to determine a partial redistribution and partial duplication (PRPD) optimization plan.

A PRPD join strategy helps to minimize the impact of skew on join performance.

### Field Group

Performance

### Valid Range

1 through 500

### Default

50

### Changes Take Effect

After the DBS Control record has been written.

### Related Information

- [PRPDskewPct](#)
- *Teradata Vantage™ - SQL Request and Transaction Processing*, B035-1142

## PRPDSkewPct

Determines the percentage of rows in a join that are used by the optimizer to determine the number of skewed values used for split conditions when partial redistribution and partial duplication (PRPD) optimization is used.

A PRPD join strategy helps to minimize the impact of skew on join performance.

### Field Group

Performance

### Valid Range

1 through 100

### Default

50

### Changes Take Effect

After the DBS Control record has been written.

### Related Information

- [PRPDMaxSkewedVals](#)
- *Teradata Vantage™ - SQL Request and Transaction Processing*, B035-1142

## ReadAhead

Enables or disables read-ahead operations for sequential table access.

### Field Group

Performance

### Valid Settings

Setting	Description
TRUE	Read-ahead is enabled for sequential file access.
FALSE	No read-ahead is issued.

### Default

TRUE

### Changes Take Effect

After the DBS Control Record has been written. Operations in progress at the time of the change are not affected.

### Usage Notes

The File System can issue a read-ahead I/O to bring the next data block, or group of data blocks, into memory whenever a data block is read. Preloading data blocks in advance allows processing to occur concurrently with I/Os, and can improve processing time significantly when tables are being scanned sequentially, for example, when running commands such as SCANDISK.

When ReadAhead is enabled, the number of data blocks that are preloaded is determined by the Read Ahead Count, StandAloneReadAheadCount, and UtilityReadAheadCount fields.

#### Note:

If the Cylinder Read feature is enabled, and there are cylinder slots available, all data blocks on the cylinder are read into memory in one read operation. In this case, individual data blocks are not preloaded. For more information on Cylinder Read, see [Control GDO Editor \(ctl\)](#).

### Related Information

- [Read Ahead Count](#)
- [StandAloneReadAheadCount](#)
- [UseVirtualSysDefault](#)

## Read Ahead Count

Specifies the number of data blocks that will be preloaded into memory in advance of the current data block when performing sequential table scans if the ReadAhead field is set to TRUE.

### Field Group

Performance

### Valid Range

1 through 100 data blocks

### Default

1 data block

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

Increasing the Read Ahead Count field setting can reduce the amount of CPU time spent waiting for read operations to finish. Read Ahead Count should be set high enough to make the typical sequential scan limited only by the CPU, rather than by the read I/O.

The CPU must work harder when data blocks are large or row selection criteria are complex. Consequently, if either of these conditions exist, read ahead counts can be lower.

For example, if a large default data block size is used, most of the tables will consist of large data blocks, and the default ReadAheadCount will suffice. If the default data block size is made very small, most of the tables will consist of small data blocks and system performance might benefit by increasing the ReadAheadCount to 25 or higher.

### Related Information

- [ReadAhead](#)
- [StandAloneReadAheadCount](#)
- [UseVirtualSysDefault](#)

## ReadLockOnly

Enables or disables the special read-or-access lock protocol on the DBC.AccessRights table during privileges validation and on other dictionary tables accessed by read-only queries during request parsing.

### Field Group

Performance

### Valid Settings

Setting	Description
TRUE	Disable the read-or-access lock protocol.
FALSE	Enable the read-or-access lock protocol.

### Default

FALSE

### Changes Take Effect

After the DBS Control record has been written.

### Usage Notes

ReadLockOnly can be set to TRUE for problem isolation troubleshooting if you suspect a concurrency issue is caused by competing locks, such as if a query returns “object does not exist” for an object that you know exists in the database.

## RedistBufSize

Determines the buffer size for AMP-level hashed row redistributions, as used by load utilities (MultiLoad and FastLoad), and archive/recovery operations. This field also determines the size of the buffers used to redistribute USI rows when creating an index on a populated table with the CREATE UNIQUE INDEX SQL statement.

---

### Note:

This field should be changed only under the direction of Teradata Support Center personnel.

---

### Field Group

Performance

### Valid Settings

- -1 specifies the optimal buffer size to avoid extra memory overhead for sending row redistribution messages on the current system.
- 0 specifies the default buffer size for the current system.
- 1 through 63 specifies the buffer size in units of kilobytes. For settings within this value range, the actual buffer size may be adjusted internally by Vantage for better memory utilization.
- 512 through 65024 specifies the buffer size in units of bytes. This is equivalent to 0.5 through 63.5 kilobytes. For settings within this value range, the buffer size is fixed, and used exactly as is, without any adjustment by Vantage.

### Default

The default is about 3,832 bytes, to optimize message efficiency.

To see the exact default value for RedistBufSize, use the 'help perf' DBS Control command and read the online information for the RedistBufSize field. This information is updated dynamically to reflect the most current default value.

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

For the redistribution of data from AMP to AMP, the system reduces message overhead by grouping individual rows before sending them on to their destination AMPs. The rows are grouped into buffers on the originating AMP, one for each destination AMP.

When all the rows to be sent to a particular AMP have been collected in the corresponding buffer, they are sent to their destination AMP with a single message.

If there are  $N$  AMPs in the system, then each AMP has  $N$  buffers for managing redistribution data, making a total of  $N^2$  buffers in the system used per redistribution. Multiplying the number of redistribution buffers



per node by the value of RedistBufSize gives the total amount of system memory that will be used on each node for each redistribution. For example, each FastLoad or MultiLoad job that is importing data at a given time requires a separate redistribution.

### **Example: The RedistBufSize DBS Control field and redistributions**

Assume a system has 12 nodes, with 8 AMPs per node.

The system would have a total of  $12 \times 8 = 96$  AMPs.

Therefore, each AMP would need to use 96 buffers for each redistribution.

To calculate the amount of memory per node used for each AMP-level redistribution, first multiply the number of buffers per AMP by the number of AMPs per node:

$96 \times 8 = 768$  redistribution buffers per node per redistribution

Then multiply the number of redistribution buffers by the RedistBufSize value:

$768 \times 4 \text{ KB} = 3,072 \text{ KB}$  or 3 MB per node used for each AMP-level redistribution

The memory used for redistributions scales with system size. Adding more nodes or more AMPs to a system necessitates more memory for redistributions.

### **RedistBufferSize Performance Implications**

If a system has relatively few AMPs, a larger redistribution buffer size usually has a positive effect on load performance. However, on larger systems with many AMPs, a large buffer size can consume excessive memory, especially if many load jobs are run concurrently.

For more information on row redistribution and performance, see *Teradata Vantage™ - Database Administration*, B035-1093.

## RedriveDefaultParticipation

Determines whether or not the Redrive feature is used for all sessions by default. This field is effective only when the RedriveProtection field is set to enable Redrive.

### Field Group

General

### Valid Settings

Setting	Description
0	Redrive participation is disabled by default for all sessions. Automatic request resubmission will not occur for all sessions. You can enable Redrive for an individual session using logon parameters or query bands.
1	Redrive participation is enabled by default for all sessions. All sessions within the system will have their uncompleted requests automatically resubmitted after a database restart. You can disable Redrive for an individual session using logon parameters or query bands.

### Default

0

### Changes Take Effect

After the next Teradata system restart.

### Usage Notes

The Redrive feature can be enabled and disabled at the session level using the REDRIVE query band and by means of the TDWM.redrive table, which can determine if redrive is used at the user, account, or profile level.

### Related Information

For more information on...	See...
Query bands	<i>Teradata Vantage™ - SQL Data Definition Language Detailed Topics</i> , B035-1184.
TDWM.redrive table	<i>Teradata Vantage™ - Database Administration</i> , B035-1093.

## **RepCacheSegSize [Deprecated]**

Do not use.

## RedriveProtection

Determines system-level support for the Redrive feature. After a system restart, the Redrive feature preserves responses for completed requests and resubmits uncompleted requests.

### Field Group

General

### Valid Settings

Setting	Description
0	Redrive is disabled. Automatic request resubmission cannot occur.
1	Memory-based automatic request resubmission is enabled. SQL request text is maintained in memory on the requesting PE node. The request is automatically resubmitted after a database restart. Completed responses are stored in persistent spool tables.

### Default

0 on systems that are upgraded from Teradata Database releases prior to 14.10.

1 on systems that are initialized on Teradata Database 14.10 or later.

### Changes Take Effect

After the next Teradata system restart.

### Usage Notes

To disable Redrive, you must disable logons from the Supervisor window of Database Window (DBW) or similar console subsystem interface, such as cnstern. Ensure that no sessions are logged on before setting RedriveProtection to 0.

The Redrive feature can be enabled and disabled at the session level using the REDRIVE query band and by means of the TDWM.redrive table, which can determine if redrive is used at the user, account, or profile level.

### Related Information

For more information on...	See...
DISABLE LOGONS command in DBW	<a href="#">DISABLE LOGONS/ DISABLE ALL LOGONS.</a>
Query bands	<i>Teradata Vantage™ - SQL Data Definition Language Detailed Topics</i> , B035-1184.
TDWM.redrive table	<i>Teradata Vantage™ - Database Administration</i> , B035-1093.



## RequestDuration

Controls how long Vantage waits for the completion of a request to an external file system after the request timeout or connection timeout has been exceeded and request retries have been commenced. This setting is effective only when the Native Object Store (NOS) feature is enabled.

### Field Group

Native Object Store

### Valid Settings

2 to 1200 seconds

### Default

300 seconds

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

*Foreign tables* allow Vantage to access data from external, cloud-based data storage, such as AWS S3, without requiring you to manually move the data into the database first from where it natively resides. Foreign tables are identified by a hostname, path, and other metadata that point to the external storage. Vantage can read and process semi-structured or unstructured external data in foreign tables using standard SQL. For example, you can use Teradata analytic functions to examine the data, join it to the relational data in the database, and issue queries against it as you can for other data in Vantage.

### Related Information

The following fields affect the NOS feature and foreign table access.

- [EnableNOS](#)
- [ConnectionBufferSize](#)
- [ConnectionTimeout](#)
- [DefaultRowGroupSize](#)
- [RequestTimeout](#)
- [UserRetryLimit](#)

## RequestTimeout

Controls how long Teradata Database tolerates a slow network connection to an external file system before aborting the request, re-establishing the connection, and issuing the request again. This setting is effective only when the Native Object Store (NOS) feature is enabled.

### Field Group

Native Object Store

### Valid Settings

1 to 600 seconds

### Default

60 seconds

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

*Foreign tables* allow Vantage to access data from external, cloud-based data storage, such as AWS S3, without requiring you to manually move the data into the database first from where it natively resides. Foreign tables are identified by a hostname, path, and other metadata that point to the external storage. Vantage can read and process semi-structured or unstructured external data in foreign tables using standard SQL. For example, you can use Teradata analytic functions to examine the data, join it to the relational data in the database, and issue queries against it as you can for other data in Vantage.

### Related Information

The following fields affect the NOS feature and foreign table access.

- [EnableNOS](#)
- [ConnectionBufferSize](#)
- [ConnectionTimeout](#)
- [DefaultRowGroupSize](#)
- [RequestDuration](#)
- [UserRetryLimit](#)

## RevertJoinPlanning

Determines whether the Teradata query Optimizer uses newer or older join planning techniques.

### Field Group

Performance

### Valid Settings

Setting	Description
TRUE	Causes the Teradata query Optimizer to use older join planning logic, and limits the maximum number of tables that can be used for join planning to 64. To restrict all users to no more than 64 tables joined per query block, set RevertJoinPlanning in DBS Control to TRUE.
FALSE	Causes the Teradata query Optimizer to use new join planning logic.

### Default

FALSE

### Changes Take Effect

After the DBS Control record has been written.

### Related Information

For more information on...	See...
the maximum number of tables that can be joined	<a href="#">MaxJoinTables</a> .
join planning and optimization	<i>Teradata Vantage™ - SQL Request and Transaction Processing</i> , B035-1142.



## RollbackPriority

Determines the priority given to rollback operations.

### Field Group

General

### Valid Settings

Setting	Description
TRUE	Rollbacks are executed within the aborted job's workload.
FALSE	Rollback is at system priority, a super-high priority, greater than any user-assigned priority. In the Viewpoint Workload Designer portlet, system priority is represented as TDWM WD 0. Within the SLES 11 and SLES 12 priority schedule hierarchy, the rollback will be running within the Default Virtual Partition, which is reserved for critical internal work. It will appear as internal workload 254.

### Default

FALSE

This value should be changed only after careful consideration of the consequences to system performance, based on the information here and in *Teradata Vantage™ - Database Administration*, B035-1093.

### Changes Take Effect

After the next database restart.

### Usage Notes

RollbackPriority affects only individual rollbacks resulting from aborted sessions. It does not affect rollbacks resulting from a database restart. The priority of a rollback resulting from a restart is determined by a setting in the Recovery Manager (rcvmanager) utility. This setting can be changed at any time, even if there is no system rollback underway. If a system rollback is underway when the setting is changed, the tasks working on the rollback will have their priorities changed immediately.

### RollbackPriority Performance Implications

Because rollbacks can involve millions or billions of rows, competing for CPU and other system resources, rollbacks can impact system performance. Rollbacks can keep locks on affected tables for hours or days until the rollback is complete. During a rollback, a trade-off occurs between overall system performance and table availability.

How RollbackPriority affects performance is not always straightforward. It is related to the TASM/Viewpoint Ruleset, job mix, and other processing dynamics. The RollbackPriority setting should only be changed after full consideration of the performance consequences:

- When RollbackPriority is set to FALSE, rollbacks are performed at system priority, a special priority higher than any user-assigned priority, that is reserved for critical internal work. As a result, faster rollbacks occur at the expense of other online performance.

The default setting of FALSE is especially appropriate when rollbacks are large, occurring to critical tables that are accessed by many users. It is better to complete these rollbacks as quickly as possible to maximize table availability.

- When RollbackPriority is set to TRUE, rollbacks are executed within the aborted job's workload. This isolates the rollback processing to the aborted job's priority, and minimizes the effect on the performance of the rest of the system. However, if the rollback places locks on tables that other users are waiting for, this causes a greater performance impact for those users, especially if the rollback is running at a low priority.

A setting of TRUE is appropriate when rollbacks are typically smaller, occurring to smaller tables that are less critical, and less extensively used.

Rollbacks can be affected by Workload Management Capacity On Demand (WM COD) and Hard Limits, depending on their running context:

- When RollbackPriority is FALSE, the rollback runs under the system priority, which is subject to WM COD throttling.
- When RollbackPriority is TRUE, the rollback runs under the current user workload, which is also subject to WM COD throttling. The current workload may be further throttled if it is running under a Virtual Partition with a fixed limit, or a SLG Tier Workload Management Method with a hard limit.

## Related Information

For more information on...	See...
rollbacks, rollback priority, and their affect on performance	<i>Teradata Vantage™ - Database Administration</i> , B035-1093.
the Recovery Manager utility	<a href="#">Recovery Manager (rcvmanager)</a> .
workloads and the Workload Designer portlet for Viewpoint	<i>Teradata® Viewpoint User Guide</i> , B035-2206.

## RollForwardLock

Defines the system default for the RollForward using Row Hash Locks option. This allows the database administrator to specify that row hash locks should be used to lock the target table rows during a RollForward. Row hash locks reduce lock conflicts, so that users are more likely to be able to access data during the RollForward operation.

### Field Group

General

### Valid Settings

Setting	Description
TRUE	Enables this feature.
FALSE	Disables this feature.

### Default

FALSE

### Changes Take Effect

After the next database restart.

# RoundHalfwayMagUp

Determines how rounding is performed when computing values of DECIMAL type. A halfway value is exactly halfway between representable decimal values.

## Field Group

General

## Valid Settings

The rounding behavior is different depending upon the setting of the RoundHalfwayMagUp field.

IF you set the field to...	THEN the Teradata system uses the rounding semantics...
TRUE	appropriate for many business applications: The magnitudes of halfway values are rounded up. Halfway values are rounded away from zero so that positive halfway values are rounded up and negative halfway values are rounded down (toward negative infinity). For example, a value of 2.5 is rounded to 3.
FALSE	traditional for Vantage: A halfway value is rounded up or down so that the least significant digit is even. For example, a value of 2.5 is rounded to 2.

## Default

FALSE

## Changes Take Effect

After the next database restart.

## RoundNumberAsDec

Determines how rounding is performed when computing values of NUMBER type.

### Field Group

General

### Valid Settings

Setting	Description
TRUE	Rounding for NUMBER data types matches rounding for DECIMAL data types as determined by the setting of RoundHalfwayMagUp.
FALSE	Rounding for NUMBER data types always rounds the number away from zero.

### Default

FALSE

### Changes Take Effect

After the next database restart

## SessionMode

Defines the Teradata system default transaction mode, case sensitivity, and character truncation rule for a session.

### Field Group

General

### Valid Settings

The setting...	Defaults SQL sessions to...
0	Vantage transaction semantics, case insensitive data, and no error reporting on truncation of character data.
1	ANSI transaction semantics, case sensitive data, and error reporting on truncation of character data.

### Default

0. The default can be overridden at the user or session level (at logon).

### Changes Take Effect

After the next database restart.

## ShowAllUserNames

Determines whether user names are shown in Vantage logs, including user names that have not been authenticated.

### Field Group

General

### Valid Settings

Setting	Description
TRUE	Show all user names in logs, regardless of whether user names are authenticated.
FALSE	Conceal any unauthenticated user name in logs as "Non-existent User".

### Default

FALSE

### Changes Take Effect

After the DBS Control record has been written.

### Usage Notes

Affected logs are:

- DBC.Eventlog table
- Views based on DBC.Eventlog, such as DBC.LogOnOffV
- Operating system log, for example, /var/log/messages

### Note:

If you set ShowAllUserNames to TRUE, all user names, including those that cannot be authenticated, are logged and visible in the affected logs. This can be a security risk, because a common user error is to enter a password in the user name logon field, which would fail authentication. If ShowAllUserNames is TRUE, these misplaced passwords would be visible in the affected logs.

The default setting of FALSE prevents these unauthenticated user names from being visible in the logs by recording them as Non-existent User.

Change this field value to TRUE only if you determine that the utility of seeing unauthenticated user names in the affected logs overrides the risk of exposing potentially misplaced passwords.

Related Information

For More Information on...	See...
Gateway logging	<a href="#">Gateway Control (gtwcontrol)</a> and <a href="#">Gateway Global (gtwglobal)</a> .
User authentication and security	<i>Teradata Vantage™ - Advanced SQL Engine Security Administration</i> , B035-1100.



## SHOW IN XML Formatted Output

Controls the formatting of information returned by the SHOW statement when the IN XML option is used.

### Field Group

General

### Valid Settings

Setting	Description
TRUE	The output XML document is pretty-printed with indentation.
FALSE	The output XML document is an unbroken string of text without indentation.

### Default

FALSE

### Changes Take Effect

After the DBS Control Record has been written.

## SHOW IN XML Memory Limit

Sets the maximum amount of memory in kilobytes used for internal processing of SHOW IN XML statements. If the memory required to process a request exceeds the value of SHOW IN XML Memory Limit, the system returns an error.

### Field Group

Performance

### Valid Range

8 through 128 MB

### Default

64 MB

### Changes Take Effect

After the DBS Control Record has been written.

## SkewAllowance

Makes allowance for data skew in the build relation. SkewAllowance specifies a percentage factor used by the Optimizer in choosing the size of each hash join partition.

SkewAllowance reduces the memory size for the hash join specified by HTMemAlloc. This allows the Optimizer to take into account a potential skew of the data that could make the hash join run slower than a merge join.

### Field Group

Performance

### Valid Range

20 though 80%

### Default

75%

This is the recommended setting.

### Changes Take Effect

After the DBS Control Record has been written.

### Related Information

See [HTMemAlloc](#).

## SmallDepotCylsPerPdisk

Determines the number of Depot cylinders the file system allocates per pdisk (storage device) to contain small slots (1 MB). A small slot can hold a single data block during depot operations.

The actual number of small-depot cylinders used per AMP is this value multiplied by the number of pdisks per AMP.

### Field Group

File System

### Valid Range

1 through 10 cylinders

### Default

2

### Changes Take Effect

After the next database restart.

### Usage Notes

The Depot is a set of transitional storage locations (a number of cylinders) used by the file system for performing in-place writes of DBs or WAL DBs (WDBs). An in-place write means that the changed DB is written back to exactly the same place on disk from which it was originally read. In-place writes are only performed for modifications to DBs that do not change the size of table rows, and therefore do not require any reallocation of space.

Writing the changed DB directly back to its original disk location could leave the data vulnerable to various hardware and system problems that can occur during system resets, such as a disk controller malfunctions or power failures. If such a problem occurred during the write operation, the data could be irretrievably lost.

The Depot protects against such data loss by allowing the file system to perform disk writes in two stages. First the changed DB (or WDB) is written to the Depot. After the data has been completely written to the Depot, it is written to its permanent location on the disk. If there is a problem while the data is being written to the Depot, the original data is still safe in its permanent disk location. If there is a problem while the data is being written to its permanent location, the changed data is still safe in the Depot.

During database startup, the Depot is examined to determine if any of the DBs or WDBs should be rewritten from the Depot to their permanent disk locations.

## SnapBypassAggrCache

Specifies whether snapshot dumps are suppressed if resource exhaustion is detected when reserving memory for aggregate processing.

### Field Group

General

### Valid Settings

Setting	Description
TRUE	Snapshot dumps are suppressed. In this case, although snapshot dumps are not created when resources are exhausted, insufficient memory errors are still returned.
FALSE	Snapshot dumps are created when memory resources are exhausted.

### Default

TRUE

### Changes Take Effect

After the next database restart.

### Usage Notes

SnapBypassAggrCache can be used when submitting transactions that involve many aggregate functions, which require much cache processing. In these situations, memory resource exhaustion would normally cause snapshot dumps, slowing the system down. Use SnapBypassAggrCache to turn off snapshot dumps temporarily. If memory resources become exhausted, the transaction will be aborted with an error message, but the system will not record a snapshot dump.

To avoid masking real memory leak issues, reset SnapBypassAggrCache to FALSE when the memory intensive query has completed. Snapshot dumps are useful for tracing the cause of real memory problems.

## Spill File Path

Specifies a directory that the Relay Services Gateway (RSG) can use for spill files.

### Field Group

General

### Valid Setting

Any existing path

### Default

/opt/teradata/tdat/temp/tdrsg

### Changes Take Effect

After the next database restart.

## SpoolBigDBAllocUnit

Determines the storage allocation unit size in sectors for multirow big data blocks in spool tables.

### Field Group

File System

### Valid Range

8 to 127 sectors

A sector is 512 bytes.

On cliques with 4 KB aligned devices, the setting is rounded up to the next 8-sector multiple.

### Default

1 sector

### Changes Take Effect

After the DBS Control Record has been written.

### For More Information

Changes to this field may be of less consequence than changes to the analogous PermDBAllocUnit field because spool tables are not updated frequently. Nevertheless, care should be taken when considering a change to this setting. Before making changes to this field, refer to the information for [PermDBAllocUnit](#).

## StandAloneReadAheadCount

Specifies the number of data blocks beyond the current block that will be preloaded into memory during the following sequential scan operations:

- File system startup
- File system utility (for example, Ferret) operations, such as SCANDISK, that occur when Vantage is not running

---

### Note:

To enable the read-ahead feature, the ReadAhead field must be set to TRUE.

---

### Field Group

Performance

### Valid Range

1 through 100 blocks

### Default

20 blocks

### Changes Take Effect

After the DBS Control record has been written.

### Usage Notes

The number of data blocks to be preloaded during file system utility operations when Vantage is running is determined by the UtilityReadAheadCount setting.

### Related Information

- [ReadAhead](#)
- [Read Ahead Count](#)
- [UseVirtualSysDefault](#)



## StepsSegmentSize

Defines the maximum size (in KB) of the plastic steps segment (also known as OptSeg).

### Field Group

Performance

### Valid Range

64 through 1024 KB

### Default

1024 KB

### Changes Take Effect

After the DBS Control Record has been written. Any plastic step generation operations in progress at the time of the change are not affected.

### Usage Notes

When decomposing a Teradata SQL statement, the parser generates plastic steps, which the AMPs then process.

Large values allow the parser to generate more SQL optimizer steps that the AMPs use to process more complex queries.

Set this field to a small number to limit the query complexity.

## SyncScanCacheThr

Determines the amount of FSG cache that can be used for synchronized table scans.

SyncScanCacheThr is effective only when DisableSyncScan is set to FALSE.

---

### Note:

This setting should be changed only under the direction of Teradata Services personnel.

---

### Field Group

Performance

### Valid Range

0 through 100%

When this field is set to zero, Vantage uses the default value for the field.

### Default

10%.

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

The synchronized full-table scan feature of Teradata allows several table scanning tasks to simultaneously access the portion of a large table that is currently in the cache. Synchronized table scans happen only when full-table scans access similar areas of a large table. Synchronized table scans can improve database performance by reducing the required disk I/O. Synchronized table scans are available only to large tables undergoing full-table scans.

Full table scans of large tables can fill up the cache quickly, flushing data from smaller reference tables out of the cache too soon, or preventing such data from being cached at all. Because it is unlikely that the cached data from these large tables will be accessed again before the data has been replaced in the cache, the benefits of caching are normally not realized for full table scans of large tables. Therefore, large tables are normally excluded from the cache. The DBSCacheThr setting demarcates "small" from "large" tables for purposes of most system caching decisions.

However, if several tasks are scanning the same large table, efficiencies can be realized by caching a portions of the table, and allowing several scans to access the cached data simultaneously. When synchronized scanning is enabled, a portion of the cache can be used for synchronized scanning of large tables that might otherwise be excluded from the cache. Vantage determines which tables qualify for potential synchronized scanning.

The relative benefits of synchronized large table scans versus allowing more of the cache to be used for small reference tables depends on the specific mix of work types on the system, and may change with

time. Cache in use for synchronized scans is not available for caching frequently accessed data from small reference tables. Therefore, Teradata recommends changing SyncScanCacheThr only if you are directed to do so by Teradata Services personnel. Make only small changes to the field, and carefully observe the effects on system performance before committing the change on a production system.

**Related Information**

- [DBSCacheThr](#)
- [DisableSyncScan](#)

## SysChangeThresholdOption

Specifies whether Vantage should automatically determine an appropriate data change threshold for statistics recollection.

This field is used only if the [DefaultUserChangeThreshold](#) setting is disabled (set to zero) when a COLLECT STATISTICS SQL statement does not specify the USING THRESHOLD PERCENT option.

### Field Group

Optimizer Statistics

### Settings

Value	Description
0	Use Teradata default, currently equivalent to setting 1.
1	Enables system-determined change threshold checking in statistics collection when DBQL ObjectUsage (update/delete/insert counts) is enabled.
2	Enables system-determined change threshold checking in statistics collection regardless of ObjectUsage setting in DBQL logging options.
3	Disables system-determined change threshold.

### Default

0

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

Statistics are used by the optimizer to improve query performance.

### Related Information

For more information on...	See...
Optimizer statistics fields	<ul style="list-style-type: none"> <li>• <a href="#">DefaultTimeThreshold</a></li> <li>• <a href="#">DefaultUserChangeThreshold</a></li> <li>• <a href="#">SysSampleOption</a></li> </ul>

## SysInit

Indicates whether the system has been initialized properly.

---

**Note:**

This field is read only. It is set during system initialization. System initialization (sysinit) destroys all user and dictionary data, and reinitializes the system.

---

### Field Group

General

### Settings

Value	Description
TRUE (time unknown)	Vantage was upgraded without being reinitialized.
TRUE ( <i>timestamp</i> )	Vantage was upgraded with a successful system initialization at the time specified by <i>timestamp</i> .
FALSE	Vantage was upgraded with an unsuccessful system reinitialization.

### Usage Notes

This field must be TRUE for Teradata system startup to begin.

## SysInitRelease

Specifies the major release number on which the system was last initialized.

---

**Note:**

This field is automatically updated during system initialization, and cannot be modified using DBS Control. A system initialization (sysinit) destroys all user and dictionary data, and reinitializes the system.

---

### Field Group

General

### Usage Notes

A setting of zero indicates that the system was last initialized on a major release prior to Teradata Database release 13.10.

## SysSampleOption

Specifies whether Vantage should determine the appropriate amount of table data to sample for statistics collection. If not, the sample size will be 100%.

This field is used only when a COLLECT STATISTICS SQL statement does not specify the USING SAMPLE option.

### Field Group

Optimizer Statistics

### Valid Settings

Setting	Description
0	Use Teradata default, currently equivalent to setting 1.
1	Enables system-determined sample percentage up to 100%.
2	Sample percentage is 100% (full statistics).

### Default

0

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

Statistics are used by the optimizer to improve query performance.

### Related Information

For more information on...	See...
Optimizer statistics fields	<ul style="list-style-type: none"> <li>• <a href="#">DefaultTimeThreshold</a></li> <li>• <a href="#">DefaultUserChangeThreshold</a></li> <li>• <a href="#">SysChangeThresholdOption</a></li> </ul>

## System TimeZone Hour

Defines the System Time Zone Hour offset from Universal Coordinated Time (UTC).

### Field Group

General

### Valid Range

-12 through 14

### Default

0

### Changes Take Effect

For new sessions begun after the DBS Control Record has been written. Existing sessions are not affected.

### Usage Notes

If a time zone string and rule is defined in an SDF file, and processed using the `tdlocaledef` utility, that value takes precedence over the setting of System TimeZone Hour.

If the TimeZoneString entry in an SDF file is not used, for locations that observe Daylight Savings Time (DST), the System TimeZone Hour value must be updated to adjust the offset used for system time when DST begins and ends.

Like system time, session time zones defined for individual users (with `CREATE USER` or `MODIFY USER`) or set for individual sessions (with `SET TIME ZONE`) can use time zone strings to ensure that the session time automatically adjusts for DST.

---

#### Note:

If the `CURRENT_TIME` or `CURRENT_TIMESTAMP` built-in functions are used in stored procedures, those procedures must be recompiled whenever System TimeZone Hour is changed. Recompilation of stored procedures is not necessary if a time zone string is set using the `tdlocaledef` utility.

---

### Related Information

For more information on...	See...
Built-in time and date functions	<i>Teradata Vantage™ - SQL Date and Time Functions and Expressions</i> , B035-1211.
Setting system time zone minute offset	<a href="#">System TimeZone Minute</a>
Setting session time zones	Information about <code>SET TIME ZONE</code> , <code>CREATE USER</code> , and <code>MODIFY USER</code> in <i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.



For more information on...	See...
Tdlocaledef utility	<a href="#">Teradata Locale Definition Utility (tdlocaledef)</a>

## System TimeZone Minute

Defines the System Time Zone Minute offset from Universal Coordinated Time (UTC).

### Field Group

General

### Valid Range

-59 through 59

### Default

0

### Changes Take Effect

After the DBS Control Record has been written. Existing sessions are not affected.

### Usage Notes

If the `CURRENT_TIME` or `CURRENT_TIMESTAMP` built-in functions are used in stored procedures, those procedures must be recompiled whenever System TimeZone Minute is changed. Recompilation of stored procedures is not necessary if a time zone string is set using the `tdlocaledef` utility.

### Related Information

For more information on...	See...
built-in time and date functions	<i>Teradata Vantage™ - SQL Functions, Expressions, and Predicates</i> , B035-1145.
Setting system time zone hour offset	<a href="#">System TimeZone Hour</a>
Setting session time zones	Information about SET TIME ZONE, CREATE USER, and MODIFY USER in <i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.
Tdlocaledef utility	<a href="#">Teradata Locale Definition Utility (tdlocaledef)</a>

## System TimeZone String

Displays the currently set system time zone string. This value cannot be changed using DBS Control.

### Field Group

General

### Usage Notes

System TimeZone String reflects the current time zone string setting. A system time zone string may be set using the `tdlocaledef` utility. If no system time zone string has been specified, System TimeZone String reports “Not Set.”

### Related Information

For more information on...	See...
tdlocaledef and SDF files	<a href="#">Teradata Locale Definition Utility (tdlocaledef)</a> .
system time zone strings	<i>Teradata Vantage™ - SQL Date and Time Functions and Expressions</i> , B035-1211.

## Target Level Emulation

---

**Note:**

This field is for the use of Teradata Support Center personnel and Teradata field engineers. It should not be enabled on a production system.

---

The Target Level Emulation field allows field test engineers to change Optimizer costing parameters used to choose the most cost-efficient path to satisfy a query. These parameters are system- and release-specific.

**Field Group**

General

**Valid Settings**

Setting	Description
TRUE	Enable target-level emulation.
FALSE	Disable target-level emulation.

**Default**

FALSE

**Changes Take Effect**

After the DBS Control Record has been written.

**Related Information**

For more information, see *Teradata Vantage™ - SQL Request and Transaction Processing*, B035-1142.

## TempBLCInterval

Specifies the amount of time in minutes that the background temperature-based block-level compression (TBBLC) task waits after compressing or uncompressing data before checking for other data to process. The field is effective only when TBBLC is enabled.

### Field Group

Compression

### Valid Range

1 through 120 minutes

### Default

10 minutes

### Changes Take Effect

The first time the background compression task runs after the DBS Control Record has been written.

### Usage Notes

Once all cylinders of a Master Index AMP have been processed during a given scan cycle, temperature-based block-level compression (TBBLC) sleeps until TempBLCRescanPeriod triggers the start of a new scan cycle or a database restart triggers a new scan cycle.

Data temperatures refers to frequency of access:

- Hot data is frequently accessed.
- Cold data is infrequently accessed.
- Warm data is accessed moderately frequently.

Data temperature is measured separately for each AMP at a cylinder level and is a relative measure because the temperature of cylinders can change even if the data it is not directly accessed. This is because it depends on how often other cylinders are accessed in comparison. TBBLC automatically compresses cold data to save storage space and decompresses Warm or Hot data if it was previously Cold (and thus compressed). At any time, different portions of a table, which are spread among many cylinders, may exist in different states of compression depending on their temperatures.

BLC enables data compression at the data block (DB) level of the file system. Compression reduces the amount of storage required for a given amount of data. The BlockLevelCompression field of DBS Control enables and disables BLC.

### Related Information

For more information on...	See...
Block-Level Compression	<i>Teradata Vantage™ - Database Design</i> , B035-1094.

For more information on...	See...
BLOCKCOMPRESSION table attribute	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.
Enabling TBBLC	<a href="#">EnableTempBLC</a> .
Ferret COMPRESS, UNCOMPRESS, and FORCE commands	<a href="#">Ferret Utility (ferret)</a> .
Query bands for compression	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.
Storage group of DBS Control fields	<a href="#">Storage Group</a> .

## TempBLCIOThresh

Specifies the maximum threshold number of outstanding I/O operations that can exist on a node above which the background temperature-based block-level compression (TBBLC) task pauses for a short period before running again. If the system remains busy, the process waits according to the interval defined by [TempBLCInterval](#).

This field is effective only when TBBLC is enabled.

### Field Group

Compression

### Valid Range

1 through 1,000

### Default

1

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

This field limits the impact of TBBLC on foreground workloads.

Data temperatures refers to frequency of access:

- Hot data is frequently accessed.
- Cold data is infrequently accessed.
- Warm data is accessed moderately frequently.

Data temperature is measured separately for each AMP at a cylinder level and is a relative measure because the temperature of cylinders can change even if the data it is not directly accessed. This is because it depends on how often other cylinders are accessed in comparison. TBBLC automatically compresses cold data to save storage space and decompresses Warm or Hot data if it was previously Cold (and thus compressed). At any time, different portions of a table, which are spread among many cylinders, may exist in different states of compression depending on their temperatures.

BLC enables data compression at the data block (DB) level of the file system. Compression reduces the amount of storage required for a given amount of data. The BlockLevelCompression field of DBS Control enables and disables BLC.

### Related Information

For more information on...	See...
Block-Level Compression	<i>Teradata Vantage™ - Database Design</i> , B035-1094.

For more information on...	See...
BLOCKCOMPRESSION table attribute	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.
Enabling TBBLC	<a href="#">EnableTempBLC</a> .
Ferret COMPRESS, UNCOMPRESS, and FORCE commands	<a href="#">Ferret Utility (ferret)</a> .
Query bands for compression	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.
Storage group of DBS Control fields	<a href="#">Storage Group</a> .



## TempBLCPriority

Specifies the priority at which the temperature-based block-level compression (TBBLC) background compression task runs. This setting is effective only when TBBLC is enabled.

### Field Group

Compression

### Valid Settings

Setting	Description
LOW	Lowest priority
MEDIUM	Medium priority
DEFAULT	Same as MEDIUM
HIGH	High priority
RUSH	Highest priority
TOP	Same as RUSH

### Default

MEDIUM

### Changes Take Effect

The first time the background compression task runs after the DBS Control Record has been written.

### Usage Notes

Data temperatures refers to frequency of access:

- Hot data is frequently accessed.
- Cold data is infrequently accessed.
- Warm data is accessed moderately frequently.

Data temperature is measured separately for each AMP at a cylinder level and is a relative measure because the temperature of cylinders can change even if the data it is not directly accessed. This is because it depends on how often other cylinders are accessed in comparison. TBBLC automatically compresses cold data to save storage space and decompresses Warm or Hot data if it was previously Cold (and thus compressed). At any time, different portions of a table, which are spread among many cylinders, may exist in different states of compression depending on their temperatures.

BLC enables data compression at the data block (DB) level of the file system. Compression reduces the amount of storage required for a given amount of data. The BlockLevelCompression field of DBS Control enables and disables BLC.

**Related Information**

For more information on...	See...
Block-level compression	<i>Teradata Vantage™ - Database Design</i> , B035-1094.
BLOCKCOMPRESSION table attribute	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.
Enabling TBBLC	<a href="#">EnableTempBLC</a> .
Ferret COMPRESS, UNCOMPRESS, and FORCE commands	<a href="#">Ferret Utility (ferret)</a> .
Query bands for compression	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.
Storage group of DBS Control fields	<a href="#">Storage Group</a> .

## TempBLCRescanPeriod

The number of days between two scan intervals of the temperature-based block-level compression (TBBLC) task, where TBBLC data is validated to determine if data needs to be compressed or uncompressed. This determination is based on data temperature changes. When the new scan cycle starts, TBBLC either starts a completely new scan or continues where it was. See Usage Notes below.

This field is effective only when TBBLC is enabled.

### Field Group

Compression

### Valid Range

1 through 90 days

### Default

7 days

### Changes Take Effect

The first time the background compression task runs after the DBS Control Record has been written.

### Usage Notes

Each AMP has a TBBLC task that starts scanning through the Master Index after a database restart, starting at the table that occupies the most cylinders.

Following that initial scan, the task always starts a new scan after the number of days defined in TempBLCRescanPeriod.

Depending on the progress of TBBLC in the previous scan cycle, one of the following will occur:

- If all cylinders of the AMP were processed by TBBLC during the previous scan cycle, a new TBBLC scan starts from scratch, starting at the table that occupies the most cylinders.
- If the previous TBBLC scan cycle had not completed processing all cylinders of the AMP:
  - The TBBLC task continues with the same table that it was going to process in the previous cycle, but the cylinders that were processed in the previous cycle are marked to be scanned again.
  - Eventually the TBBLC task will need to scan all cylinders again, including cylinders that were already scanned in the previous cycle.
  - A TBBLC scan cycle is not considered complete until all cylinders are processed during a given TempBLCRescanPeriod.

Assuming stable temperatures for AUTOTEMP tables, cylinders that were done previously should be quickly processed because on most cylinders compression/uncompression should not be necessary.

Data temperatures refers to frequency of access:

- Hot data is frequently accessed.
- Cold data is infrequently accessed.
- Warm data is accessed moderately frequently.

Data temperature is measured separately for each AMP at a cylinder level and is a relative measure because the temperature of cylinders can change even if the data it is not directly accessed. This is because it depends on how often other cylinders are accessed in comparison. TBBLC automatically compresses cold data to save storage space and decompresses Warm or Hot data if it was previously Cold (and thus compressed). At any time, different portions of a table, which are spread among many cylinders, may exist in different states of compression depending on their temperatures.

BLC enables data compression at the data block (DB) level of the file system. Compression reduces the amount of storage required for a given amount of data. The BlockLevelCompression field of DBS Control enables and disables BLC.

### Related Information

For more information on...	See...
Block-level compression	<i>Teradata Vantage™ - Database Design</i> , B035-1094.
BLOCKCOMPRESSION table attribute	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.
Enabling TBBLC	<a href="#">EnableTempBLC</a> .
Ferret COMPRESS, UNCOMPRESS, and FORCE commands	<a href="#">Ferret Utility (ferret)</a> .
Query bands for compression	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.
Storage group of DBS Control fields	<a href="#">Storage Group</a> .

## TempBLCSpread

Specifies the percentage above and below the [TempBLCThresh](#) setting at which data should be compressed or uncompressed if temperature-based block-level compression (TBBLC) is enabled. For example, if TempBLCThresh is COLD and TempBLCSpread is 5%, the data must be 5% colder than COLD to be compressed and 5% warmer than COLD to be uncompressed.

This setting can be used to prevent data that normally has a temperature close to the threshold from being repeatedly compressed and uncompressed.

### Field Group

Compression

### Valid Range

1 through 25 percent

### Default

5%

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

Data temperatures refers to frequency of access:

- Hot data is frequently accessed.
- Cold data is infrequently accessed.
- Warm data is accessed moderately frequently.

Data temperature is measured separately for each AMP at a cylinder level and is a relative measure because the temperature of cylinders can change even if the data it is not directly accessed. This is because it depends on how often other cylinders are accessed in comparison. TBBLC automatically compresses cold data to save storage space and decompresses Warm or Hot data if it was previously Cold (and thus compressed). At any time, different portions of a table, which are spread among many cylinders, may exist in different states of compression depending on their temperatures.

BLC enables data compression at the data block (DB) level of the file system. Compression reduces the amount of storage required for a given amount of data. The BlockLevelCompression field of DBS Control enables and disables BLC.

When changed, causes TBBLC to immediately start a new cycle.

### Related Information

For more information on...	See...
Block-Level Compression	<i>Teradata Vantage™ - Database Design</i> , B035-1094.

For more information on...	See...
BLOCKCOMPRESSION table attribute	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.
Enabling TBBLC	<a href="#">EnableTempBLC</a> .
Ferret COMPRESS, UNCOMPRESS, and FORCE commands	<a href="#">Ferret Utility (ferret)</a> .
Query bands for compression	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.
Storage group of DBS Control fields	<a href="#">Storage Group</a> .

## TempBLCThresh

Specifies the temperature at and below which data blocks are automatically compressed if temperature-based block-level compression (TBBLC) is enabled.

### Field Group

Compression

### Valid Settings

Setting	Description
COLD	Only COLD data is compressed by TBBLC. By default, cold data is the least frequently accessed 20% of data.
WARM	WARM and COLD data is compressed by TBBLC.
HOT	All data is compressed by TBBLC.

### Default

COLD

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

Tables using TBBLC are those having the BLOCKCOMPRESSION table attribute set to AUTOTEMP.

Data temperatures refers to frequency of access:

- Hot data is frequently accessed.
- Cold data is infrequently accessed.
- Warm data is accessed moderately frequently.

Data temperature is measured separately for each AMP at a cylinder level and is a relative measure because the temperature of cylinders can change even if the data it is not directly accessed. This is because it depends on how often other cylinders are accessed in comparison. TBBLC automatically compresses cold data to save storage space and decompresses Warm or Hot data if it was previously Cold (and thus compressed). At any time, different portions of a table, which are spread among many cylinders, may exist in different states of compression depending on their temperatures.

BLC enables data compression at the data block (DB) level of the file system. Compression reduces the amount of storage required for a given amount of data. The BlockLevelCompression field of DBS Control enables and disables BLC.

When changed, causes TBBLC to immediately start a new cycle.

**Related Information**

For more information on...	See...
Block-level compression	<i>Teradata Vantage™ - Database Design</i> , B035-1094.
BLOCKCOMPRESSION table attribute	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.
Enabling TBBLC	<a href="#">EnableTempBLC</a> .
Ferret COMPRESS, UNCOMPRESS, and FORCE commands	<a href="#">Ferret Utility (ferret)</a> .
Query bands for compression	<i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.
Storage group of DBS Control fields	<a href="#">Storage Group</a> .



## TempLargePageSize

Specifies the size (in KB) of the large memory allocation storage page used for temporary storage in the Relay Services Gateway (RSG).

### Field Group

General

### Valid Range

64 through 1024 KB

### Default

64 KB

### Changes Take Effect

After the next database restart.

### Usage Notes

If a transaction contains any record that exceeds the standard page size defined by the Temporary Storage Page Size DBS Control field, the RSG will allocate memory using the large page size. If the record exceeds the large page size, the entire transaction is spilled to disk.

---

#### Note:

Pages are not shared between transactions, so increasing the page size may cause a decrease in storage utilization efficiency.

---

### Related Information

See [Temporary Storage Page Size](#).

## Temporal Behavior

Determines the default temporal behavior for temporal DDL, queries and DML statements.

### Note:

This field should be changed only under the direction of Teradata Support Center personnel or field engineers.

### Field Group

General

### Valid Settings

Setting	Description
0	Default temporal session qualifier is CURRENT VALIDTIME AND CURRENT TRANSACTIONTIME, compatible with Teradata's original implementation of temporal tables and syntax documented in <i>Teradata Vantage™ - Temporal Table Support</i> , B035-1182.
1	Default temporal session qualifier is ANSIQUALIFIER, required to use ANSI-compliant temporal tables documented in <i>Teradata Vantage™ - ANSI Temporal Table Support</i> , B035-1186.

### Default

- 0 for systems that were upgraded from releases prior to Teradata Database Release 15.0, and that had the Teradata Temporal feature enabled.
- 1 for all other systems.

### Changes Take Effect

After the DBS Control Record has been written.

### Related Information

For more information on temporal tables and syntax, see *Teradata Vantage™ - ANSI Temporal Table Support*, B035-1186 and *Teradata Vantage™ - Temporal Table Support*, B035-1182.

## Temporary Storage Page Size

Specifies the standard memory allocation granule for Relay Services Gateway (RSG) temporary storage.

### Field Group

General

### Valid Range

1 through 1024 KB

### Default

4 KB

### Changes Take Effect

After the next database restart.

### Usage Notes

If a transaction contains any record that exceeds the standard page size, the RSG allocates memory from the large page size defined by the TempLargePageSize field. If the large page size is not sufficient to hold the record, the entire transaction is spilled to disk.

---

#### Note:

Pages are not shared between transactions, so an increase in the page size may cause a decrease in storage utilization efficiency.

---

### Related Information

For more information, see [TempLargePageSize](#).

## TIM Cache Load Disabled

Determines whether the Teradata Intelligent Memory (TIM) cache preload task is enabled.

When enabled, the TIM preload task loads the most frequently accessed data into the TIM cache (a portion of FSG cache) during system startup, after the logons-enabled stage. This allows faster access to this data.

### Note:

This field applies only when FSG cache is not otherwise preserved across a system reset, such as when the `tpareset -x` or `-f` options are used to force a reset.

### Field Group

Storage

### Valid Settings

Setting	Description
TRUE	TIM preloading task is disabled. Data is not preloaded into cache during Teradata system startup.
FALSE	TIM preloading task is enabled. TIM preloads the most frequently accessed data into FSG cache during system startup. This is the default.

### Default

FALSE

### Changes Take Effect

During the next Teradata system restart.

### Usage Notes

Frequency of data access is monitored by Vantage. Cylinders containing the most frequently accessed data are designated to have a data temperature of VERYHOT. These cylinders are loaded into the TIM cache during system startup.

### Related Information

For more information, see [TIM Cache Load Throttle](#).

## TIM Cache Load Throttle

Determines whether the Teradata Intelligent Memory (TIM) cache preload task is throttled during system startup to allow other I/O operations to complete first.

The TIM preload task loads the most frequently accessed data into the TIM cache (a portion of FSG cache) during system startup, after the logons-enabled stage. This allows faster access to this data.

### Note:

This field applies only when FSG cache is not otherwise preserved across a system reset, such as when the `tpareset -x` or `-f` options are used to force a reset.

### Field Group

Storage

### Valid Settings

Setting	Description
TRUE	TIM background cache preload process is throttled. It only loads VERYHOT cylinders into cache when the database I/O subsystem is idle. This can improve startup performance on busy systems.
FALSE	TIM background loading task is unthrottled, and loads VERYHOT cylinders into cache immediately after the system startup sequence reaches the logons-enabled stage. This is the default.

### Default

FALSE

### Changes Take Effect

During the next Teradata system restart.

### Usage Notes

Frequency of data access is monitored by Vantage. Cylinders containing the most frequently accessed data are designated to have a data temperature of VERYHOT. These cylinders are loaded into the TIM cache during system startup.

### Related Information

For more information, see [TIM Cache Load Disabled](#).

## TimeDateWZControl

Determines how TIME and TIMESTAMP data is stored in the Teradata file system. Also determines whether the built-in functions CURRENT\_TIME, CURRENT\_TIMESTAMP, and CURRENT\_DATE reflect the session time and session time zone offset.

---

### Note:

This field does not affect the output of the DATE and TIME built-in functions.

---

### Field Group

General

### Valid Settings

Setting	Description
0	Disabled. The values returned by the built-in time and date functions reflect the time value local to the database server, but the time zone value that is defined for user or session. This is identical to a setting of 1.
1	Disabled. The values returned by the built-in time and date functions reflect the time value local to the database server, but the time zone value that is defined for user or session. This is identical to a setting of 0.
2	Enabled. The values returned by the built-in time and date functions reflect the session time and session time zone. DateTime values are stored in the database as UTC.
3	Enabled. The values returned by the built-in time and date functions reflect the session time and session time zone. DateTime values without zone information are stored in the database as system local time. System local time offset is defined either using an SDF file and the tdlocaledef utility, or using the System TimeZone Hour and System TimeZone Minute DBS Control settings.

### Default

2 for:

- new Teradata systems.
  - systems that have been upgraded or re-initialized (sysinited) from prior database releases in which TimeDateWZControl had been 0 and System TimeZone Hour or System TimeZone Minute had been non-zero.
- 

### Note:

For systems that have been upgraded or sysinited from prior database releases in which TimeDateWZControl had been a non-zero value, that value remains unchanged after the upgrade or sysinit.

---

## Changes Take Effect

After the DBS Control Record has been written.

## Usage Notes

TimeDateWZControl must be enabled to use temporal tables. For more information on temporal tables, see *Teradata Vantage™ - ANSI Temporal Table Support*, B035-1186 and *Teradata Vantage™ - Temporal Table Support*, B035-1182.

The session time zone can be set in the following ways:

- TIME\_ZONE parameter of the CREATE USER statement  
Sets the default time zone displacement from UTC used for sessions opened by the user.
- SET TIME\_ZONE statement  
Sets the time zone displacement from UTC for the current session. Overrides the time zone defined for the current user.

If no session time zone is defined in either of these ways, the session time zone defaults to the database system time zone, as defined by the DBS Control fields [System TimeZone Hour](#) and [System TimeZone Minute](#).

## Examples: Using the TimeDateWZControl field of DBS Control

For the following examples, assume:

- The time local to the database server is 11:59:00 Coordinated Universal Time (UTC), January 31, 2010.
- User TK lives in Tokyo, and has a time zone defined as +9 hours offset from UTC.
- User LA lives in Los Angeles, and has a time zone defined as -8 hours offset from UTC.
- User TK and User LA run the CURRENT\_TIMESTAMP function at exactly the same time.

When this field is enabled:

For User TK, the CURRENT\_TIMESTAMP function returns:

```
2010-02-01 20:59:00.000000+09:00
```

For User LA, the CURRENT\_TIMESTAMP function returns:

```
2010-01-31 03:59:00.000000-08:00
```

When this field is disabled:

For User TK, the CURRENT\_TIMESTAMP function returns:

```
2010-01-31 11:59:00.000000+09:00
```

For User LA, the CURRENT\_TIMESTAMP function returns:

```
2010-01-31 11:59:00.000000-08:00
```

CURRENT\_TIME and CURRENT\_DATE functions display analogous behavior.

**Related Information**

For more information on...	See...
Teradata built-in time and date functions	<i>Teradata Vantage™ - SQL Date and Time Functions and Expressions</i> , B035-1211.
Setting session time zones	Information about SET TIME ZONE, CREATE USER, and MODIFY USER in <i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144.
DateTime, Interval, and Period data types	<i>Teradata Vantage™ - Data Types and Literals</i> , B035-1143.
System TimeZone fields	<a href="#">System TimeZone Hour</a> and <a href="#">System TimeZone Minute</a> .
Temporal tables	<i>Teradata Vantage™ - Temporal Table Support</i> , B035-1182.



## TruncRoundReturnTimestamp

Determines whether the TRUNC and ROUND system functions return a TIMESTAMP or DATE value when passed a TIMESTAMP value.

### Field Group

General

### Valid Settings

Setting	Description
TRUE	TRUNC and ROUND return a TIMESTAMP data type value when passed a TIMESTAMP value.
FALSE	TRUNC and ROUND return a DATE data type value when passed a TIMESTAMP value.

### Default

FALSE

### Changes Take Effect

After the next database restart.

### Usage Notes

The default behavior of returning a DATE value when passed a TIMESTAMP value is for compatibility with Teradata Database releases prior to 16.10.

### Related Information

For more information on the TRUNC and ROUND system functions, see *Teradata Vantage™ - SQL Functions, Expressions, and Predicates*, B035-1145.

# UncompressReservedSpace

Specifies the minimum percentage of storage space that must remain available while DBs are uncompressed using the Ferret UNCOMPRESS command. The uncompress operation is terminated at the point where this threshold would be exceeded.

## Field Group

Compression

## Valid Range

1 through 90%

## Default

20%

## Changes Take Effect

After the DBS Control Record has been written.

## Usage Notes

The table undergoing decompression at the time this field value is exceeded may remain in a partially compressed state. This will not affect the functioning of the table or database. To make the table DBs consistent with respect to compression, either issue the Ferret COMPRESS command, or free some storage space, then reissue the UNCOMPRESS command.

Set this field to represent the expected peak amount of required spool space.

## Related Information

For more information on...	See...
COMPRESS and UNCOMPRESS commands	<a href="#">Ferret Utility (ferret)</a> .

## UserRetryLimit

Determines the number of times Vantage will re-try a request to an external file system after the request timeout or connection timeout has been exceeded and request retries have been commenced. After this number of retries, the request is aborted. This setting is effective only when the Native Object Store (NOS) feature is enabled.

### Field Group

Native Object Store

### Valid Settings

0 to 4

0 disables request retries for external file systems.

### Default

1

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

*Foreign tables* allow Vantage to access data from external, cloud-based data storage, such as AWS S3, without requiring you to manually move the data into the database first from where it natively resides. Foreign tables are identified by a hostname, path, and other metadata that point to the external storage. Vantage can read and process semi-structured or unstructured external data in foreign tables using standard SQL. For example, you can use Teradata analytic functions to examine the data, join it to the relational data in the database, and issue queries against it as you can for other data in Vantage.

### Related Information

The following fields affect the NOS feature and foreign table access.

- [EnableNOS](#)
- [ConnectionBufferSize](#)
- [ConnectionTimeout](#)
- [DefaultRowGroupSize](#)
- [RequestDuration](#)
- [RequestTimeout](#)

## UseVirtualSysDefault

This field is no longer used. For cost profiling information, see [CostProfileId](#).

### Field Group

General

## UtilityReadAheadCount

Specifies the number of data blocks beyond the current block that will be preloaded into memory during file system utility (for example, Ferret) operations, such as SCANDISK, that occur when Vantage is running.

---

**Note:**

To enable the read-ahead feature, the ReadAhead field must be set to TRUE.

---

**Field Group**

Performance

**Valid Range**

1 through 100 blocks

**Default**

10 blocks

**Changes Take Effect**

After the DBS Control record has been written.

**Usage Notes**

The number of data blocks to be preloaded during file system utility operations when Vantage is not running is determined by the StandAloneReadAheadCount setting.

**Related Information**

- [ReadAhead](#)
- [Read Ahead Count](#)
- [StandAloneReadAheadCount](#)

# Version

**NOTICE**

This field is modified only during system initialization, which destroys all user and dictionary data.

Indicates the version number of the DBS Control Record.

**Field Group**

General

**Valid Range**

1 ... MAXLONGINT

**Default**

4

**Usage Notes**

The Version field is incremented by one when the DBS Control Record must be migrated to a new format.

## WAL Buffers

Determines the number of WAL append buffers allocated by the File System.

### Field Group

File System

### Valid Range

5 through 40

### Default

20

### Changes Take Effect

After the next database restart.

### Usage Notes

A larger number of buffers increases the chance that there will be an available buffer to hold a WAL log record when a task needs to append a WAL log record.

A smaller number of buffers risks that buffers may be unavailable because they are full and writes are pending.

## WAL Checkpoint Interval

Determines the amount of time that elapses between WAL checkpoints.

### Field Group

File System

### Valid Range

1 through 240 seconds

### Default

60 seconds

### Changes Take Effect

After the DBS Control Record has been written.

### Usage Notes

A WAL checkpoint is used to indicate the oldest part of the WAL log that must be scanned when recovering from a system crash. It differentiates the WAL log records that have been written to disk from the records that must be applied during system recovery. The checkpoint is used as the starting point for the Redo forward scan of the WAL log during recovery.



## WorkDBSize

Specifies the maximum size for multirow data blocks in work tables. Rows that are larger than WorkDBSize are stored in single-row data blocks, which are not limited by WorkDBSize.

### Field Group

File System

### Valid Range

42 through 2047 sectors for systems that use large cylinders.

18 through 512 sectors for systems that use small cylinders.

A sector is 512 bytes.

### Default

254 sectors

Setting this field to 0 causes Vantage to use the system default size for this type of data block.

### Changes Take Effect

After a system initialization (sysinit) or an upgrade to a major xx.00 release.

### Usage Notes

When database tables are initially populated, Vantage stores as many rows as possible into each data block, until the block reaches the size specified by the various DB size settings in DBS Control. As tables are subsequently modified, rows can grow such that the existing data blocks would exceed the maximum size. When this happens, the data block is split, and roughly half the rows are moved to a new data block, with the result that the original and new data blocks are each one half of the original size. The result of this type of growth and splitting is that data blocks for heavily modified tables tend to be about 75% of the maximum size defined in DBS Control.

Systems initialized on Teradata Database 13.10 or later release use large cylinders. On large-cylinder systems, the minimum data block size is 42 sectors (21504 bytes). The minimum data block size accepted by the CREATE TABLE and ALTER TABLE SQL statements is 21248 bytes (41.5 sectors). Tables created or altered to use this minimum size will have the value rounded up by the parser to 42 sectors, so these tables will have an actual minimum data block size of 21504 bytes.

Systems initialized on a Teradata Database release prior to 13.10 and subsequently upgraded without a sysinit use small cylinders. On small-cylinder systems, the minimum data block size is 18 or 19 sectors (9216 bytes or 9728 bytes). The minimum data block size accepted by the CREATE TABLE and ALTER TABLE statements is 8960 bytes (17.5 sectors). Tables created or altered to use this minimum size will have the value rounded up by the parser to 18 sectors, so these tables will have an actual minimum data block size of 9216 bytes.

If DBs are compressed, this setting applies to the size of the uncompressed DBs.

## XML\_Memory Limit

Specifies the maximum amount of memory available for operations involving the XML data type.

### Field Group

General

### Valid Range

0 through 32 MB

### Default

4 MB

### Changes Take Effect

When the DBS Control record is written.

### Usage Notes

The XML data type allows you to store XML content in Vantage in a compact binary form that preserves the information set of the XML document.

## 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.

## Checksum Fields

Checksums can be used to check the integrity of database disk I/O operations. A checksum is a calculated numeric value computed from a given set of data, or specific portions of the data. For a given set of data, the checksum value will always be the same, provided the data is unchanged.

Checksums can be used to detect when there are errors in disk I/O operations. When checksums are enabled, and data is initially read, a checksum is calculated for the data and stored in the system. When the same data is subsequently read, the checksum is recalculated and compared to the original checksum value. Differing checksum values for a given set of data indicate an inconsistency in the data, most often due to errors in disk I/O operations.

The Checksum group of fields shows the current checksum settings for these classes of tables:

- System
- System Journal
- System Logging
- User

- Permanent Journal
- Temporary

These table classes are described in the sections that follow.

Because calculating checksums requires system resources, and may affect system performance, the checksum feature is disabled by default on most platforms, but can be enabled by the Teradata Support Center if disk corruption is suspected.

## System Tables

Enables or disables checksums for system tables.

---

**Note:**

This field can be changed only by Teradata Support Center personnel.

---

**Default**

OFF

**Usage Notes**

System tables include all system table types in database DBC (for example, data dictionaries, and session information). These tables have the two-value, unique portion of their table IDs in the range of (0, 1) through (0, 999).

The System Tables checksum setting does not affect tables included under System Journal tables and System Logging tables.

## System Journal Tables

Enables or disables checksums for system journal tables.

---

**Note:**

This field can be changed only by Teradata Support Center personnel.

---

**Default**

OFF

**Usage Notes**

System journal tables include the following transient journals, change tables, and recovery journals:

- DBC.ChangedRowJournal
- DBC.LocalTransactionStatusTable
- DBC.UtilityLockJournalTable
- DBC.LocalSessionStatusTable
- DBC.SysRcvStatJournal (System Recovery Status Journal)
- DBC.SavedTransactionStatusTable
- DBC.OrdSysChngTable (Ordered System Change Table)
- DBC.RecoveryLockTable
- DBC.RecoveryPJTable (Recovery Permanent Journal Table)

## System Logging Tables

Enables or disables checksums for system logging tables.

---

**Note:**

This field can be changed only by Teradata Support Center personnel.

---

**Default**

OFF

**Usage Notes**

System logging tables include the following:

Table	Description
DBC.AccLogTbl	Logging activity controlled by DBC.AccLogRuleTbl
DBC.Acctg	Log of each account a user owns on each AMP
DBC.EventLog	Log of session events
DBC.RCEvent	Log of storage media for events
DBC.SW_Event_Log	Log of software system errors

All Resource Usage (RSS) tables are also affected by the System Logging tables checksum field.



## User Tables

Enables or disables checksums for user tables.

---

**Note:**

This field can be changed only by Teradata Support Center personnel.

---

**Default**

OFF

**Usage Notes**

User tables include all tables for which the two-value, unique portions of the table ID is in the range of (0, 1001) through (16383, 65535). These include the following:

- Stored procedures
- User-defined functions
- User-defined methods
- Join indexes
- Hash indexes

This also includes fallback for these and secondary indexes for tables and join indexes.

## Permanent Journal Tables

Enables or disables checksums for permanent journal tables.

---

**Note:**

This field can be changed only by Teradata Support Center personnel.

---

**Default**

OFF

**Usage Notes**

Permanent journal tables include tables that have the two-value, unique portions of their table IDs in the range of (16384, 0) through (32767, 65535).

## Temporary Tables

Enables or disables checksums for temporary and spool tables.

---

**Note:**

This field can be changed only by Teradata Support Center personnel.

---

**Default**

OFF

**Usage Notes**

Temporary tables include all temporary and spool tables. These are tables that have the two-value, unique portions of their table IDs in the range of (32768, 0) through (65535, 65535), and include the following:

- Global temporary tables
- Volatile tables
- Intermediate result spool tables
- Response spool tables
- Persistent spool tables

## Dump Unload/Load Utility (dul)

---

**Note:**

The utility described is not available for as-a-service customers.

---

The Dump Unload/Load utility, `dul`, saves and restores system dump tables.

If an Advanced SQL Engine failure occurs, the system automatically saves the contents of the affected AMP and its associated PE in a system-generated “dump table.” The information in these tables can be used to determine the cause of a system failure. Use `dul` to load the dump tables to disk.

Before you can use `dul`, the associated packages must be installed on Advanced SQL Engine.

The `dul` utility:

- Transfers dump information from Advanced SQL Engine to file. This unload operation is normally performed at the customer site, where the crash dump information is transferred to the Teradata Support Center for analysis.
- Is used by Teradata Support Center personnel to restore the dump information to a Support system for analysis.
- Can be used to drop tables and obtain summary information about dumps without performing a load or unload operation.

Tables containing crash dump information are named according to the following syntax:

```
CrashDumps.Crash_yyyymmdd_hhmmss_nn
```

***yyymmdd***

Year, month, and day.

***hhmmss***

Hour, minute, and second.

***nn***

Sequence number, which is increased by one for each dump saved.

`Dul` can unload dump tables having any name.

`Dul` automatically saves unloaded dumps to a file with a compressed, gzip file format, and adds a `.gz` extension to the dump file name. When loading dumps, `dul` looks for the specified file name including a `.gz` extension. If that file is not found, `dul` looks for the specified file name without a `.gz` extension.

## Runs From

- Linux command line (system node or client)

For general information about starting the utilities from different interfaces, see [Starting the Utilities](#).

## Starting and Running Dul

Dul runs on SUSE Linux Enterprise Server 10, 11, and 12 in either batch or interactive mode.

Because dul is used to move large amounts of data, it is usually run in batch mode.

### Starting Dul

You can run dul interactively from the Linux command line. To start dul interactively, do the following:

1. At the command prompt, type the following and press **Enter**:

```
dul
```

Dul prompts you for your logon:

```
Dump Unload/Load - Enter your logon:
```

2. Type:

```
.logon crashdumps
```

and press **Enter**.

3. Enter the password for the crashdumps logon.

---

#### Note:

The default password for the crashdumps logon is crashdumps, however, your System Administrator may have changed the password.

---

To log off the database and exit du, type LOGOFF, END, or QUIT followed by a semicolon, and press **Enter**.

## Space Requirements

The following sections discuss the space requirements for the database and hosts.

### Crashdumps Database

The crashdumps database is used to store system dumps. Because system dumps take up disk space, you should periodically clear old or unwanted dump tables from the crashdumps database.

If the space available in the crashdumps database is exceeded, a message is displayed on the system console every hour until you make space available for the dump to be saved. To make more room for new dumps, examine the existing dumps and delete the ones that you no longer need.

If you still need the existing dumps, then copy them to removable media. Once the Teradata Support Center has received and evaluated a dump, you can delete the tables.

## Database Space Allocation

The proper size for the crashdumps database depends on the following configuration information:

- Number of nodes on the system
- Number of dumps you want to have available online at any one time (minimum of one dump)

The size of the dump for any given node depends on several unpredictable variables. To calculate the approximate size of the crashdumps database that you need for your system, use the following formula:

*number of DBS nodes x number of dumps x 100 MB*

Allow at least 100 MB of space per dump. Teradata recommends even larger multiples, if available.

## Privileges

To perform a dump/unload operation, you must have the following Vantage privileges:

- CREATE, DROP, and SELECT privileges on the tables in the crashdumps database
- SELECT privileges on the DBC.SW\_EVENT\_LOG system table

To perform a load operation, you must have CREATE, DROP, and SELECT privileges on the DBC database tables.

If the username you specify when you invoke dul does not have the appropriate privileges, then the system returns an error message and the operation is canceled.

For detailed information on access privileges, see *Teradata Vantage™ - Database Administration*, B035-1093 or consult your system administrator.

## Saving Linux Dumps to Removable Media

Use the Linux commands below to save a dump to removable media. Each site is different, so the exact commands, file names, and device names will vary.

To save the dump from one node to another that has a tape drive, do the following:

1. After the node where the dump occurred has been recovered and is up and running, save the dump to a disk file.

For example:

```
csp -mode save -target stream
```

---

### Note:

For a full description of the *csp* command, see the man page or pdehelp for csp.

---

2. After the file is written to disk, FTP the dumpfile to the node with the tape drive.
3. After you have FTPed the dumpfile to the node with the tape drive, copy the file to tape:

```
dd if=input_dumpfile of=output_device_or_file bs=block_size_bytes
```

**Note:**

For a description of the dd command, see the Linux man page for the command.

4. If necessary, perform step 3 again until all of the dump files are on the tape. Then remember to use the “no rewind on open” and “no rewind on close” tape device options.

Use standard operating system commands to burn the dump file to removable optical media, such as DVD.

## Mailing Crash Dumps to the Teradata Support Center

To mail a crash dump that has been saved to removable media:

1. Label every cartridge or disk with the following:
  - The incident number.
  - The database version number (for example, 16.00.00.00 or 15.10.01.00).
  - A database dump or a UNIX® operating system dump.

If applicable, write a volume number, such as 1 of 4, 2 of 4, and so on, on each cartridge or disk.

2. Write the Incident number on the outside of the package.
3. Include your name or some other person to contact as part of your return business address.
4. Address the package to the following address:

```
Teradata Corporation
Dump Administrator
Teradata Customer Support Engineering
Ref.: Incident #number
17095 Via del Campo
San Diego, CA 92127
```

**Note:**

The version, incident number, and volume number are necessary for the Teradata Support Center tester to know which installation the dump is reporting and how to load the crash dump properly.

## Restarting DUL

### Restarting During a Load Operation

If your system fails during a dul load operation, dul must be restarted. You must resubmit the LOAD command.

To restart dul during a load operation, enter the following commands.

Command	Description
LOGON ZZ/Admin, abc ;	Logs user Admin onto the database.
HELP DATABASE crashdumps ;	Lists any dump tables that might have been created during a load operation.
DROP crashdumps.crash_20000606_142500_01 ;	Ensures that any partially created tables are deleted prior to submitting a new load operation.
SELECT ERROR ;	Sets the selection criteria that determines the dump data that is loaded into tables on your database. In the example above, only processors that contain error codes are selected. The LOAD command can then be used to resubmit the operation.
LOAD	Resubmits the load operation. For example: LOAD CL200, FALLBACK FILE = FILEPATH; For a detailed explanation of loading dump files into tables, see <a href="#">LOAD</a> .

## Restarting During an Unload Operation

If your system fails during an unload operation, dul must be restarted. You must resubmit the UNLOAD command.

To restart dul during an unload operation, enter the following commands.

Command	Description
LOGON ZZ/ Admin, abc ;	Logs user Admin onto the database.
SELECT ERROR ;	Sets the selection criteria that determines the dump data that is unloaded onto your host system. In the previous example, only processors that contain error codes are selected. The UNLOAD command can then be used to resubmit the unload operation.
UNLOAD	Resubmits the unload operation. For example: UNLOAD crashdumps.crash_20000606_142500_01 FILE=filepath; For a detailed explanation, see <a href="#">UNLOAD</a> .

## Return Codes

Dul issues return codes to report processing success or failure. Dul supports return codes via shell commands (\$?).

A return code of 0 indicates that processing was successful; a nonzero return code indicates that processing failed. In order of severity, dul return codes are 02, 04, 08, and 12. These codes are defined as shown below:

- 02 - Special warning



- 04 - Warning
- 08 - User error
- 12 - Severe internal error

A 02 code is returned if you attempt a Vantage operation without logging onto the system.

The following table shows messages resulting in dul Return Code 04.

Return Code	Error Code	Description
04	3747	No startup string defined for this user.
04	3803	Table <i>table_name</i> already exists.
04	3804	View <i>view_name</i> already exists.
04	3805	Macro <i>macro_name</i> already exists.

The following table shows messages resulting in dul Return Code 08.

Return Code	Error Code	Description
08	2538	A disk read error occurred in the tables area.
08	2541	End of Hash Code range reached.
08	2631	Transaction ABORTED due to <i>abort_reason</i> .
08	2632	All AMPs own sessions for this Fast/MultiLoad.
08	2639	Too many simultaneous transactions.
08	2641	<i>database_name.table_name</i> was restructured. Resubmit.
08	2644	No more room in database <i>database_name</i> .
08	2654	Operation not allowed: <i>database_name.table_name</i> is being Restored.
08	2805	Maximum row length exceeded in <i>table_name</i> .
08	2809	Invalid recovery sequence detected.
08	2815	Apparent invalid restart of a restore.
08	2818	Invalid lock to dump table without after-image journaling.
08	2825	No record of the last request was found after Teradata Database restart.
08	2826	Request completed but all output was lost due to Teradata Database restart.
08	2827	Request was aborted by user or due to command error.
08	2828	Request was rolled back during system recovery.
08	2830	Unique secondary index must be dropped before restoring table.
08	2835	A unique index has been invalidated; resubmit request.

Return Code	Error Code	Description
08	2837	Table being FastLoaded; no data dumped.
08	2838	Table is unhashed; no data dumped.
08	2840	Data rows discarded due to inconsistent hash codes.
08	2843	No more room in the database.
08	2866	Table was Recovery Aborted; no data dumped.
08	2868	This permanent journal table is damaged; no data dumped.
08	2920	Delete journal and AMP down without dual.
08	2921	No saved subtable for journal <i>database_name.table_name</i> .
08	2926	No more room in <i>database_name.table_name</i> .
08	3001	Session is already logged on.
08	3111	The dispatcher has timed out the transaction.
08	3116	Response buffer size is insufficient to hold one record.
08	3119	Continue request submitted but no response to return.
08	3120	The request is aborted because of a Teradata Database recovery.
08	3523	<i>User</i> does not have <i>privilege</i> access to <i>database_name.table_name</i> .
08	3524	<i>User</i> does not have <i>privilege</i> access to database <i>database_name</i> .
08	3566	Teradata Database does not have a PERMANENT journal.
08	3596	RESTORE Teradata Database invalid if table, view, or macro exists outside of Teradata Database.
08	3598	Concurrent change conflict on Teradata Database. Please try again.
08	3603	Concurrent change conflict on table. Please try again.
08	3613	Dump/Restore, no hashed nonfallback tables found.
08	3656	Journal table specified no longer exists.
08	3658	ROLLBACK/ROLLFORWARD table specifications are invalid.
08	3705	Teradata Database/SQL request is longer than the Simulator maximum.
08	3737	Name is longer than 30 characters.
08	3802	Database <i>database_name</i> does not exist.
08	3807	Table/view <i>view_name</i> does not exist.

The following table shows messages resulting in dul Return Code 12.

Return Code	Error Code	Description
12	CLI0001	Parameter list invalid or missing.
12	CLI0002	Invalid number of parameters received.
12	CLI0003	Error validating HSIRCB.
12	CLI0004	Error validating HSICB.
12	CLI0005	Error validating HSISPB.
12	CLI0006	Invalid destination HSICB detected.
12	CLI0007	Invalid destination RCB detected.
12	CLI0008	DBCFCRC unable to free RCB/HSICB control blocks because they are not contiguous in storage.
12	CLI0009	Invalid DBCAREA pointer or id.
12	CLI0010	ECB already waiting.
12	CLI0530	Character Set Name or Code unknown.
12	2123	A segment could not be read successfully.
12	2971	The AMP Lock table has overflowed.
12	2972	No table header exists for table.

For details on error messages, see *Teradata Vantage™ - Database Messages*, B035-1096.

## DUL Commands

### DUL Command Syntax

Unless otherwise specified, a DUL command must either begin with a period or end with a semicolon, and it can do both:

```
{ [.] dul_command |
  dul_command [;] |
  [.] dul_command [;]
}
```

For example, these are equivalent:

```
.SHOW VERSIONS
SHOW VERSIONS ;
.SHOW VERSIONS ;
```

*dul\_command* is a command in either table in the following section.

Syntax for individual DUL commands does not include the period or semicolon.

## DUL Command Categories

Dul commands are divided into the following two categories:

- Session Control
- Data Handling

The following table summarizes the functions of the session control commands.

Command	Function
<a href="#">ABORT</a>	Aborts a LOAD or UNLOAD command.
<a href="#">DATABASE</a>	Changes the default database.
<a href="#">HELP</a>	Displays information about dul commands and dump databases.
<a href="#">LOGOFF</a> or <a href="#">END</a> or <a href="#">QUIT</a>	Ends a Vantage session and exits the dul utility. The END and QUIT commands are synonyms of LOGOFF.
<a href="#">LOGON</a>	Begins a Vantage session.
<a href="#">.OS</a>	Submits a command to your host operating system.
<a href="#">SHOW VERSIONS</a>	Displays dul software module release versions.
<a href="#">SHOW CONTROLS</a>	Displays the current user-configurable settings.

The following table summarizes the functions of the data handling commands.

Command	Function
<a href="#">DROP</a>	Removes a dump table from the database.
<a href="#">LOAD</a>	Moves dump data from removable media to a Teradata system.
<a href="#">NOSPOOL</a>	Uses NOSPOOL feature of the database to dump the crashdump table.
<a href="#">SEE</a>	Reports statistics about the contents of a dump.
<a href="#">SELECT</a>	Sets selection criteria.
<a href="#">UNLOAD</a>	Moves dump data from a table on the database to a file on the host.
<a href="#">MULTIFILE</a>	Instructs DUL to generate or read from multiple split files of defined size during an UNLOAD or LOAD operation respectively.

## ABORT

The ABORT command aborts a LOAD or UNLOAD request.

### Syntax

```
ABORT
```

### Usage Notes

IF you are working under ...	THEN ...
Linux	press the Ctrl+C keys prior to typing the ABORT command.

## DATABASE

The DATABASE command changes the default database for the current Vantage session.

### Syntax

```
DATABASE database ;
```

### Syntax Elements

#### *database*

Name of the new default database.

### Usage Notes

When you invoke dul and log onto the database, the crashdumps database automatically becomes your default database. Vantage uses the database specified in the DATABASE command as the default database until the end of the session, or until you type a subsequent DATABASE command.

To use the DATABASE command, you must have SELECT privileges on the specified database.

### Example: Changing the default database

To make the Personnel database the default database for the current session, type the following:

```
DATABASE Personnel ;
```

The following appears:

```
*** Sending database Personnel to Teradata Database.  
*** New default database accepted.
```

## DROP

The DROP command removes an existing table (created as a result of a system dump) and all of its rows from the specified database on the Teradata system.

### Syntax

```
DROP [ database. ] table ;
```

### Syntax Elements

#### *database*

Name of the database in which the table resides. If the database is not specified, the currently set database is assumed. Use a period (.) to separate the database name from the table name.

#### *table*

Name of the table to be dropped.

### Usage Notes

The DROP command removes the specified table and any tables with the same name that end with a \_C, \_L, or \_M suffix.

In general, enter a DROP command before performing a load operation to remove any existing tables that might have the same name as the table specified on the next LOAD command.

To use the DROP command, you must have the DROP privilege on the specified table.

### Example: Dropping a system dump table

Assume that Crash\_20000407\_1013\_02, Crash\_20000407\_1013\_02\_C, Crash\_20000407\_1013\_02\_L, and Crash\_20000407\_1013\_02\_M tables were produced as an result of the previous load operation.

To drop all four tables from the Crashdumps database, type the following:

```
DROP Crash_20000407_1013_02;
```

The following appears:

```
*** Dropping table Crash_20000407_1013_02;
*** Table has been dropped.
*** Dropping table Crash_20000407_1013_02_1;
*** Failure 3807 Table/view 'Crash_20000407_1013_02_1' does not exist.
        Statement# 1, Info =0
*** Dropping table Crash_20000407_1013_02_2;
*** Failure 3807 Table/view 'Crash_20000407_1013_02_2' does not exist.
```

```

Statement# 1, Info =0
*** Dropping table Crash_20000407_1013_02_C;
*** Table has been dropped.
*** Dropping table Crash_20000407_1013_02_C_1;
*** Failure 3807 Table/view 'Crash_20000407_1013_02_C_1' does not exist.
Statement# 1, Info =0
*** Dropping table Crash_20000407_1013_02_C_2;
*** Failure 3807 Table/view 'Crash_20000407_1013_02_C_2' does not exist.
Statement# 1, Info =0
*** Dropping table Crash_20000407_1013_02_L;
*** Table has been dropped.
*** Dropping table Crash_20000407_1013_02_L_1;
*** Failure 3807 Table/view 'Crash_20000407_1013_02_L_1' does not exist.
Statement# 1, Info =0
*** Dropping table Crash_20000407_1013_02_L_2;
*** Failure 3807 Table/view 'Crash_20000407_1013_02_L_2' does not exist.
Statement# 1, Info =0
*** Dropping table Crash_20000407_1013_02_M;
*** Table has been dropped.
*** Dropping table Crash_20000407_1013_02_M_1;
*** Failure 3807 Table/view 'Crash_20000407_1013_02_M_1' does not exist.
Statement# 1, Info =0
*** Dropping table Crash_20000407_1013_02_M_2;
*** Failure 3807 Table/view 'Crash_20000407_1013_02_M_2' does not exist.
Statement# 1, Info =0
*** Dropping table Crash_20000407_1013_02_S;
*** Failure 3807 Table/view 'Crash_20000407_1013_02_S' does not exist.
Statement# 1, Info =0

```

Dul displays additional messages if you abort the load operation because it tries to dump tables used by the FastLoad utility even if they do not exist.



## END

The END command exits the dul utility and terminates a Vantage session.

### Syntax

```
{ END | LOGOFF | QUIT } ;
```

### Usage Notes

The LOGOFF and QUIT commands are synonyms for the END command.

### Example: Terminating a Vantage session and exiting DUL

To terminate a Vantage session and exit dul, type the following:

```
END ;
```

The following appears:

```
*** DUL Terminated  
*** Highest return code = <n>
```

where *n* is the highest return code by dul.

## HELP

The HELP command returns syntax information about dul commands and lists the tables, views, and macros stored in a database.

### Syntax

```
HELP { DATABASE database | DUL } ;
```

### Syntax Elements

#### *database*

list of tables, views, and macros stored in the specified database.

#### DUL

syntax summary of all the commands available with the dul utility.

### Example: Listing tables, views, and macros in the crashdumps database

To list all tables, views, and macros in the crashdumps database, type the following:

```
HELP DATABASE Crashdumps ;
```

The following appears:

```
*** Sending HELP DATABASE crashdumps; to Teradata Database.
*** Help information returned. 10 rows.
    Crash_20000526_134503_01
    Crash_20000526_154702_01
    Crash_20000526_154925_01
    Crash_20001120_103233_01
    Crash_20001225_163434_01
```

If the database does not exist, the following appears:

```
help database non_exist;
*** Sending HELP DATABASE non_exist; to Teradata Database.
*** Failure 3802 Data base 'non_exist' does not exist.
        Statement# 1, Info =0
```

## LOAD

The LOAD command moves dump data from files created by an unload operation into tables on a database.

### Syntax

```
LOAD [ database. ] table [ FALLBACK ] FILE = filepath ;
```

### Syntax Elements

#### *database*

Name of the database in which the table resides. Use a period (.) to separate the database name from the table name.

#### *table*

Name of the table to receive dump data.

The table cannot already exist. However, the database in which the table will reside must already exist.

#### FALLBACK

Option that creates a fallback copy of the table specified in the LOAD command as well as the tables with the same name ending in \_L, \_C, and \_M.

#### *filepath*

The path of the file created in a previous unload operation.

The *filepath* specification is required.

The *filepath* is specified as *directory/filename*.

---

#### Note:

Dul expects dump files to be in a compressed, gzip format. For compatibility, dul can load compressed files regardless of whether the file name includes the .gz extension.

---

### Usage Notes

Before you can perform a load operation, your username specified in the logon ID must have CREATE, DROP, and SELECT privileges on tables in the DBC database.

If you do not have the appropriate privileges, the system returns an error message, and the operation is canceled.

Before you can perform a load operation, dul displays summary information, such as which processors are selected and error dates about the selection criteria that is set. After the load operation, dul displays event codes, if any exist, for the specified processors. For information on setting selection criteria, see [SELECT](#).

As a general rule, type a DROP command before performing a load operation. This removes any existing tables that might have the same name as the name specified on the current LOAD command. For more information, see [DROP](#).

---

**Note:**

Dul uses the FastLoad utility to improve transfer speed. However, you can still transfer dump files on your host to a database using BTEQ.

---

In most instances, a load operation is not performed at a customer site.

For detailed information on privileges, see *Teradata Vantage™ - Database Administration*, B035-1093 or consult your system administrator.

For more information about the FastLoad utility, see *Teradata® FastLoad Reference*, B035-2411.

For information about BTEQ, see *Basic Teradata® Query Reference*, B035-2414.

**Example: Checking for duplicate table names in the Crashdumps database**

To check whether tables with the same name exist in the crashdumps database, type the following:

```
HELP DATABASE crashdumps ;
```

The following list of all the tables in the crashdumps database appears:

```
*** Sending HELP DATABASE crashdumps to Teradata Database.
*** Help information returned.  4 rows.
CL100
CL100_C
CL100_L
CL100_M
```

Since no tables have the same name, you can type the LOAD command next. If tables with the same name already exist, verify that they are not needed and then use the DROP command to delete them from the Crashdumps database.

**Example: Loading dump information to tables**

To transfer dump information to tables on Vantage, type the following:

```
load crashdumtable1 file=TPFILE;
```

The following appears:

```
load CRASH_20170219_004121_02 file=sanity;
```

```
*** Creating table 'CRASH_20170219_004121_02'.
*** Table has been created.
*** Loading data into 'CRASH_20170219_004121_02'
*** Logging on Amp sessions.
*** Growing Buffer to 789
*** Growing Buffer to 977
*** Growing Buffer to 2198
*** Growing Buffer to 2427
*** Growing Buffer to 2481
*** Growing Buffer to 2662
*** Growing Buffer to 2758
*** Growing Buffer to 4118
*** Starting Row 10000 at Mon Feb 20 19:25:49 2017

*** Starting Row 20000 at Mon Feb 20 19:26:01 2017
.
.
.

*** Starting Row 70000 at Mon Feb 20 19:27:06 2017

*** END LOADING phase...Please stand by...
Loading data into CRASH_20170219_004121_02 completes successfully.

Dump Unload/Load - Enter your command:
```

## LOGOFF

The LOGOFF command exits the dul utility and terminates the Vantage session. It is synonymous with the END and QUIT commands.

For more information see [END](#).

## LOGON

The LOGON command establishes a Vantage session.

---

### Note:

This describes the standard TD 2 (Teradata authentication) logon format. For more information about other logon formats and types of authentication see *Teradata Vantage™ - Advanced SQL Engine Security Administration*, B035-1100.

---

### Batch Mode Syntax

```
.LOGON [ tdpid/ ] username, password [, 'acctid' ] [;]
```

### Interactive Mode Syntax

```
.LOGON [ tdpid/ ] username [;]
```

### Syntax Elements

#### *tdpid*

Identifier which is associated with a particular database.

The default identifier is TDP0.

#### *username*

ID of the user on the corresponding database.

The maximum length of username is 30 characters.

#### *password*

Password associated with the username.

The maximum length of a password is 30 characters.

---

### Note:

In interactive mode, the password is accepted on the next line in a protected area.

---

#### *acctid*

Account identifier associated with the username. The acctid can contain up to 30 characters.

Each doubled apostrophe, if any, counts as one character of the 30 characters.

## Usage Notes

- The LOGON command is the first command you type in a dul session. The command establishes a session on a database and identifies you and the account that is charged for system resources used during a dul operation. To ensure system security, the password is not displayed when dul is used interactively.
- You must have CREATE, DROP, and SELECT privileges on DBC.SW\_EVENT\_LOG, as well as on the dump table on the database, to perform a load or unload operation.
- After you log on to Vantage, dul changes the default database to crashdumps. The crashdumps database is where the database initially saves a dump. However, you can change the default database with the DATABASE command.
- For additional information on granting privileges, see *Teradata Vantage™ - Database Administration*, B035-1093. For more information on the crashdumps database, see [DATABASE](#).
- Logon with Teradata Wallet

Teradata Wallet is a secure store for client credentials. It stores passwords for Vantage authentication, which can automatically be retrieved at run time and submitted to the database. Teradata Wallet must be configured with usernames and passwords before it can be used for logons using DUL. For more information on configuring Teradata Wallet, see *Teradata Vantage™ - Advanced SQL Engine Security Administration*, B035-1100.

## Example: Batch Mode Logon to Vantage

To log onto a dDatabase with a tdpid of TDP0, a username of Admin, and a password of abc, type the following:

```
LOGON TDP0/Admin,abc ;
```

The following appears:

```
*** Logon successfully completed.
*** Teradata Database Release is 14.00.00
*** Teradata Database Version is 14.00.00
*** Transaction Semantics are BTET.
*** Character Set Name is 'ASCII'.
*** Changing current database to crashdumps.
*** New default database accepted.
*** File Compression Enabled.
*** MultiFile option ON.
*** NoSpool option OFF.
```

## Example: Interactive Mode Logon to Vantage

To log onto a database with a tpid of crashdumps, a username of crashdumps, and a password of crashdumps, type the following:

1. Type your logon and press **Enter**.



For example:

```
.logon crashdumps
```

Your logon is repeated, and you are prompted for your password:

```
.logon crashdumps
Password:
```

2. Type your password and press **Enter**.

For example:

```
crashdumps
```

Your password will not appear on screen.

The following appears:

```
*** Logon successfully completed.
*** Teradata Database Release is 14.00.00
*** Teradata Database Version is 14.00.00
*** Transaction Semantics are BTET.
*** Character Set Name is 'ASCII'.
*** Changing current database to crashdumps.
*** New default database accepted.
*** File Compression Enabled.
*** MultiFile option ON.
*** NoSpool option OFF.
Dump Unload/Load - Enter your command:
```

### Example: Batch Mode logon using Teradata Wallet

To log on to Teradata system tdpid01, using the username and password string created by Teradata Wallet, enter the following:

```
logon tdpid01/$tdwallet(dultuser),
```

The following appears:

```
*** Logon successfully completed.
*** Teradata Database Release is 14.00.00.00
*** Teradata Database Version is 14.00.00.00
*** Transaction Semantics are BTET.
*** Character Set Name is 'ASCII'.
*** Changing current database to crashdumps.
*** New default database accepted.
*** File Compression Enabled.
```

```
*** MultiFile option OFF.  
*** NoSpool option OFF.
```

### **Example: Interactive Mode logon using Teradata Wallet**

To log on to Teradata system tdpid02 using the Vantage username and password for Teradata Wallet user dultuser, type the following at the interactive prompt:

```
logon tdpid02/$tdwallet(dultuser);
```

At the Password prompt enter:

```
$tdwallet(dultuser)
```

The following appears:

```
*** Logon successfully completed.  
*** Teradata Database Release is 14.00.00.00  
*** Teradata Database Version is 14.00.00.00  
*** Transaction Semantics are BTET.  
*** Character Set Name is 'ASCII'.  
*** Changing current database to crashdumps.  
*** New default database accepted.  
*** File Compression Enabled.  
*** MultiFile option ON.  
*** NoSpool option OFF.
```

## MULTIFILE

The MULTIFILE command instructs dul to generate or read from multiple split files of defined size during an UNLOAD or LOAD operation, respectively.

### Syntax

```
MULTIFILE { ON [ n ] | OFF }
```

### Syntax Elements

#### OFF

Multiple split files are not generated as part of UNLOAD operation. By default, unloaded information is not split into multiple files. Use MULTIFILE OFF if you have previously used MULTIFILE ON, and no longer want the unloaded information split into multiple files.

#### ON

Multiple split files are generated as part of UNLOAD operation. The split files are numbered as FN, FN1, FN2, FN3... Multiple split files are considered as part of LOAD operation if they exist in a numbered fashion as above (FN, FN1, FN2, FN3...).

#### *n*

An optional argument which defines the maximum size in megabytes of the split files that should be created during UNLOAD operation. N is a valid integer from 1 through 2000. If n is not specified with MULTIFILE ON command, by default 2000 is assumed.

### Usage Notes

- The MULTIFILE command is helpful when the unloaded file size is extremely large. It causes the information to be split into several smaller files, which can be more easily transmitted to the Teradata Support Center.
- When using MULTIFILE ON command, the split files that are generated during an UNLOAD operation could be of different sizes, but will always be less than the maximum that has been set (either via use of the *n* option or default 2000 MB).

### Example: Using MULTIFILE to split large amounts of crashdump data

To load the operation:

```
Dump Unload/Load - Enter your command:
.LOAD crash_20000606_143623_01 FILE = single_unload.gz;
.LOAD crash_20000606_143623_01 FILE = single_unload.gz;
**** Creating table 'crash_20000606_143623_01'.
**** Table has been created.
```

```

**** Loading data into 'crash_20000606_143623_01'
**** Logging on Amp sessions.
**** Growing Buffer to 4118
**** Starting Row 10000 at Sun Jul 26 22:58:12 2015

**** Starting Row 20000 at Sun Jul 26 22:58:12 2015

**** Starting Row 30000 at Sun Jul 26 22:58:12 2015

**** END LOADING phase...Please stand by...

Loading data into crash_20000606_143623_01 completes successfully.

```

To UNLOAD operation using MULTIFILE ON 2 (meaning split files' maximum size would be 2 MB):

```

Dump Unload/Load - Enter your command:
.MULTIFILE ON 2
.MULTIFILE ON 2
*** Multi-file flag is ON
Maximum split file size is 2097152

Dump Unload/Load - Enter your command:
.UNLOAD crash_20000606_143623_01 FILE = multi_unload;
.UNLOAD crash_20000606_143623_01 FILE = multi_unload;
*** Logging on Amp sessions.
*** All processors selected ...
*** Returned data consists of 2098 blocks
*** Unloading data from crash_20000606_143623_01
the number of blocks received 1000

the node number is 30720
the instigating node is 33
the time the error occurred is Tue Jun 23 06:23:30 2015

the event is 6649854, severity is 15 and category 0
Severity = (15)
Category = None

the number of blocks received 2000
*** Unloading Crashdumps table completed.
*** The number of blocks unloaded: 2098

Split files after the UNLOAD operation:
-----

```

```
multi_unload.gz 2070423
multi_unload1.gz 2068977
multi_unload2.gz 2069766
multi_unload3.gz 2072556
multi_unload4.gz 1287700
```

To load back with multiple split files:

```
Dump Unload/Load - Enter your command:
.MULTIFILE ON
.MULTIFILE ON
*** Multi-File flag is ON
Maximum split file size is 2097152000 bytes

Dump Unload/Load - Enter your command:
.LOAD testreload FILE = multi_unload;
.LOAD testreland FILE = multi_unload;

*** Creating table 'testreload'.
*** Table has been created.
*** Loading data into 'testreload'
*** Logging on Amp sessions.
*** Growing Buffer to 4118
*** Starting Row 10000 at Sun Jul 26 23:05:05 2015

*** Starting Row 20000 at Sun Jul 26 23:05:06 2015

*** Starting Row 30000 at Sun Jul 26 23:05:07 2015

*** END LOADING phase ... Please stand by ...
Loading data into testreload completes successfully.
```

## NOSPOOL

Uses the Teradata NO SPOOL feature to unload a crashdump faster.

### Syntax

```
NOSPOOL { ON | OFF } ;
```

### Syntax Elements

#### ON

NOSPOOL is enabled

#### OFF

NOSPOOL is disabled

### Usage Notes

NOSPOOL is disabled by default, so must be explicitly enabled before using the UNLOAD command. NOSPOOL remains enabled until explicitly changed, or until the dul session is ended.

When NOSPOOL is enabled, the number of blocks received is displayed only when the unload operation has completed.

### Example: Using the NOSPOOL command in dul

To enable and disable the feature:

```
NOSPOOL ON;  
*** NoSpool flag is ON
```

```
NOSPOOL OFF;  
*** NoSpool flag is OFF
```

## .OS

The .OS command submits an operating system command to your host during a dul session.

### Syntax

```
.OS oscommand [;]
```

### Syntax Elements

#### *oscommand*

Command that is legal on your host operating system.

### Usage Notes

At the prompt below, type one of the following, depending on your system:

Dump Unload/Load - Enter your command:

Type:

```
.os date
```

The following appears:

```
.os date  
Mon Aug 3 13:00:24 EDT 2015
```

## QUIT

The QUIT command exits the dul utility and terminates the Vantage session. It is synonymous with the END and LOGOFF commands.

For more information see [END](#).



## SEE

The SEE command provides a summary of the processors and error codes captured in a dump.

The information, which you retrieve from the dump table, helps you confirm system failure information forwarded to the Teradata Support Center.

Information about tables with the `_L`, `_C`, and `_M` suffixes is not returned.

## Syntax

```
SEE [ database. ] table ;
```

## Syntax Elements

### *database*

Database in which the table resides.

### *table*

Table that contains dump information.

## Example: Displaying the processors and error codes from a dump

To access a specific dump table, type the following:

```
see crashdumps.Crash_20000412_194517_02;
```

The following appears:

```
*** Looking at crashdumps.Crash_20000412_194517_02.
*** Query completed. One row found. 7 columns returned.

the node number is 1024
the instigating node is 1024
the time the error occurred is Wed April 12 19:45:17 2000
the event is 12140, severity is 40 and category 10
  Severity = UserError
  Category = User
```

## SELECT

The SELECT command sets criteria determining the dump data selected for a load or unload operation.

### Syntax

```
SELECT {
  ALL |
  ERROR |
  RESET |
  PROC nod-num [...]
}
[ ERRORDATE { 'yymmdd' | 'yyyymmdd' | "yymmdd" | "yyyymmdd" } ] ;
```

### Syntax Elements

#### ALL

Load or unload operation select dump data from all processors. This is the default.

#### ERROR

#### RESET

Load or unload operation.

Only processors with an error code are selected.

#### PROC *nod-num*

Unload operation.

*nod-num* specifies the number of the virtual processor to be selected. Dump data will be loaded or unloaded only from this node. Valid values for *nod-num* are 30720 and greater.

#### *yymmdd*

#### *yyyymmdd*

Error date to be selected from the DBC.SW\_EVENT\_LOG table.

*yy* or *yyyy* is the year, *mm* is the month, and *dd* is the day. For example, August 6, 2019 is 20200806 or 190806.

Dul retrieves all error records with a date equal or later than *yymmdd* or *yyyymmdd*.

### Usage Notes

You must always type the SELECT command before you type an UNLOAD command. If you do not specify a SELECT command, the entire dump is selected.

Negative processor numbers are used for special record types. When RESET is specified, dul retrieves dump data with negative processor numbers.

**Example: Selecting dump data for load or unload**

To select dump data, type the following:

```
select proc 16384;
```

The following appears:

```
*** Processor selection set to list of processors.
```

## SHOW CONTROLS

The SHOW CONTROLS command displays the user-configurable settings.

The user-configurable settings are:

- File Compression option
- NoSpool option
- Multifile option
- Current default database

### Syntax

```
SHOW { CONTROLS | CONTROL } ;
```

### Example: Displaying the values of user-configurable settings

To display the current dul settings:

```
.SHOW CONTROLS
```

The following appears:

```
*** File Compression Enabled.  
*** NoSpool option OFF.  
*** Multifile option ON.  
Maximum split file size is 2097152000 bytes.  
*** Current default database is crashdumps.
```

## SHOW VERSIONS

The SHOW VERSIONS command displays the current level of all dul utility software modules.

This command is helpful in reporting software problems.

### Syntax

```
SHOW { VERSIONS | VERSION } ;
```

### Example: Displaying the version of dul software

```
.SHOW VERSIONS;  
Dump Unload/Load Version 15.10.00.00 for Linux running Socket TCP/IP  
DULMain      : 15.00.00.00  
DULCAP       : 15.00.00.00  
MOSIDEP      : 15.10.00.02  
CLIV2        : 15.10.00.16  
MTDP         : 15.10.00.07  
OSENCRYPT    : N/A  
OSERR        : 15.10.00.00  
MOSIos       : 15.10.00.02
```

## UNLOAD

The UNLOAD command moves dump data from a system-generated table on the database to a file on a host or directly to removable media.

### Syntax

```
UNLOAD [ database. ] table FILE = filepath [ f ] ;
```

### Syntax Elements

#### *database*

The name of the database in which the table resides and the separator between the database name and table name. Use a period (.) to separate the database name from the table name.

The default database is crashdumps.

#### *table*

The name of the table that contains the dump data.

#### *filepath*

The path of the file into which the dump data is unloaded.

*filepath* is specified as *directory/filename*.

---

#### Note:

Dul saves files in a compressed, gzip format, and appends a file extension of .gz to the file name automatically, if the specified name does not include .gz.

---

#### *f*

An option for unloading dumps from a foreign database.

### Usage Notes

Before you can perform a dump/unload operation, you must have the following privileges:

- CREATE, DROP, and SELECT on the tables in the crashdumps database
- SELECT on the DBC.SW\_EVENT\_LOG system table

The UNLOAD command selects dump data according to the selection criteria specified in a previous SELECT command. If you do not specify a SELECT command, the entire dump table is searched. You should always type a SELECT command before an UNLOAD command.

Before you perform an unload operation, dul displays summary information, such as which processors are selected and error dates about the selection criteria that is set. After the load operation, dul displays event codes, if any exist, for the specified processors. For information on setting selection criteria, see [SELECT](#).

In moving dump data, the \_C, \_L, and \_M files are always included. If the \_C or \_L table is not found, it is generated.

If you load the unloaded data back into some other Teradata system, the table created by the LOAD command is called dump data in a foreign database and will be one of the following tables:

- tname\_C
- tname\_L
- tname\_M

When you enter the UNLOAD command without the F option, dul unloads the data specified in *tname* from the DBC.SW\_EVENT\_LOG table.

The F option is required to unload the dump data from the foreign database. Dul unloads data from the table specified in the UNLOAD command and the corresponding tname\_C, tname\_L, and tname\_M tables.

Dul uses the FastExport utility to improve transfer speed for unloading dumps. However, you can still unload dumps using BTEQ.

To use the UNLOAD command, you must have CREATE, DROP, and SELECT privileges on the following:

- Dump table (Crashdumps.Crash\_YYYYMMDD\_HHMMSS\_NN)
- System table DBC.SW\_EVENT\_LOG

For more information about the FastExport utility, see *Teradata® FastExport Reference*, B035-2410.

For information about BTEQ, see *Basic Teradata® Query Reference*, B035-2414.

For additional information on access privileges, see *Teradata Vantage™ - Database Administration*, B035-1093.

### **Example: Unloading dump data from Vantage tables**

The following example assumes that a system failure occurred on 6/6/00 at 2:36 p.m. To confirm that a dump table has been created, you would examine the crashdumps databases using the following command:

```
HELP DATABASE crashdumps ;
```

Since dump tables are named according to the date and time that the system failure occurred, you should be able to find the correct dump table. In this example, the table named crash\_20000606\_143623\_01 contains the dump information.

To display the contents of the dump table, use the SEE command:

```
SEE crash_20000606_143623_01 ;
```

The SEE command displays summary information about all of the processors and error codes that were captured in the dump. Some of the processors might not contain any information. Generally, only processors that contain errors are needed for an unload operation.

By typing a SELECT command next, you can choose the processors for the unload operation. To select only the processors that contain error codes, type the following command:

```
SELECT ERROR ;
```

Dul responds with this message:

```
*** Processor selection set to list of processors.
```

Now you are ready to unload the dump data from table crash\_20000606\_143623\_01 on the crashdumps database onto your host using the following command:

```
UNLOAD crash_20000606_143623_01 file=filepath;
```

Dul responds with these messages:

```
*** Unloading data from crash_20000606_143623_01 for processor(s) 1-6.
```

```
*** Query completed. 263 rows found. 3 columns returned.
```

```
*** Processor 1-6
```

```
*** Number of rows = 263
```

```
*** Unloading Procedure Information
```

```
*** Query completed. 200 rows found. 4 columns returned.
```

```
*** Number of rows = 200
```

```
*** Unloading Errorlog Information
```

```
*** Query completed. 413 rows found. 6 columns returned.
```

```
*** Number of rows = 413
```

```
*** Unloading Memo Information
```

```
*** Query completed. 123 rows found. 5 columns returned.
```

```
*** Number of rows = 123
```

```
Event = 2490:
```

```
On 6/6/96 at 09:26:27 in processor 1-6, partition 14,  
task SEMTSK.
```

```
Severity = UserError
```

```
Category = User
```

```
HostEvent = None
```

```
*** Number of rows = 263
```



---

**Note:**

If the dump data is copied to the local hard disk on your host, you must copy the data to a removable medium for shipment.

---

## Ferret Utility (ferret)

---

**Note:**

The utility described is not available for as-a-service customers.

---

The Ferret utility, `ferret`, lets you display and set storage space utilization attributes of the Advanced SQL Engine. Ferret dynamically reconfigures the data in the Teradata file system while maintaining data integrity during changes. Ferret works on various data levels as necessary: vproc, map, table, subtable, WAL log, disk, and cylinder.

### Runs From

- Database Window or comparable interface to the Teradata console subsystem, such as `cnstern`
  - Teradata Viewpoint Remote Console portlet
- 

**Note:**

Ferret cannot be used when Vantage is in Debug-Stop state.

---

For general information on starting the utilities from different interfaces, see [Starting the Utilities](#). For information on Viewpoint, see *Teradata® Viewpoint User Guide*, B035-2206.

### Redirecting Input and Output

When you first start Ferret, it accepts input from STDIN, the console input file, and directs output to STDOUT, the console output file.

By using the Ferret INPUT and OUTPUT commands, you can redirect the Ferret input and output data to files that you specify. Also, you can do the following:

- Append Ferret data output to an existing output file
- Overwrite an existing output file
- Write only to a new file
- Display the current output file

When you direct output to a file with the OUTPUT command, Ferret echoes all input and diagnostic messages to that file. For more information, see [INPUT](#) and [OUTPUT](#).

## The Teradata File System

The Teradata File System is not a general-purpose file system. It helps isolate the database from hardware platform dependencies, and supports the creation and maintenance of database tables under the direction of Vantage.

The file system stores data in physical rows. A physical row is a general data structure that consists of a row header (metadata that includes a unique row ID) and stored data. Each physical row stores one of several kinds of data. For example:

- A table row
- A series of one or more column partition values
- A table header (metadata for a table)
- Index data structures

This list is not comprehensive; physical rows also store other types of data used by Vantage.

To the file system, the kind of data stored in the physical row is immaterial. The interpretation and differentiation is made by the higher-level database software that requests and receives the data from the file system.

In the context of this file system discussion, the term *row* generally refers to a physical row in the file system, irrespective of the kind of data the physical row stores. It may store data that corresponds to a row in a database table or other data.

In the Teradata File System, a data block (DB) is a disk-resident structure that contains one or more physical rows from the same table. Any single physical row is fully contained within a DB, and every DB is fully contained within a cylinder. A cylinder is a unit of contiguous physical memory that is allocated as a single unit.

At the beginning of each cylinder is a cylinder index (CI) that identifies the data blocks (DBs) the cylinder contains. The Master Index (MI) is a per-AMP global structure that is the top-level file system internal index structure. It is periodically written to disk as a *snapshot* to speed up system startup times. If the snapshot copy of the MI has problems, the system can reconstruct the MI by inspecting all CIs allocated to an AMP, however, this takes significantly more time than reading the MI from a snapshot.

For more information about data blocks, cylinders, and the physical row layouts for different data structures, see *Teradata Vantage™ - Database Design*, B035-1094.

## Write Ahead Logging (WAL)

WAL is a log-based file system recovery scheme in which modifications to permanent data are written to a log file, the WAL log. The log file contains change records (Redo records) which represent the updates. At key moments, such as transaction commit, the WAL log is forced to disk. In the case of a reset or crash, Redo records can be used to transform the old copy of a permanent data block on disk into the version that existed in memory at the time of the reset.

By maintaining the WAL log, the permanent data blocks that were modified no longer have to be written to disk as each block is modified. Only the Redo records in the WAL log must be written to disk. This allows a write cache of permanent data blocks to be maintained.

WAL protects all permanent tables and all system tables but is not used to protect the Transient Journal (TJ), since TJ records are stored in the WAL log. WAL also is not used to protect spool or volatile tables.

The WAL log is maintained as a separate logical file system from the normal table area. Whole cylinders are allocated to the WAL log, and it has its own index structure.

The WAL log data is a sequence of WAL log records and includes the following:

- Redo records, used for updating disk blocks and insuring file system consistency during restarts.
- TJ records used for transaction rollback.

The WAL log is subject to many of the same problems as the normal file system, such as data corruption from hardware, software, and operational problems. Teradata Services personnel can display and modify the WAL log and its index, and can repair such problems.

## Ferret Command Syntax

```
ferret_command [ ; ... ]
```

***ferret\_command***

```
cmd [ ; ... ] [ cmdoptions | /dispt | /Y ] [ parameter ]
```

### Syntax Elements

***cmd***

A Ferret command from [Ferret Commands](#).

***cmdoptions***

Options specific to *cmd*.

***dispt***

A letter that specifies the amount of display information: S (short), M (medium), or L (long).

This option is available only for the SHOWBLOCKS, SHOWCOMPRESS, SHOWDEFAULTS, SHOWSPACE, and SHOWWHERE commands.

**Y**

The following:

- If the option is used, Ferret does not prompt for confirmation of the command.

- If the option is omitted for a command where confirmation is required, Ferret prompts for confirmation before executing the command.

This confirmation option is not available for all commands. Only specific commands allow you to use it. If a command takes /Y, the command does not take any other options.

### ***parameter***

Parameters are included only with specific commands. Types of parameters include multitoken parameters and decimal or hexadecimal numeric input.

## **Usage Notes**

Although several command options are specific to a particular Ferret command, the following command usage rules apply to Ferret commands:

- A space is required between the following:
  - *cmdoption* and *parameter*
  - *cmd* and *parameter*, if you do not specify *cmdoption* or if *cmdoption* is at the end of the command
- You can combine multiple Ferret commands on a single command line, as shown below:

```
command ; command ;
```

- Ferret is case insensitive, which means that you can enter any command, keyword, or option in uppercase, lowercase, or mixed-case letters.
- If you end a command line with a backward slash (\), the command line continues on the next line.
- If an error occurs during processing of a string of commands, processing stops because the latter commands usually depend on the correct processing of commands entered earlier.
- If an error occurs while parsing the command line, you can type a single question mark (?) as the next command. Ferret displays a list of alternatives expected by the parser at that point of the syntax error. If you type another single question mark (?) after Ferret displays the alternatives, Ferret returns the HELP text for the command in question. For example:

```
Ferret ==>radix in dex
radix in de<-Syntax error->x

Ferret ==> ?
Valid input at the point of error is:
;                end of command

Ferret ==> ?
RADIX [ ( IN/PUT | OUT/PUT ) ] [ ( H/EX | D/EC ) ]

Ferret ==> ?
RADIX [ ( IN/PUT | OUT/PUT ) ] [ ( H/EX | D/EC ) ]
```

Sets the Flags for how to treat Unqualified numbers. Either Hex (base 16) or Decimal (Base 10), respectively. See HELP NUMBER for a description of unqualified INPUT. The initial setting of these Flags is HEX. If neither INPUT nor OUTPUT is specified the command applies to both Flags. If neither HEX nor DEC is specified, the current setting of the Flag is displayed.

```
Ferret ==> ?
```

```
No more information available. Use HELP /L
```

- Comments are allowed anywhere a blank is allowed. Enclose comments between braces { }. If Ferret does not find a closing brace ( } ) on a line, Ferret interprets the rest of the line as a comment. For example:

```
Ferret ==> SCANDISK db 0 2 1fa { this is the broken data block
```

- If you use a single question mark (?) in place of a legal syntactic element, Ferret informs you as to what you can type at that point in the command. For example:

```
Ferret==> output ?
```

```
Valid input at the ? is :
```

```
,          INTO          OVER          TO          ON          ;
end of command
```

Other examples of using a single question mark (?) in place of a legal syntactic element are shown below:

```
output ?{comment}
output {comment} ?
```

## Using Ferret Parameters

The variable, *parameter*, in the Ferret syntax diagram includes various types of parameters, including multitoken parameters and decimal and hexadecimal numeric input.

### Multitoken Parameters

Multitoken parameters, such as subtable identifiers, which are typed as three values, are typed on a single line with one or more spaces separating the individual tokens. You can also enter multitoken parameters separated by hyphens.

For example, either of the following as subtable identifiers is acceptable:

```
0 200 0
0-200-0
```

You can separate parameters from each other and from the command by spaces or a comma. To specify a command option (*cmdoption* ), type it on the same line as the command (*cmd*).

## Numeric Input

Numeric values can be entered in either decimal or hexadecimal format. The default numeric base for data input to and output depends on the radix settings for input and output:

- When the radix for input is decimal, numeric input is interpreted as decimal numbers. For example, input of 45 is interpreted as the value 45.
- When the radix for input is hexadecimal, numeric input is interpreted as hexadecimal numbers. For example, input of 45 is interpreted as the value 69.

The initial radix setting is hexadecimal for input and output. For more information on setting the radix, see the documentation for the RADIX command.

The following special numeric formatting conventions can be used to force entered numeric values to be interpreted as decimal or hexadecimal, regardless of the radix setting:

- Decimal values can be signified by adding a 0i or i prefix, or by adding a decimal point suffix:

```
0i45 45. 0I45
```

- Hexadecimal values can be signified by adding a 0x or x prefix, or by adding an h suffix. Leading zeros are optional:

```
0x2D X2D 2Dh 002DH
```

The valid range of for numeric values is unsigned 16-bit values, 0 through 65535 (0x0 through 0xFFFF), except when patching and using /L or /W for 32-bit integers, in which case the valid range is 0 through 4,294,967,295 (0x0 through 0xFFFFFFFF).

You must separate two numbers on the same line from each other by a space or a comma.

## Specifying a Subtable Identifier (tid)

Commands operate on subtables, rather than on tables. A table is a collection of subtables, and a subtable is a collection of physical rows. Each subtable has physical rows that store one particular type of data. For example, primary and fallback data subtables have data rows, primary and fallback index subtables have index rows, and the physical rows of table header subtables contain table metadata.

Each subtable is identified by a subtable identifier ( *tid* ). The tid has two parts. The first part identifies the table to which the subtable belongs. All subtables that are part of the same table have the same table identifier as part of their tid. The second part identifies one of the subtables that comprise the table, and indicate the subtable type.

A *tid* is defined as follows:

```
{ table_unique | = } type_and_index
```

*table\_unique* and *type\_and\_index* together uniquely identify a subtable.

Syntax Element	Description
<i>table_unique</i>	<p>Unique identifier of the table to be acted upon.</p> <p>The table can be uniquely identified in either of two ways:</p> <ol style="list-style-type: none"> <li>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: <ul style="list-style-type: none"> <li>" <i>database_name</i> . <i>table_name</i> "</li> <li>" <i>database_name</i> ." <i>table_name</i> "</li> <li>' <i>database_name</i> . <i>table_name</i> '</li> <li>' <i>database_name</i> .' <i>table_name</i> '</li> </ul> </li> <li>Specify the unique numeric identifier of the table, which consists of two numeric values separated by a space. The table identifier consists of the first two numeric values returned by the TABLEID command. These two numbers are common to all subtables that comprise the table.</li> </ol> <p><b>Note:</b></p> <p>The format of the input numbers depends on the current radix setting, which is displayed by the RADIX command.</p>
=	Specifies to use the most recently saved value for <i>table_unique</i> .
<i>type_and_index</i>	Uniquely identifies the subtable within the group of subtables that comprise a table. The <i>type_and_index</i> identifies both the type of subtable, and the individual subtable itself.

The syntax for *type\_and\_index* is as follows:

```
{ type [ / index [ / variant ] ] | number }
```

Syntax Element	Description												
<i>type</i>	<p>The type of subtable.</p> <p><i>type</i> can be one of the following:</p> <table> <tr> <th>Type</th><th>Description</th></tr> <tr> <td>*</td><td>All the subtables of this table</td></tr> <tr> <td>H</td><td>The table header subtable. Because table header subtables have only one row and no secondary indexes or work subtables, any index and variant specifications that follow an H type are ignored.</td></tr> <tr> <td>P</td><td>The primary data subtable.</td></tr> <tr> <td>F <i>number</i></td><td>The fallback data subtable specified by <i>number</i>. The default is 1.</td></tr> <tr> <td>F *</td><td>All of the fallback subtables.</td></tr> </table>	Type	Description	*	All the subtables of this table	H	The table header subtable. Because table header subtables have only one row and no secondary indexes or work subtables, any index and variant specifications that follow an H type are ignored.	P	The primary data subtable.	F <i>number</i>	The fallback data subtable specified by <i>number</i> . The default is 1.	F *	All of the fallback subtables.
Type	Description												
*	All the subtables of this table												
H	The table header subtable. Because table header subtables have only one row and no secondary indexes or work subtables, any index and variant specifications that follow an H type are ignored.												
P	The primary data subtable.												
F <i>number</i>	The fallback data subtable specified by <i>number</i> . The default is 1.												
F *	All of the fallback subtables.												



Syntax Element	Description												
<i>index</i>	<p>The index subtable to examine. If an index subtable value is not provided, it defaults to the data subtable. Index 1 is the first secondary index. index 2 is the second secondary index, and so forth.</p> <p><i>index</i> can be one of the following:</p> <table> <tr> <th><i>index</i> Value</th><th>Description</th></tr> <tr> <td><i>number</i></td><td> <p>A secondary index that can be used instead of specifying X <i>number</i> . if you enter <i>number</i> by itself, it must be a multiple of 4, and it is interpreted in the following ways:</p> <ul style="list-style-type: none"> <li>• 0 is the same as entering X0</li> <li>• 4 is the same as entering X1</li> <li>• 8 is the same as entering X2</li> <li>• 12 is the same as entering X3</li> </ul> <p>and so forth.</p> </td></tr> <tr> <td>*</td><td>All the indexes of the table.</td></tr> <tr> <td>D</td><td>The primary data index (same as X0 or 0).</td></tr> <tr> <td>X <i>number</i></td><td>The secondary index specified by <i>number</i>. The default is 1.</td></tr> <tr> <td>X *</td><td>All the secondary indexes, starting at 0.</td></tr> </table>	<i>index</i> Value	Description	<i>number</i>	<p>A secondary index that can be used instead of specifying X <i>number</i> . if you enter <i>number</i> by itself, it must be a multiple of 4, and it is interpreted in the following ways:</p> <ul style="list-style-type: none"> <li>• 0 is the same as entering X0</li> <li>• 4 is the same as entering X1</li> <li>• 8 is the same as entering X2</li> <li>• 12 is the same as entering X3</li> </ul> <p>and so forth.</p>	*	All the indexes of the table.	D	The primary data index (same as X0 or 0).	X <i>number</i>	The secondary index specified by <i>number</i> . The default is 1.	X *	All the secondary indexes, starting at 0.
<i>index</i> Value	Description												
<i>number</i>	<p>A secondary index that can be used instead of specifying X <i>number</i> . if you enter <i>number</i> by itself, it must be a multiple of 4, and it is interpreted in the following ways:</p> <ul style="list-style-type: none"> <li>• 0 is the same as entering X0</li> <li>• 4 is the same as entering X1</li> <li>• 8 is the same as entering X2</li> <li>• 12 is the same as entering X3</li> </ul> <p>and so forth.</p>												
*	All the indexes of the table.												
D	The primary data index (same as X0 or 0).												
X <i>number</i>	The secondary index specified by <i>number</i> . The default is 1.												
X *	All the secondary indexes, starting at 0.												
<i>variant</i>	<p>The possible subtables.</p> <p><i>variant</i> can be one of the following:</p> <table> <tr> <th><i>variant</i> Value</th><th>Description</th></tr> <tr> <td>*</td><td>All of the possible variant subtables.</td></tr> <tr> <td>0</td><td>The default used if a <i>variant</i> value is not specified.</td></tr> <tr> <td>1</td><td>The value during a Sort or Table Modify operation.</td></tr> <tr> <td>2</td><td>Unused.</td></tr> <tr> <td>3</td><td>Unused.</td></tr> </table>	<i>variant</i> Value	Description	*	All of the possible variant subtables.	0	The default used if a <i>variant</i> value is not specified.	1	The value during a Sort or Table Modify operation.	2	Unused.	3	Unused.
<i>variant</i> Value	Description												
*	All of the possible variant subtables.												
0	The default used if a <i>variant</i> value is not specified.												
1	The value during a Sort or Table Modify operation.												
2	Unused.												
3	Unused.												
<i>number</i>	<p>A single number that represents internally the type of subtable (header, primary data, or fallback data), the index to use when ordering the physical rows (primary data index or one of the secondary indexes), and the variant. The table header has a <i>type_and_index</i> number of zero. Examples of these numbers are listed below.</p>												

The following table gives examples that describe the *type\_and\_index* fields.

Subtable Description	Number	Type	Type/Index	Type/Index/ Variant
Table header	0	H		
Primary data subtable	1024 (0x0400)	P	P/D	

Subtable Description	Number	Type	Type/Index	Type/Index/ Variant
First secondary index	1028 (0x0404)		P/X1	
First fallback table	2048 (0x0800)	F1	F1/D	
Second secondary index of the third fallback table	4104 (0x1008)		F3/X2	
All primary subtables			P/*	
All primary secondary indexes			P/X*	
All fallback subtables		F*		
All subtables of this table		*		
Sort table of the first secondary index	1029 (0x0405)			P/X1/1
Both tables during a sort of the first secondary index				P/X1/ *

For example, assume that table T4 is a table in database MYDB and has a *table\_number* of 0 1198. Also assume that input is accepted in hexadecimal format.

Some valid specifications of a *tid* for primary subtables of table T4 are as follows:

- "MYDB.T4" 400
- "MYDB.T4" 1024.
- "MYDB"."T4" P
- u 'MYDB'.'T4' 400 h
- 0 1198 400
- 0 1198 P

## Classes of Tables

Vantage differentiates among the following five classes of tables. You can specify a table classification when defining the SCOPE parameters of an action.

Table Type	Description
Permanent Journal (PJ) Tables	Tables that can survive system restarts, but do not contain user-visible data. Each PJ table contains data generated internally by Vantage. The PJ data is usually used to restore the journaled tables to a given checkpointed state by rolling transactions forward or backwards from an archived copy of the tables.
Permanent Tables	Tables containing the real data, which can survive system restarts.
Temporary Tables	Tables that can exist as either global or volatile temporary tables as defined below: <ul style="list-style-type: none"> <li>• Global temporary tables exist only during the duration of the SQL session in which they are used.</li> </ul>

Table Type	Description
	<ul style="list-style-type: none"> <li>• Volatile temporary tables reside in memory and do not survive a system restart. They are treated like spool tables, discussed below.</li> </ul>
Spool Tables	<p>Tables that contain non-permanent data and can be divided into classes according to their scope of persistence.</p> <p>Intermediate result spool tables hold temporary results during the processing of a single SQL query and persist only for the duration of that processing. Response spool tables hold the final answer set from a query and a limited number can optionally persist across further queries in the same session. Spool tables can be discarded as follows:</p> <ul style="list-style-type: none"> <li>• Normally, when they are no longer needed.</li> <li>• As part of a specific resource cleanup on a transaction abort or session logoff.</li> <li>• As part of a general resource cleanup every time the system restarts.</li> <li>• Rows for volatile tables are placed in spool spaces and are discarded at the end of a transaction or at the end of a session (depending on a table option or by a DROP TABLE statement).</li> <li>• Volatile table definitions reside in memory and do not survive a system restart.</li> </ul>
Persistent Spool Tables	These are treated like spool tables, but they survive system restarts.

The attributes associated with each class of tables can affect system performance, since the attributes are set individually, and each class of tables is used for a different purpose.

For example, you might want to pack only Permanent and PJ tables. Therefore, you would specify these tables when defining the SCOPE of the PACKDISK command.

## Rows and Row Ranges

Every physical row is uniquely identified by a row ID. A row ID specification, also referred to as a *ridspec* or a *rowspec*, is a value that includes a partition number, rowhash or hash bucket number, and uniqueness value. It can occupy up to 16 bytes in a physical row:

- The partition number is stored as 2 bytes if the number of partitions defined for the table is up to 65535, or 8 bytes if the number is greater. If the table is not partitioned, no space is reserved in the row ID for a partition specification. A value of zero means the table is not partitioned.
- The specification of the remainder of the row ID depends on whether the table includes a primary index:
  - PI tables have a 4-byte rowhash value and a 4-byte uniqueness value, which is used to impose uniqueness in cases of rowhash synonyms for different physical rows.
  - Tables without a primary index (NoPI tables) and tables with a primary AMP index (PA tables) have a 16- or 20-bit hash bucket value and a 44-bit uniqueness value. For systems using 16-bit hash bucket values, the four bits between the hash bucket value and the uniqueness value are unused.

**Note:**

The remainder of the RowID is treated as two 4-byte values, and present them as hash and uniq fields:

```
hash0 hash1 uniq0 uniq1
-----
FFA9 CB78 0000 0002
```

For NoPI and PA tables displayed or manipulated, hash0 and hash1 actually represent a 20-bit hash bucket value and the 12 high-order bits of the uniqueness value, while uniq0 and uniq1 display the 32 low-order bits of the uniqueness value. For more information on NoPI and PA tables, see *Teradata Vantage™ - SQL Data Definition Language Syntax and Examples*, B035-1144.

A range of rows, also called a *ridrange*, is specified by the ridspec of the starting and ending rows:

```
start_rid_spec [ TO end_rid_spec ]
```

For purposes of input, the required and optional components of a ridspec depend on whether the table containing the physical row is partitioned.

Table Type	Ridspec Syntax
partitioned	<i>partitioned_ridspec</i> : { row_partition   = } [ { row_hash   = } [ row_unique   = ] ]
nonpartitioned	<i>nonpartitioned_ridspec</i> : { <i>partitioned_ridspec</i>   [ row_partition   = ] { row_hash   = } [ row_unique   = ] }

Syntax Element	Description
<i>row_partition</i>	<p>The partition number of the physical row. The partition number is a single numeric value. It is the internal partition number used by the file system. The value must be between 0 and 9223372036854775807 (0 and 7FFFFFFFFFFFFFFFFF in hexadecimal notation).</p> <p>For tables with multiple levels of partitioning, each physical partition results from the combination of all partitioning expressions in the PARTITION BY clause for a specific set of values in the partitioning columns. This gives a single, unique combined partition number, which is mapped to an internal partition number. Although the combined partition number of a partition can change if partition ranges are dropped or added for the partition level, the internal partition number does not change.</p>

Syntax Element	Description
	For rows in nonpartitioned tables, the partition number must be zero, if specified, or can be omitted from the <i>ridspec</i> .
<i>row_hash</i>	<p>The row hash value of the physical row. The row hash determines the AMP to which the physical row is distributed. The row hash consists of 2 numeric values, usually shown in output screens under column headings <i>hash0</i> and <i>hash1</i>.</p> <p>For <i>start_rid_spec</i> :</p> <ul style="list-style-type: none"> <li><i>row_hash</i> defaults to 0x0000 0x0000.</li> </ul> <p>For <i>end_rid_spec</i> :</p> <ul style="list-style-type: none"> <li><i>row_hash</i> defaults to the value of <i>row_hash</i> specified in <i>start_rid_spec</i> . If no value for <i>row_hash</i> was specified in <i>start_rid_spec</i> , <i>row_hash</i> in <i>end_rid_spec</i> defaults to 0xFFFF 0xFFFF.</li> </ul>
<i>row_uniq</i>	<p>The system-generated uniqueness value which is used with the row hash value to uniquely identify a physical row. The uniqueness value consists of two numeric values, usually shown in output screens under column headings <i>u0</i> and <i>u1</i>.</p> <p>For <i>start_rid_spec</i> :</p> <ul style="list-style-type: none"> <li>If <i>row_uniq</i> is omitted, and no <i>end_rid_spec</i> is specified, all the rows in <i>row_hash</i> are selected.</li> <li>If <i>row_uniq</i> is omitted, and an <i>end_rid_spec</i> is specified, <i>row_uniq</i> defaults 0x0000 0x0000.</li> </ul> <p>For <i>end_rid_spec</i> :</p> <ul style="list-style-type: none"> <li><i>row_uniq</i> defaults to the value of <i>row_uniq</i> specified in <i>start_rid_spec</i> . If no value for <i>row_uniq</i> was specified in <i>start_rid_spec</i> , <i>row_uniq</i> in <i>end_rid_spec</i> defaults to 0xFFFF 0xFFFF.</li> </ul>
=	<p>The program will use the most recently saved values for <i>row_partition</i>, <i>row_hash</i>, or <i>row_uniq</i>. The most recently specified values for <i>row_partition</i> , <i>row_hash</i> , and <i>row_uniq</i> are stored. Separate values are stored for <i>start_rid_spec</i> and <i>end_rid_spec</i> . <i>row_partition</i> , <i>row_hash</i> , or <i>row_uniq</i> . An error is reported if a saved <i>start_rid_spec</i> or <i>end_rid_spec</i> value is used for a partitioned table, but the table referenced in the current command is not partitioned, and vice versa.</p>

For more information on the Teradata File System and physical rows, see [The Teradata File System](#).

For more information on table row structure, see *Teradata Vantage™ - Database Design*, B035-1094.

## Vproc Numbers

In Ferret, the vproc number (*vproc\_number*) is used in the SCOPE command to specify one AMP or a range of AMPs for which the utility performs an action, such as reconfiguration or disk space display.

Valid AMP vprocs have numbers in the range of 0 through 16199.

## Map Names

A map includes one or more AMP vprocs. If you specify a map name in the Ferret SCOPE command, the name must be enclosed with double quotation marks. The scope is effectively the same as if you had scoped Ferret to the individual vprocs that are included in the specified map.

## Ferret Commands

Ferret gives you a wide range of commands that allow you to display specific information about the Teradata system, and optimize the system to improve performance. Use Ferret commands to:

- Define the level or scope of a command, such as one or more tables, vprocs, maps, cylinders, or the WAL log.
- Display the parameters and scope of a previous action.
- Perform an action to increase performance, such as moving data to reconfigure data blocks and cylinders.
- Displaying utilization of storage space by percentage and cylinders.

### Defining Command Parameters Using SCOPE

Some Ferret commands require that you first define the parameters, or scope, of the action you want to initiate.

The SCOPE command allows you to limit the command action in the following areas:

SCOPE Parameter	Variables
VPROC	<ul style="list-style-type: none"> <li>• A single AMP vproc</li> <li>• A range of AMP vprocs</li> <li>• All AMP vprocs in a configuration</li> </ul>
MAP	<ul style="list-style-type: none"> <li>• The AMP vproc or range of AMP vprocs in a single contiguous map</li> <li>• The AMP vprocs in several contiguous maps</li> <li>• All maps, and therefore all AMP vprocs in a configuration</li> </ul>
CYLINDER	<ul style="list-style-type: none"> <li>• A single cylinder</li> <li>• A set of cylinders</li> </ul>
TABLE	<ul style="list-style-type: none"> <li>• A single table</li> <li>• A set of tables</li> <li>• All tables in the system</li> <li>• A class of tables</li> </ul>
WAL	The WAL log

For more information, see [SCOPE](#).

### Summary of Ferret Commands

The following table summarizes the Ferret commands, their valid scopes, and gives a brief description of the function of each command:

Command	Valid SCOPE Options	Function
<a href="#">COMPRESS</a>	Not Applicable	Compresses data blocks of a specified table, or shows an estimate of what results would be if tables were compressed.
<a href="#">COMPRESS</a>	None	Displays the current system day, date, and time.
<a href="#">DEFRAGMENT</a>	Vprocs or tables	Combines free sectors, and moves them to the end of a cylinder.
<a href="#">DISABLE</a>	None	Sets a specific flag in the file system to FALSE, disabling certain features of Ferret. Most such flags are for internal use only.
<a href="#">ENABLE</a>	None	Sets a specific flag in the file system to TRUE, enabling certain features of Ferret. Most such flags are for internal use only.
<a href="#">ERRORS</a>	None	Redirects diagnostic messages to a file that you specify or to the default file STDERR. Using the ERRORS command, you can append an existing message file, overwrite an existing message file, write only to a new file, or display the current diagnostic message file.
<a href="#">FORCE</a>	None	Sets the cylinders occupied by a table or range of rows, to a specified temperature, regardless of how frequently the data has been historically accessed.
<a href="#">HELP</a>	None	Provides general help for Ferret or detailed help if you specify an option or parameter.
<a href="#">INPUT</a>	None	Informs Ferret to read commands from a specified file rather than from the default input file STDIN.
<a href="#">MAP</a>	None	Displays information about a specified map.
<a href="#">OUTPUT</a>	None	Redirects Ferret output to a file you specify or to the default file STDOUT.
<a href="#">PACKDISK</a>	Vprocs or tables	Reconfigures cylinders within a defined scope.
<a href="#">PRIORITY</a>	None	Sets the priority class of the Ferret process.
<a href="#">QUIT</a>	None	Ends a Ferret session.
<a href="#">RADIX</a>	None	Sets the default radix used as the numeric base for Ferret data input and output.
<a href="#">RESETBLCTSKCNT</a>	None	Resets the count of running COMPRESS and UNCOMPRESS tasks to zero.
<a href="#">RESETBLCTSKCNT</a>	WAL log, vprocs, or tables	Performs a verification of the file system B-Tree structure.

Command	Valid SCOPE Options	Function
<a href="#">SCOPE</a>	All	Defines the scope for subsequent COMPRESS, DEFAGMENT, PACKDISK, SCANDISK, SHOWAMPRECOVERY, SHOWBLOCKS, SHOWCOMPRESS, SHOWCYLALLOC, SHOWFSP, SHOWSPACE, SHOWWHERE, and UNCOMPRESS commands.
<a href="#">SHOWAMPRECOVERYBLC</a>		Displays which compress and uncompress operations that happen during system recovery to minimize transaction recovery time after a restart.
<a href="#">SHOWBLOCKS</a>	WAL log, vprocs, or tables	Displays data block size and the number of rows per data block for a defined scope. Also indicates whether data blocks use block-level compression.
<a href="#">SHOWCOMPRESS</a>	Vprocs, and PERMANENT and JRNL tables	Lists subtables that have data blocks using block-level compression.
<a href="#">SHOWDEFAULTS</a>	None	Displays the current default radix for input and output, the current input, output, and error file names, and the current scope settings.
<a href="#">SHOWFSP</a>	Vprocs or tables	Displays table names and space utilization for those tables that would free or consume some number of cylinders if PACKDISK is executed at a particular FSP. The scope can include one or more tables, one or more vprocs, or the entire system.
<a href="#">SHOWSPACE</a>	WAL log, vprocs, or tables	Displays storage space utilization for permanent, spool, WAL log, temporary, and journal data, including the amount of free space remaining.
<a href="#">SHOWSPACE</a>	Vprocs, tables, or WAL log	Displays information about cylinder allocation and temperature for cylinders in the currently set scope.
<a href="#">TABLEID</a>	None	Displays the table number of the specified table when given the database name and table name.
<a href="#">UNCOMPRESS</a>	Not Applicable	Uncompresses data blocks of a specified table, or shows an estimate of what results would be if tables were uncompressed.

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
- SHOWCYLALLOC



## **Ferret Error Messages**

All Ferret error messages are directed by default to your system console screen.

Ferret and file system error messages can be redirected through use of the ERRORS command.

For more information on file system messages, see *Teradata Vantage™ - Database Messages*, B035-1096.

## BLCINFO

Displays the expected initial compression status of data in rows added to empty tables for specified combinations of block-level compression (BLC) settings.

### Note:

This command displays the initial subtable compression status that would result from different combinations of compression settings. It is for planning purposes only, and does not actually change any compression settings on the system.

Compression status is listed for four types of subtables:

- Primary data subtable
- Fallback data subtable
- Primary CLOB data subtable
- Fallback CLOB data subtable

### Syntax

```
{ BLCINFO | BLCI } {
    [ALL] { BLOCKCOMPRESSIONTLA | BLOCK } =
        { AUTOTEMP | MANUAL | ALWAYS | NEVER | DEFAULT }
        { DEFAULTTABLEMODE | DEFAULT } =
        { AUTOTEMP | MANUAL | ALWAYS | NEVER | USEDDBSCONTROL } |

    { FLOWDIAGRAM | FLOW } |

    { ALGORITHM | ALGO }
}
```

### Syntax Elements

#### BLOCKCOMPRESSIONTLA BLOCK

Specifies a valid setting of the BLOCKCOMPRESSION table level attribute. The table level attribute is the setting applied when the table was created using CREATE TABLE or altered using ALTER TABLE. For more information on this CREATE TABLE and ALTER TABLE option, see *Teradata Vantage™ - SQL Data Definition Language Syntax and Examples*, B035-1144.

**DEFAULTTABLEMODE****DEFAULT**

Specifies a valid setting of DefaultTableMode in DBS Control. For more information on DefaultTableMode, see the description of DBS Control in *Teradata Vantage™ - Database Utilities*, B035-1102.

USEDBSCONTROL means to use the value of DefaultTableMode that is currently set in DBS Control.

**ALL**

Given the selected options for BLOCKCOMPRESSIONTLA and DEFAULTTABLEMODE, shows which subtable data will be compressed for all combinations of the remaining compression settings and query bands.

**FLOWDIAGRAM****FLOW**

Displays a flow diagram that describes the decisions that determine whether a subtable is compressed or uncompressed based on several BLC settings.

**ALGORITHM****ALGO**

Displays the pseudo code that describes the algorithm used to determine whether a subtable is compressed or uncompressed based on several BLC settings.

**Usage Notes**

The command prompts for DBS Control and query band settings as necessary to describe a unique set of BLC conditions.

Compression status of each of the four subtable types is indicated by a C for compressed or U for uncompressed.

**Note:**

The listed compression status represents only the initial compression status of data added to an empty table. If temperature-based BLC (TBBLC) is enabled, the compression status can change with time for tables with a BLOCKCOMPRESSION value of AUTOTEMP.

The BLOCKCOMPRESSION = AUTOTEMP table level attribute can be explicitly specified when a table is created or altered. If DefaultTableMode in DBS Control is set to AUTOTEMP, BLOCKCOMPRESSION is automatically set to AUTOTEMP for tables that do not otherwise specify a value for that attribute. For AUTOTEMP tables, data is automatically compressed or uncompressed based on the frequency of data access.

**Example: Using BLCINFO to show expected compression status of newly loaded data**

The following example shows that, given the specified combination of DBS Control compression settings, and using a Query Band of BLOCKCOMPRESSION = FALLBACK when data is loaded into a new, empty table that was created with table-level attribute of BLOCKCOMPRESSION = MANUAL, all primary data (regular and CLOB) will be loaded into the table in a non-compressed state, but all fallback data for the table will be compressed.

```
Ferret ==>
blcinfo blockcompressiontla=manual defaulttablemode=manual
```

```
Enter BLOCKCOMPRESSION Query Band:
```

1. DEFAULT
  2. YES
  3. NO
  4. ALL
  5. NONE
  6. FALLBACK
  7. ONLYCLOBS
  8. WITHOUTCLOBS
  9. FALLBACKANDCLOBS
- ```
> 6
```

```
DBSControl Compression Group
```

```
Enter CompressPermFallbackDBs:
```

1. UNLESSQBNO
  2. NEVER
  3. ONLYIFQBYES
- ```
> 1
```

```
Enter CompressPermFallbackCLOBDBs:
```

1. UNLESSQBNO
  2. NEVER
  3. ONLYIFQBYES
- ```
> 1
```

```
-----
| Query Band | DBSControl Tunables |
| Primary | Fallback |
-----
| BLOCKCOMPRESSION | PrimaryDBs | PrimaryCLOBDBs | FallbackDBs | FallbackCLOBDBs |
| Base | CLOB | Base | CLOB |
-----
| FALLBACK | * | * | UNLESSQBNO | UNLESSQBNO |
| U | U | C | C |
-----
```

- C - This subtable will load compressed
- U - This subtable will load uncompressed

- The DBSControl Compression group field BlockLevelCompression must be set to ON otherwise no compression will take place.

- Fallback secondary indexes follow the compression state of the Fallback base subtable. Primary subtable secondary indexes are not eligible for compression.

- The result only applies to Permanent tables. To get the results for Global Temporary tables substitute CompressPerm\_\_DBs

with CompressGlobalTemp\_\_DBs. Where \_\_ stands for Primary, Fallback, PrimaryCLOBs, FallbackCLOBs.

## COMPRESS

Compresses specified types of data in specified databases and tables, or shows an estimate of what results would be if tables were compressed. If data of one of the specified types is subsequently added to the specified table or database, it will also be compressed.

Compression for a table uses the algorithm and level specified by the BLOCKCOMPRESSIONALGORITHM and BLOCKCOMPRESSIONLEVEL options that were specified in the CREATE TABLE or ALTER TABLE statement. If either of these options has not been specified for the table, or if the value was specified as DEFAULT, the data is compressed using the algorithm and level specified by the [CompressionAlgorithm](#) and [CompressionLevel](#) settings in DBS Control.

---

### Note:

BLOCKCOMPRESSIONLEVEL is effective only for the ZLIB compression algorithm.

---

Compression uses the algorithm specified by the CompressionAlgorithm setting in DBS Control.

### Syntax

```
COMPRESS [ /Y ] table_unique [
  PRIMARY |
  FALLBACK |
  FALLBACKANDCLOBS |
  { WITHOUT | ONLY } CLOBS
]
[ ESTIMATE | E ]
```

### Syntax Elements

#### /Y

Prevents Ferret from displaying a confirmation message.

#### *table\_unique*

Unique identifier of the table for which data blocks will be compressed.

The table to be compressed 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'
  - "database\_name.\*"

- 'database\_name.\*'
- "database\_name".\*\*"
- 'database\_name'.\*'

The asterisk denotes all tables in the given database.

- Specify the unique numeric identifier of the table, which consists of two numeric values separated by a space. The table identifier consists of 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 following options specify the type of data to compress:

**PRIMARY**

All primary base table data and primary LOB data that is eligible for compression within the specified database or table.

**FALLBACK**

All fallback base table data and fallback LOB data that is eligible for compression within the specified database or table.

**FALLBACKANDCLOBS**

All fallback base table data, fallback LOB data that is eligible for compression, and primary LOB data eligible for compression within the specified database or table.

**WITHOUT CLOBS**

All data except LOB data that is eligible for compression within the specified database or table.

**ONLY CLOBS**

Only the primary and fallback LOB data that is eligible for compression within the specified table. Other table DBs are not affected.

The following option does not compress any data:

**ESTIMATE****E**

Estimates data block (DB) sizes that would exist after a compress operation is performed, and estimates the CPU usage (time per DB) that would be required. Does not compress any data.

Multiply the Table Size Factor shown in the output by the current table size to determine the estimated table size after a compress operation.

**Note:**

Estimates are most accurate when the table DB size is at or near the size defined by DATABLOCKSIZE (optionally specified when the table was created or altered), or defined by the PermDBSize setting in DBS Control if no DATABLOCKSIZE was specified for the table.

The Ferret INQUIRE (or INQ) and ABORT commands can be used to check on the progress or halt this operation during command execution.

The command progress percentage reported by the INQUIRE command can change unexpectedly if the table being compressed is being modified concurrently by another process.

**Usage Notes**

The COMPRESS command is part of the block-level Compression (BLC) feature of Vantage.

BLC enables data compression at the data block (DB) level of the Teradata file system. Compression reduces the amount of storage required for a given amount of data. The BlockLevelCompression field of DBS Control enables and disables BLC.

Tables that have their BLOCKCOMPRESSION value set to NEVER in the CREATE TABLE or ALTER TABLE statement cannot be compressed using the COMPRESS command.

BLC is subject to the compression-related DBS Control settings. Some data blocks in a compressed table may not be compressed for the following reasons:

- The block consists of fewer sectors than the number specified by MinDBSectsToCompress.
- The block size after compression would not be reduced by the percentage specified by MinPercentCompReduction.
- The DBS Control compression setting for the specified table type is set to NEVER compress.

After a data block has been compressed using a particular algorithm, the data retains that compression, even if the algorithm is subsequently changed. New data added to the table is compressed using the new algorithm. If previously compressed data is updated, which requires uncompressing and recompressing the data, the currently set algorithm is used for the recompression.

Consequently, a single subtable can exhibit a mixture of uncompressed data blocks and data blocks compressed using different compression algorithms, depending on the settings at the time the data was added to the table. To make the data consistent with respect to compression, use the Ferret COMPRESS



command to compress all the data using the currently set algorithm, or use the Ferret UNCOMPRESS command to uncompress all the table data.

The PermDBSize and JournalDBSize field settings in DBS Control apply to the uncompressed size of these types of DBs.

Ferret logs the beginning and end of COMPRESS operations. If the database is reset during the operation, the compression operation continues as part of the recovery process after the database restarts. However, compression will continue and complete only on the table that was in the process of being compressed at the time of the reset. The rest of the database will remain in the state it was prior to the reset.

Only one instance of the COMPRESS command is allowed to run at any time on the system.

If you receive an error when you issue the COMPRESS or UNCOMPRESS command, and you are certain there is no other instance of either command currently running, use the RESETBLCTSKCNT command to reset the count of running COMPRESS and UNCOMPRESS tasks to zero.

## Related Information

| For more information on...       | See...                                                                               |
|----------------------------------|--------------------------------------------------------------------------------------|
| DBS Control compression settings | <a href="#">DBS Control (dbscontrol)</a> .                                           |
| Query Banding                    | <i>Teradata Vantage™ - SQL Data Definition Language Detailed Topics</i> , B035-1184. |
| RADIX command                    | <a href="#">RADIX</a> .                                                              |
| RESETBLCTSKCNT command           | <a href="#">RESETBLCTSKCNT</a> .                                                     |
| TABLEID command                  | <a href="#">TABLEID</a> .                                                            |
| UNCOMPRESS command               | <a href="#">UNCOMPRESS</a> .                                                         |

## DATE/TIME

The DATE and TIME commands are synonyms. Both show the same display of current system day, date, and time.

### Syntax

```
{ DATE | TIME }
```

### Usage Notes

The DATE or TIME command shows a timestamp in the following format:

*DDD mmm dd, yyyy HH:MM:SS*

| Format              | Description                                                               |
|---------------------|---------------------------------------------------------------------------|
| <i>DDD</i>          | Abbreviated name of the day of the week.                                  |
| <i>mmm dd, yyyy</i> | Abbreviated name of the calendar month, numeric day, and four-digit year. |
| <i>HH:MM:SS</i>     | Current system time in hour, minutes, and seconds.                        |

### Examples: DATE and TIME command outputs

```
> date
date
```

```
Thu Jul 02, 2016 18:17:05
```

```
> time
time
```

```
Thu Jul 02, 2016 18:22:14
```

## DEFRAGMENT

---

### Note:

Because of the difficulty in determining how severely a cylinder is fragmented, only trained personnel should use this command.

---

The DEFRAGMENT command causes free sectors on each qualifying logical cylinder contained within the current scope to be combined together into a single contiguous block of free space on that cylinder. The scope used is designated by the SCOPE command or the default if no SCOPE command was issued.

### Syntax

```
{ DEFRAGMENT | DEFRAG } [ FORCE ] [ /Y ]
```

### Syntax Elements

#### FORCE

To defragment every cylinder whether a cylinder meets the criteria specified in Usage Notes below.

#### /Y

To bypass the confirmation prompt that asks if you are sure you want to run this command.

The Ferret ABORT command can be used to halt this operation during command execution. After it is initiated, ABORT stops the defragmenting and reports the current status of the vprocs.

---

### Note:

Up to 30 seconds can elapse before the ABORT operation begins.

---

### Usage Notes

A cylinder becomes fragmented as a result of modifications to table data. As data blocks (DBs) are created, deleted, or change size as a result of these modifications, what started out as a small number of large free spaces on the newly allocated cylinder becomes a large number of small free spaces. Although there may still be a large total number of free sectors on the cylinder, with time there are fewer ranges of contiguous free sectors that are large enough to store new DBs. When new DBs cannot be stored on the cylinder, new free cylinders must be found and allocated.

The defragmentation process moves the data blocks closer together, coalescing the free space into areas that are large enough that new DBs can be allocated on the cylinder. This can reduce the need to allocate free cylinders for new data storage. The defragmentation process actually moves DBs from the fragmented cylinder to a new cylinder, where the DBs can be packed together, leaving a single, large set of consecutive free sectors. The cylinder from which the DBs were moved is freed, and made available for new allocations.

Consequently, defragmentation is a costly operation, so should be performed only when system resources are plentiful.

Unless the FORCE option is used, the DEFRAGMENT command performs defragmentation of a logical cylinder within the scope only if all of the following criteria are met.

- More than one set of free sectors is on the cylinder, so there is something that could be combined.
- 25% or more free sectors are on the cylinder.
- The average number of sectors comprising a set of free sectors is less than the average size of a data block on that cylinder.

The FORCE option causes the DEFRAGMENT command to defragment a logical cylinder within the scope if it has more than one block of free space. In that case, it will defragment a cylinder regardless of the percentage of free sectors or average block sizes.

Note that the DEFRAGMENT operation does not change the size or content of any data blocks. Existing blocks may be moved, but they are otherwise unchanged.

For each AMP, the Ferret indicates how many cylinders it will attempt to defragment. For example:

```
vproc 2 response
Requested Cylinder Range had 197 cylinders placed on the defrag list
```

Upon completion, for each amp, the actual number of cylinders defragmented will be logged into the system's message log. For example:

```
002507 14:47:07 043100d6 ... 44 8
340516600|appl|1|S|I|U|0|0|M|0|0|PDE|0|0|0|1#PDElogd: Event number 34-05166-00
(severity 10, category 10)
5166: Defragment of cylinder[s] occurred.

On Wed Jun 22 14:47:07 2016 on NODE 001-01, VPROC 2, partition 9, task
fsubackgrnd
197 cylinder[s] defragmented
```

Defragmentation proceeds as a background task, and can continue to run even after the Ferret prompt returns. To determine when defragmentation has completed, check DBC.SW\_Event\_Log for event\_tag=34-05166-00.

Vantage can isolate some file system errors to a specific data or index subtable, or to a contiguous range of rows ("region") in a data or index subtable. In these cases, Vantage marks only the affected subtable or region down. This improves system performance and availability by allowing transactions that do not require access to the down subtable or rows to proceed, without causing a database crash that would require a system restart.

If DEFRAGMENT encounters down regions, it skips these regions, and displays the percentage of total space that was skipped.

**Example: DEFAGMENT command output**

The following example shows the output that DEFAGMENT generates:

```
Ferret ==>
defrag /y
Defrag has been sent to all AMP vprocs in the SCOPE.
Type 'ABORT' to stop them before completion

vproc 0 response
Requested Cylinder Range had 5 cylinders placed on the defrag list

vproc 1 response
Requested Cylinder Range had 9 cylinders placed on the defrag list
```

## DISABLE

The DISABLE command sets a specified flag in the file system to FALSE. This disables certain features of Ferret.

### Syntax

```
{ DISABLE | DISA } { flag | ? } [,...]
```

### Syntax Elements

#### *flag*

Specifies the flag that will be set to false.

The following flags are available:

- SCRIPT, SCRIPT MODE, and SCRIPTMODE

Script mode is disabled by default, and should be explicitly enabled before running scripts that call Ferret. When script mode is enabled, the Ferret ABORT and INQUIRE commands are disabled.

The following flags are for internal use only, and should not be disabled or enabled:

- CHECK and CHECKTS

#### ?

Displays a complete list of available flags.

### Example: Disabling Ferret Script Mode

An example is shown below:

```
Ferret ==>
disable script
```

## ENABLE

The ENABLE command sets a specified flag in the file system to TRUE. This enables certain features of Ferret.

### Syntax

```
{ ENABLE | ENA } { flag | ? } [,...]
```

### Syntax Elements

#### *flag*

Specifies the flag that will be set to true. The following equivalent flags are available:

SCRIPT, SCRIPT MODE, and SCRIPTMODE

Script mode is disabled by default, and should be explicitly enabled before running scripts that call Ferret. When script mode is enabled, the Ferret ABORT and INQUIRE commands are disabled.

#### ?

Displays a complete list of available flags.

### Example: Enabling Ferret Script Mode

An example is shown below:

```
Ferret ==>
enable script
```

## ERRORS

The ERRORS command redirects diagnostic messages to a file that you specify or to the default file STDERR. If nothing is specified, the name of the files where errors are directed displays.

### Syntax

```
ERRORS [ { TO | INTO | OVER } { file | STDERR | ME } ]
```

### Syntax Elements

#### TO

That Ferret is to write diagnostic messages to a new file or to STDERR.

If the file specified exists, Ferret returns an error.

#### INTO

That Ferret is to append diagnostic messages to a specified file or to STDERR.

If the file already exists, Ferret appends the error messages to the end of the file.

If the file does not exist, Ferret creates the file automatically.

#### OVER

That Ferret is to overwrite an existing file or STDERR with current diagnostic messages.

If the file already exists, Ferret writes over the file.

If the file does not exist, Ferret creates the file.

#### *file*

The name of the destination file for diagnostic messages.

#### STDERR

The default file to which Ferret writes diagnostic messages.

#### ME

The synonym for STDERR.

### Usage Notes

When you start Ferret, it writes diagnostic messages to STDERR by default.

You can use the ERRORS command to redirect the diagnostic messages in the following ways:

- To write to a new file only
- To append to an existing file



- To overwrite an existing file
- To display on your console

If you include the *file* parameter in the ERRORS command, the file you specify becomes the destination for diagnostic messages redirected from STDERR.

If you type the ERRORS command without any options, Ferret shows the name of the current diagnostic messages file STDERR on your system console.

**Example: Redirecting Ferret diagnostic and error messages to a file**

The following example shows the format for redirecting diagnostic messages into a specific file and directory:

```
Ferret ==>  
Errors over /home/user1/Ferret.error
```

## 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.

## HELP

The HELP command provides general help for Ferret or detailed help if you specify an option or parameter.

### Syntax

```
{ HELP | H } [ /L ] [ ALL | keyword | ? ]
```

#### /L

Long form of a help display, showing parameter descriptions in addition to command syntax.

If you do not type /L, Ferret assumes the short form of the help display, which shows command syntax only.

#### ALL

Available help for all Ferret commands and parameters.

This is the default.

#### *keyword*

Ferret command or parameter name.

#### ?

A list of all keywords for which help is available.

### Example: Ferret HELP command

The following is an example of output that HELP generates:

```
Ferret ==>
Help /L Radix
RADIX [ ( IN/PUT | OUT/PUT ) ] [ ( H/EX | D/EC ) ]
Sets the Flags for how to treat Unqualified numbers. Either Hex (base 16) or Decimal (Base 10), respectively. See HELP NUMBER for a description of unqualified INPUT. The initial setting of these Flags is HEX. If neither INPUT nor OUTPUT is specified, the command applies to both Flags. If neither HEX nor DEC is specified, the current setting of the Flag is displayed.
```

## INPUT

The INPUT command informs Ferret to read commands from a file you specify rather than from the default input file STDIN.

### Syntax

```
{ INPUT | IN } FROM file
```

#### *file*

The name of a file you specify as the source of command input to Ferret.

### Usage Notes

When you first start Ferret, it accepts input from the STDIN file by default. Using the INPUT command, you can redirect input from any file you specify.

#### Note:

The input file must be located on the node that runs the control AMP. To determine which node is running the control AMP, use the STATUS command of the Vproc Manager utility. The control AMP is designated under the Logical Configuration with an asterisk.

When Ferret reaches the end of the file you specify, Ferret again accepts input from you, if the file does not instruct Ferret to quit.

You can nest the INPUT command inside command files to a maximum of nine files deep.

#### Example: Directing Ferret to read input commands from a file

The following command example shows the format for redirecting input from a file in a specific directory:

```
Ferret ==>
Input from /home/user1/commands
```

## MAP

Provides information about a specified contiguous or sparse map.

Maps determine how table (and join index table) data is distributed among AMPs. You can assign a map to a table when you create or alter a table. Maps can be either contiguous or sparse. For more information about maps, see *Teradata Vantage™ - SQL Data Definition Language Syntax and Examples*, B035-1144, *Teradata Vantage™ - Database Design*, B035-1094, and *Teradata Vantage™ - Database Administration*, B035-1093.

### Syntax

```
MAP "mapname"
```

### Syntax Elements

**mapname**

The name of the map for which you want information.

### Usage Notes

The map information includes:

- whether the map is contiguous or sparse
- if the map is contiguous, whether it is the system default contiguous map
- the map number (indicated as MAP ID in the display)
- if the map is sparse, the base (parent) contiguous map name
- range of AMPs in the contiguous or base contiguous map

The AMPs in the map (or, for a sparse map, the AMPs in the parent map) store the rows of tables and join indexes that use the map.

### Examples: MAP command

The following examples demonstrate the MAP command output for contiguous and sparse maps.

```
==>
    >map "SparseMap1"
    map "SparseMap1"

    Map Name   : SPARSEMAP1
    Map Type   : Sparse
    Map ID     : 1046 (0x416)
    Base Contiguous Map Name : TD_MAP1
    Contiguous Map AMP Range : 0-99
```

```
==>
```

```
> map "td_map1"
```

```
map "td_map1"
```

```
Map Name   : TD_MAP1 - The Default Map
```

```
Map Type   : Contiguous
```

```
Map ID     : 1025 (0x401)
```

```
Contiguous Map AMP Range : 0-99
```

```
==>
```

```
> map "td_map4"
```

```
map "td_map4"
```

```
Map Name   : TD_Map4
```

```
Map Type   : Contiguous
```

```
Map ID     : 370 (0x0172)
```

```
Contiguous Map AMP Range : 50-99
```



## OUTPUT

The OUTPUT command redirects Ferret output to a file you specify or to the default file STDOUT.

### Syntax

```
{ OUTPUT | OUT }
  [ { TO | OVER | INTO } { file | STDOUT | ME } ]
```

### Syntax Elements

#### TO

That Ferret is to redirect output to a new file or to STDOUT.

If the file exists, Ferret returns an error.

#### INTO

That Ferret is to append output to an existing file specified by *file* or to STDOUT.

If the file exists, Ferret appends the output to the end of the file.

If the file does not exist, Ferret creates the file.

#### OVER

That Ferret is to overwrite an existing file or STDOUT with new Ferret output.

If the file exists, Ferret writes over the file.

If the file does not exist, Ferret creates the file.

#### *file*

Name of the file you specify as the destination of Ferret output.

#### STDOUT

Default file to which Ferret writes output.

#### ME

Synonym for STDOUT.

### Usage Notes

When you start Ferret, diagnostic messages are written to STDOUT by default. You can use the OUTPUT command to redirect Ferret output in any of the following ways:

- To write to a new file only
- To append an existing file

- To overwrite an existing file
- To display on your console

---

**Note:**

The output file is located on the node that runs the control AMP. To determine which node is running the control AMP, use the STATUS command of the Vproc Manager utility. The control AMP is designated under the Logical Configuration with an asterisk.

---

When Ferret redirects output to a file, all input and diagnostic messages are echoed to the output file as well as to their usual destinations.

If you include the *file* parameter in the OUTPUT command, Ferret uses that parameter as the destination for output redirected from STDOUT.

If you type the OUTPUT command without any options, Ferret displays the name of the current output file STDOUT to your system console.

**Example: Redirecting Ferret output to a file**

The following command example shows the redirecting of Ferret output into a specific file:

```
Ferret ==>  
Output into /home/user1/output.file
```

## PACKDISK

The PACKDISK command arranges (packs and unpacks) data on the storage associated with the tables in the current scope, leaving a specified percentage, the free space percent (FSP), of the storage empty. This allows for table growth within the currently allocated storage, and can free up cylinders for new storage allocations.

### Syntax

```
PACKDISK
[ /Y ]
[ { FREESPACEPERCENT | FREE | FSP } [=] number ]
[ FORCE ]
```

### Syntax Elements

#### /Y

To bypass user confirmation.

If you do not specify the /Y option, then Ferret requests for user confirmation before proceeding with PACKDISK.

#### FREESPACEPERCENT

#### FREE

#### FSP

Optionally specifies the percentage of storage space that PACKDISK should leave unoccupied on cylinders for tables in the current scope. This free space percent (FSP) allows tables to grow in place, deferring the need for allocation of new free storage cylinders.

Because read-only tables do not grow, they do not require free space.

The FSP for a table can also be specified when the table is created or modified using CREATE TABLE and ALTER TABLE statements. If defined for a table, this FSP value takes precedence over the PACKDISK FREESPACEPERCENT value, unless the FORCE option is used.

If FREESPACEPERCENT is not specified with PACKDISK, and no FSP was specified with CREATE TABLE or ALTER TABLE, PACKDISK uses the free space percent value defined by the FreeSpacePercent setting in DBS Control.

#### *number*

Percentage of storage space to be left free in the cylinders of tables in the current scope.

**FORCE**

Forces PACKDISK to pack table data to the FREESPACEPERCENT value, regardless of current FSP values that might have been set using CREATE TABLE or ALTER TABLE statements.

**Note:**

For tables that have an FSP set with CREATE TABLE or ALTER TABLE, results from using the FORCE option of PACKDISK are only temporary. When the periodic background AutoCylPack task runs, it returns table free space to the FSP value set using CREATE TABLE or ALTER TABLE statements. IF FORCE is not used, PACKDISK honors the current FSP values that have been explicitly set using CREATE TABLE or ALTER TABLE. For tables with no explicit FSP, PACKDISK uses the system default defined in DBS Control.

**Usage Notes****General**

PACKDISK packs cylinders for tables in the currently defined Ferret scope. For more information on the scope of Ferret commands, see [SCOPE](#).

Packing applies to cylinders, not to the space in individual data blocks within those cylinders. Data block sizes remain the same after the PACKDISK operation.

Vantage can isolate some file system errors to a specific data or index subtable, or to a contiguous range of rows ("region") in a data or index subtable. In these cases, Vantage marks only the affected subtable or region down. This improves system performance and availability by allowing transactions that do not require access to the down subtable or rows to proceed, without causing a database crash that would require a system restart.

If PACKDISK encounters down regions, it skips these regions, and displays the percentage of total space that was skipped.

The Ferret INQUIRE (or INQ) and ABORT commands can be used to check on the progress or halt this operation during command execution.

**Note:**

Up to 30 seconds can elapse before the ABORT operation begins.

## Precedence of FSP Settings

FSP specifies the minimum amount free space that should be left on the cylinder during a data load operation. This reserves space for future table updates. Because read-only tables do not grow, they do not require free space.

Because FSP can be specified in a number of places, the amount of space that PACKDISK leaves free is determined in the following way:

1. If the PACKDISK FREESPACEPERCENT option is used with the FORCE option, PACKDISK packs to the specified FREESPACEPERCENT. See also [PACKDISK](#).
2. If the PACKDISK FORCE option is not used, the FSP specified when the table was created or modified using the CREATE TABLE or ALTER TABLE statement takes precedence.

For more information on CREATE TABLE and ALTER TABLE, see *Teradata Vantage™ - SQL Data Definition Language Syntax and Examples*, B035-1144.

3. If no FSP was specified when the table was created or modified, PACKDISK packs to the value specified by the PACKDISK FREESPACEPERCENT option.
4. If no FSP was specified when the table was created or modified, and the PACKDISK FREESPACEPERCENT option is not used, PACKDISK packs to the value of the FSP system default, specified by the FreeSpacePercent DBS Control setting.

For additional information, see [DBS Control \(dbscontrol\)](#).

## Limiting Use of the FREESPACEPERCENT Option

The FREESPACEPERCENT option of PACKDISK is not intended for routine use. It is intended for exceptional situations, such as:

- Preliminary tests scoped to a small prototype table to determine the optimal value for the FSP.
- Later tests of various FSPs on live tables that might require adjustment to ensure more optimal cylinder packing in the future.

The AutoCylPack background task runs periodically to return cylinder FSPs to the levels specified when tables were created or modified, or, if no FSP was specified for the tables, AutoCylPack returns cylinders to the system default for FSP defined in DBS Control.

If the PACKDISK FREESPACEPERCENT option is used to explicitly specify a different FSP value for a PACKDISK run, the AutoCylPack background task will, at some point, return the FSP values to those defaults.

## Selecting an Optimal Value for FSP

The optimal amount of free space to leave on cylinders is highly dependent on the individual mix of workload, tables, and applications in use by a site.

Because of the way PACKDISK packs cylinders, some cylinders might still be fragmented after the procedure completes. If this happens, use the DEFRAGMENT command to defragment the cylinder. For information, see [DEFRAGMENT](#). The scope for PACKDISK can be either VPROC or TABLE (but not both).

The SHOWFSP command helps estimate the effectiveness of PACKDISK commands by providing information regarding cylinder utilization. In particular, you can use SHOWFSP to find loosely packed tables.

The SHOWFSP command displays table names and space utilization for those tables that would free or consume some number of cylinders if PACKDISK is executed at a particular FSP. For information, see [SHOWFSP](#). The scope can include one or more tables, one or more vprocs, or the entire system.

For general guidelines for selecting optimal FSP values, see [FreeSpacePercent](#).

For additional guidelines for selecting an appropriate value for FSP, see *Teradata Vantage™ - Database Administration*, B035-1093.

## Mini Cylinder Packs

The file system automatically performs a mini cylinder pack (MiniCylPack) background operation if the number of free cylinders equals or falls below the threshold value specified for MiniCylPackLowCylProd in the DBS Control record. For additional information, see [MiniCylPackLowCylProd](#).

MiniCylPack begins by attempting to honor the FSP that was specified for a table, either when the table was created or modified using CREATE TABLE or ALTER TABLE, or the last time PACKDISK was run on the table. If the previously defined FSP does not allow enough space to be reclaimed, MiniCylPack decreases the FSP by five percent and tries again. This continues until either the necessary cylinders are freed or MiniCylPack reaches zero FSP and still no space can be freed.

When a table affected by a MiniCylPack requires the availability of free space to permit growth without requiring the addition of new cylinders, this operation can result in non-optimal system performance. For example, when many MiniCylPacks occur, they can produce a state known as thrashing. This happens because storage is packed too tightly to permit growth, so frequent allocation of new cylinders to the table space is required. Because this action removes cylinders from the available cylinder pool, more MiniCylPacks are required and an ongoing cycle of thrashing results.

AutoCylPack, a background task that periodically runs to maintain an appropriate level of free space on cylinders, helps alleviate the need for PACKDISK and MiniCylPacks, however, because all MiniCylPacks have a negative effect on system performance, you should monitor system cylinder usage.

| To monitor ...              | Do the following ...                                                                        |
|-----------------------------|---------------------------------------------------------------------------------------------|
| cylinder utilization        | Use the Ferret SHOWSPACE command.                                                           |
| occurrences of MiniCylPacks | Check the software event log (DBC.SW_Event_Log), or the Linux event log, /var/log/messages. |

Schedule PACKDISK operations as often as is necessary to prevent MiniCylPacks from occurring.

For more information about MiniCylPacks, see *Teradata Vantage™ - Database Administration*, B035-1093.

## Example: Running Ferret PACKDISK

```
Ferret ==>
packdisk

Tue Dec 08, 2009 16:10:54 : Packdisk will be started
                        On All AMP vprocs
        Do you wish to continue based upon this scope?? (Y/N)
y
Tue Dec 08, 2009 16:10:56 : Packdisk has been started
                        On All AMP vprocs

Type 'ABORT' to stop the command before completion
Type 'INQUIRE' to check on progress of command

Tue Dec 08, 2009 16:18:29 : vproc 0 response

Packdisk completed successfully. Freed up 24 (000018) large cylinders

Tue Dec 08, 2009 16:18:32 : vproc 1 response

Packdisk completed successfully. Freed up 25 (000019) large cylinders
Packdisk has completed
```

## PRIORITY

The PRIORITY command allows you to set the priority of the Ferret process.

The PRIORITY command is most commonly used with the SCANDISK and PACKDISK commands.

### Syntax

```
[ SET ] PRIORITY = priorityclass
```

### Syntax Elements

#### *priorityclass*

The priority of the Ferret process.

Valid values for *priorityclass* are:

- LOW, L, or 0
- MEDIUM, M, or 1 (This is the default.)
- HIGH, H, or 2
- RUSH, R, or 3

These values are not case sensitive.

### Example: Setting the priority of Ferret

An example is shown below:

```
Ferret ==>  
SET PRIORITY = 3
```



## QUIT

The QUIT command ends a Ferret session.

### Syntax

```
{ QUIT | Q | STOP | ST | END | EXIT }
```

## RADIX

The RADIX command sets the default radix used as the numeric base for data input to and output from Ferret as either hexadecimal or decimal. If you type just the command RADIX, the current settings of the input and output are displayed.

### Syntax

```
{ RADIX | RAD }
  [ INPUT | IN | OUTPUT | OUT ]
  [ HEX | H | DEC | D ]
```

### Syntax Elements

#### INPUT

Input to Ferret defaults to the radix you select, either hexadecimal or decimal.

#### OUTPUT

Output from Ferret defaults to the radix you select, either hexadecimal or decimal.

#### HEX

##### H

Input to or output from Ferret defaults to a radix of hexadecimal.

#### DEC

##### D

Input to or output from Ferret defaults to a radix of decimal.

### Usage Notes

When you start Ferret, the default radix for both input and output from Ferret is decimal.

If you omit both the INPUT and OUTPUT options from the RADIX command, the radix (HEX or DEC) that you select applies to both numeric input and output.

You can select either INPUT or OUTPUT, but not both when changing the settings of Ferret data.

If you omit both the HEX and DEC options, Ferret displays the current RADIX setting.

### Example: Setting the Ferret default radix to decimal

The following command example shows how to set the input to decimal:

```
Ferret ==>
Rad Input Dec
```

## RESETBLCTSKCNT

The RESETBLCTSKCNT command resets the count of the running compression or decompression tasks.

### Syntax

```
RESETBLCTSKCNT
```

### Usage Notes

Vantage maintains a count of the running COMPRESS and UNCOMPRESS commands, and allows only one such process to be running at any time. On rare occasions, such as when a COMPRESS or UNCOMPRESS operation is killed manually, the system count may be in error.

If you receive an error when you issue the COMPRESS or UNCOMPRESS command, and you are certain there is no other instance of either command currently running, use the RESETBLCTSKCNT command to reset the count of running COMPRESS and UNCOMPRESS tasks to zero.

## SCANDISK

The SCANDISK command validates the file system and reports any errors found, including discrepancies in the following items:

- Key file system data structures, such as master index (MI), current (memory) copy of the snapshot MI, cylinder indexes (CIs), data blocks (DBs), and free CIs. Additionally, SCANDISK validates the WAL Log structures: WAL master index (WMI), WAL cylinder indexes (WCIs), and WAL data blocks (WDBs).
- The RowIDs, including the partition numbers, for key physical rows should match the values stored for those rows in the cylinder index and master index.

"Physical row" means the low-level row structure that is stored by the file system. A physical row can store one of several types of data, such as table rows, the data from one or more column partitions, a table header (metadata about the table), index data structures, and other types of data. Every physical row stores only one kind of data.

- The WAL Logical Sequence Number (WLSN) for key physical TJ and REDO records should match the values stored for those records in the WAL Cylinder Index and WAL Master Index.
- Within a subtable, the internal partition number of a physical row should be greater than or equal to the internal partition number in the preceding row, if any.
- Within a subtable, either all the physical rows are partitioned (the physical rows include an internal partition number) or all the physical rows are nonpartitioned (the physical rows do not include an internal partition number).
- Within a partition, the RowID of a physical row should be greater than the RowID in the preceding row, if any.
- The calculated checksum values of key file system data structures (Cylinder Index, Data Blocks and their WAL counterparts) should match the recorded checksum values (if checksums are enabled).

SCANDISK does not validate the following items:

- The internal partition number is not validated for consistency with the result of the partitioning expression applied to the partitioning columns in a logical row, or for consistency with currently defined column partition numbers.
- The hash value in the RowID of the physical row header is not validated with the result of hashing the index columns for a primary index or primary AMP index.

You can use Ferret commands INQUIRE and ABORT to check the progress or halt this operation.

## Syntax

```
SCANDISK
[ /Y ]
[ /S | /M | /L ]
[ CI | DB | FREECIS | MI | WCI | WDB | WMI ]
[ SKIPDECOMPRESS | SKIPDECOMP ]
```

```
[ WITHSPOOL ]
[ inquire_opt ]
[ NOCR | CR ]
[ { MAXERRCNT | MAXERR } n ]
```

***inquire\_opt***

```
{ INQUIRE | INQ } [-] { NONE | number [ timeopt ] }
```

**Syntax Elements****/Y**

Prevents Ferret from displaying a confirmation message.

**/S**

Scans the MI and WMI.

**/M**

Scans the MI, CIs, WMI, and WCIs.

**/L**

Scans the MI, CIs, DBs, WMI, WCIs, and WDBs.

**CI**

Scans the MI and CIs. If the scope is AMP or all subtables, rather than selected subtables, the free CIs are also scanned.

**DB**

Scans the MI, CIs, and DBs. This is the default for the normal file system, which can be overridden by the CI, MI, or FREECIS options. If the scope is AMP or all subtables, rather than selected subtables, the free CIs are also scanned.

**FREECIS**

Scans CIs of cylinders that the allocation map indicates are free. Reports any cylinder for which the CI does not also indicate that the cylinder is free.

**MI**

Scans the MI only.

**WCI**

Scans the WMI and WCIs.

**WDB**

Scans the WMI, WCIs, and WDBs. This is the default for the WAL log, which can be overridden by the WCI or WMI options.

**WMI**

Scans the WMI only.

**SKIPDECOMPRESS****SKIPDECOMP**

Prevents SCANDISK from uncompressing block-level compressed data blocks. This skips some validation checks, but can speed SCANDISK when there are many compressed data blocks in the system.

**WITHSPOOL**

Causes scandisk to check all spool data in addition to checking non-spool data.

If WITHSPOOL is not specified, only the first few spool cylinders are checked. If the scope of the command is limited to the MI, the entire MI is always scanned including non-spool and spool data.

***inquire\_opt***

Displays the lowest *tid* and *rowid* being scanned among the AMPS involved. This option also reports SCANDISK progress as a percentage of total time to completion and displays the errors encountered so far.

**NOCR**

Specifies to use regular data block preloads instead of cylinder reads. This is the default.

**CR**

Specifies to use cylinder reads instead of regular data block preloads.

**MAXERRCNT *n*****MAXERR *n***

Specifies the maximum number of errors SCANDISK should show per subtable per vproc. SCANDISK stops when it finds the specified number of errors. Use this setting to reduce the SCANDISK running time when there are an excessive number of errors reported.

**NONE**

Specifies that only one INQUIRE request is sent for the SCANDISK job.

***number***

An integer that defines the time interval to send to an INQUIRE request to display SCANDISK progress.

***timeopt***

One of the following:

- SECONDS, SECOND, SECON, SECO, SECS, SEC, S
- MINUTES, MINUTE, MINUT, MINU, MINS, MIN, M
- HOURS, HOUR, HOU, HO, HRS, HR, H
- DAYS, DAY, DA, D

For example, this command starts a SCANDISK job that reports SCANDISK progress every five minutes:

```
scandisk inquire 5 m
```

The default is SECONDS.

The maximum time interval allowed is seven days.

## Usage Notes

### General

You can run SCANDISK while the system is online and Vantage is available for normal operations.

Teradata recommends you run SCANDISK in the following situations:

- To validate data integrity before or after a system upgrade or expansion.
- If you suspect data corruption.
- As a routine data integrity check (perhaps weekly).

---

**Note:**

A slight performance impact might occur while SCANDISK is running.

---

You can rearrange the order of the syntax following the SCANDISK command. For example, the command SCANDISK NOCR MI is the same as the command SCANDISK MI NOCR.

If you do not type any options, SCANDISK defaults to DB and all subtables on the vproc. The default scope is to scan both the normal file system and the WAL log, each from the lowest (DB, WDB) level through the highest (MI, WMI). The free CIs are also scanned.

The SCANDISK command can be limited by the SCOPE command to scan, for example, just one table, just the WAL log, or just certain AMPs. For more information, see [SCOPE](#).

By default, SCANDISK uses regular data block preloads instead of cylinder reads. The CR option allows you to run SCANDISK using cylinder reads to preload data into cylinder slots which may improve SCANDISK performance. However, if other work also requires the use of cylinder slots, the competition for slots could slow down SCANDISK and the other work. In addition, the performance gain is dependent on the amount of data loaded, the distribution of the data, and the average block I/O size.

The NOCR option lets you turn off cylinder slot usage by SCANDISK, which could result in slower SCANDISK performance, but which will allow other work requiring cylinder slots to progress unimpeded.

SCANDISK reports only what it finds when scanning is completed.

The output from SCANDISK may include table row identifiers (RowIDs).

A RowID consists of a 16-byte value that includes the following information:

- The first 8-bytes is the internal partition number of the row.  
(For a nonpartitioned table, the internal partition number is zero, and the internal partition number is not actually stored in the row itself.)
- The next 8-bytes include a hash bucket value and uniqueness value for the row.

For more information on internal table row formats, see *Teradata Vantage™ - Database Design*, B035-1094.

SCANDISK start, finish, abort, and system reset times are logged to the Linux /var/log/messages file and also to the DBS.SW\_Event\_Log table.

## Aborting SCANDISK

Since SCANDISK DB verifies that every byte in the file system is accounted for, this process can be very time consuming. Therefore, you have the option of stopping the process by typing the following command:

```
[.] ABORT
```

ABORT can take up to 30 seconds to process.

After it is initiated, ABORT stops the SCANDISK process and reports the current status.

When the abort is successful, the following appears:

```
Are you sure you want to scandisk? (Y/N) y
Tue Feb 28, 1995 15:16:50 : Scandisk has been started on all AMP Vprocs in
the SCOPE.
```



```

Type 'ABORT' to stop them before completion
Type 'INQUIRE' to check on progress of command
ABORT
Abort request has been sent
Vproc 0 response
DB @ Cylinder 0 2 (0000 0002) Sector 16 (0010) length 1 (0001)
DB ref count doesn't match DBD row Count
Tue Feb 28, 1995 15:16:50 : The scandisk found problems
Vproc 1 response
Tue Feb 18, 1995 15:16:50 : The scandisk found nothing wrong
Ferret ==>

```

## Checking SCANDISK Status

Because SCANDISK can take a long time to run, you might want to do a status check after starting a scan. You can do this using the INQUIRE command:

```
[.] { INQUIRE | INQ }
```

INQUIRE displays the lowest *tid* and *rowid* being scanned among the AMPS involved, and reports SCANDISK progress as a percentage of total time to completion. It also displays a list of errors that occurred since the last INQUIRE command.

This .INQUIRE command is similar to the INQUIRE startup option for SCANDISK, but this command runs only once when you invoke it. Unlike the INQUIRE startup option, the INQUIRE command cannot be set to automatically repeat at specified intervals.

The SCANFREE section of INQUIRE output shows the status of scanning free CIs.

Sample INQUIRE output is shown below.

```

Ferret ==>
scandisk
Wed Apr 11, 2007 16:46:24 : Scandisk will be started
                        On All AMP vprocs
Do you wish to continue based upon this scope?? (Y/N)
y
Wed Apr 11, 2007 16:46:26 : Scandisk has been started
Type 'ABORT' to stop the command before completion
Type 'INQUIRE' to check on progress of command
inq

Inquire request has been sent

Wed Apr 11, 2007 16:46:30
SCANDISK STATUS :
Slowest vproc      0  is   2% done
Fastest vproc      0  is   2% done
The scandisk is about 2% done

Scanning Table: 0 1434 1024

Scanning Row: 0 1972 32488 0 1

```

Nobody has reached the SCANFREE stage

Wed Apr 11, 2007 16:46:30  
 2 of 2 vprocs responded with no messages or errors.  
 Type 'ABORT' to stop the command before completion  
 Type 'INQUIRE' to check on progress of command  
 inq

Inquire request has been sent

Wed Apr 11, 2007 16:46:36  
 SCANDISK STATUS :  
 Slowest vproc 1 is 4% done  
 Fastest vproc 0 is 5% done  
 The scandisk is about 4% done

Scanning Table: 0 1434 1024

Scanning Row: 0 28847 58771 0 95

Nobody has reached the SCANFREE stage

vproc 1 (0001) response

Wed Apr 11, 2007 16:46:33 : CI @ Cylinder 0 100 (0000 000064)  
 Wed Apr 11, 2007 16:46:33 : CID's First TID doesn't Match CI's First TID  
 0 1434 1024 (0000 059A 0400) ( test.trans )  
 0 1435 1024 (0000 059B 0400) ( TEST.data1 )  
 /nWed Apr 11, 2007 16:46:33 : CID's Last TID doesn't Match CI's Last TID  
 0 1434 1024 (0000 059A 0400) ( test.trans )  
 0 1435 1024 (0000 059B 0400) ( TEST.data1 )

Wed Apr 11, 2007 16:46:33 : Table Header is missing. May be in the process of being  
 dropped  
 0 1435 1024 (0000 059B 0400) ( TEST.data1 )

Wed Apr 11, 2007 16:46:34 : DB @ Cylinder 0 100 (0000 000064) Sector 3418 (0D5A) le  
 ngth 126 (007E)  
 : TID : 0 1435 1024 (0000 059B 0400) ( TEST.data1 )

Wed Apr 11, 2007 16:46:34 : SRD TID.uniq doesn't match DB tid.uniq  
 0 1435 (0000 059B)  
 0 1434 (0000 059A)

Wed Apr 11, 2007 16:46:34 : DB @ Cylinder 0 100 (0000 000064) Sector 3629 (0E2D) le  
 ngth 70 (0046)  
 : TID : 0 1435 1024 (0000 059B 0400) ( TEST.data1 )

Wed Apr 11, 2007 16:46:34 : SRD TID.uniq doesn't match DB tid.uniq  
 0 1435 (0000 059B)  
 0 1434 (0000 059A)

Wed Apr 11, 2007 16:46:34 : DB @ Cylinder 0 100 (0000 000064) Sector 3699 (0E73) le  
 ngth 70 (0046)  
 : TID : 0 1435 1024 (0000 059B 0400) ( TEST.data1 )

Wed Apr 11, 2007 16:46:34 : SRD TID.uniq doesn't match DB tid.uniq  
 0 1435 (0000 059B)  
 0 1434 (0000 059A)

Wed Apr 11, 2007 16:46:34 : DB @ Cylinder 0 100 (0000 000064) Sector 48 (0030) leng  
 th 65 (0041)  
 : TID : 0 1435 1024 (0000 059B 0400) ( TEST.data1 )

Wed Apr 11, 2007 16:46:34 : SRD TID.uniq doesn't match DB tid.uniq  
 0 1435 (0000 059B)  
 0 1434 (0000 059A)

```

Wed Apr 11, 2007 16:46:34 : DB @ Cylinder 0 100 (0000 000064) Sector 432 (01B0) len
gth 65 (0041)
                        : TID :      0 1435 1024 (0000 059B 0400) ( TEST.data1 )

Wed Apr 11, 2007 16:46:34 : SRD TID.uniq doesn't match DB tid.uniq
0 1435 (0000 059B)
0 1434 (0000 059A)

Wed Apr 11, 2007 16:46:34 : DB @ Cylinder 0 100 (0000 000064) Sector 113 (0071) len
gth 66 (0042)
                        : TID :      0 1435 1024 (0000 059B 0400) ( TEST.data1 )

Wed Apr 11, 2007 16:46:34 : SRD TID.uniq doesn't match DB tid.uniq
0 1435 (0000 059B)
0 1434 (0000 059A)

Wed Apr 11, 2007 16:46:34 : DB @ Cylinder 0 100 (0000 000064) Sector 179 (00B3) len
gth 66 (0042)
                        : TID :      0 1435 1024 (0000 059B 0400) ( TEST.data1 )

Wed Apr 11, 2007 16:46:34 : SRD TID.uniq doesn't match DB tid.uniq
0 1435 (0000 059B)
0 1434 (0000 059A)

Wed Apr 11, 2007 16:46:34 : DB @ Cylinder 0 100 (0000 000064) Sector 2646 (0A56) le
ngth 122 (007A)
                        : TID :      0 1435 1024 (0000 059B 0400) ( TEST.data1 )

Wed Apr 11, 2007 16:46:34 : SRD TID.uniq doesn't match DB tid.uniq
0 1435 (0000 059B)
0 1434 (0000 059A)

Wed Apr 11, 2007 16:46:34 : DB @ Cylinder 0 100 (0000 000064) Sector 1953 (07A1) le
ngth 66 (0042)
                        : TID :      0 1435 1024 (0000 059B 0400) ( TEST.data1 )

Wed Apr 11, 2007 16:46:34 : SRD TID.uniq doesn't match DB tid.uniq
0 1435 (0000 059B)
0 1434 (0000 059A)

Wed Apr 11, 2007 16:46:34 : DB @ Cylinder 0 100 (0000 000064) Sector 2346 (092A) le
ngth 66 (0042)
                        : TID :      0 1435 1024 (0000 059B 0400) ( TEST.data1 )

Wed Apr 11, 2007 16:46:34 : SRD TID.uniq doesn't match DB tid.uniq
0 1435 (0000 059B)
0 1434 (0000 059A)

Wed Apr 11, 2007 16:46:34 : DB @ Cylinder 0 100 (0000 000064) Sector 2932 (0B74) le
ngth 72 (0048)
                        : TID :      0 1435 1024 (0000 059B 0400) ( TEST.data1 )

Wed Apr 11, 2007 16:46:34 : SRD TID.uniq doesn't match DB tid.uniq
0 1435 (0000 059B)
0 1434 (0000 059A)

Wed Apr 11, 2007 16:46:34 : DB @ Cylinder 0 100 (0000 000064) Sector 3004 (0BBC) le
ngth 72 (0048)
                        : TID :      0 1435 1024 (0000 059B 0400) ( TEST.data1 )

Wed Apr 11, 2007 16:46:34 : SRD TID.uniq doesn't match DB tid.uniq
0 1435 (0000 059B)
0 1434 (0000 059A)

Wed Apr 11, 2007 16:46:34 : DB @ Cylinder 0 100 (0000 000064) Sector 684 (02AC) len
gth 72 (0048)
                        : TID :      0 1435 1024 (0000 059B 0400) ( TEST.data1 )

Wed Apr 11, 2007 16:46:34 : SRD TID.uniq doesn't match DB tid.uniq

```

```

0 1435 (0000 059B)
0 1434 (0000 059A)

Wed Apr 11, 2007 16:46:34 : DB @ Cylinder 0 100 (0000 000064) Sector 756 (02F4) length 72 (0048)
                        : TID :      0 1435 1024 (0000 059B 0400) ( TEST.data1 )

Wed Apr 11, 2007 16:46:34 : SRD TID.uniq doesn't match DB tid.uniq
0 1435 (0000 059B)
0 1434 (0000 059A)

Wed Apr 11, 2007 16:46:34 : DB @ Cylinder 0 100 (0000 000064) Sector 1191 (04A7) length 109 (006D)
                        : TID :      0 1435 1024 (0000 059B 0400) ( TEST.data1 )

Wed Apr 11, 2007 16:46:34 : SRD TID.uniq doesn't match DB tid.uniq
0 1435 (0000 059B)
0 1434 (0000 059A)

Wed Apr 11, 2007 16:46:34 : DB @ Cylinder 0 100 (0000 000064) Sector 1606 (0646) length 74 (004A)
                        : TID :      0 1435 1024 (0000 059B 0400) ( TEST.data1 )

Wed Apr 11, 2007 16:46:34 : SRD TID.uniq doesn't match DB tid.uniq
0 1435 (0000 059B)
0 1434 (0000 059A)

Wed Apr 11, 2007 16:46:34 : DB @ Cylinder 0 100 (0000 000064) Sector 2768 (0AD0) length 74 (004A)
                        : TID :      0 1435 1024 (0000 059B 0400) ( TEST.data1 )

Wed Apr 11, 2007 16:46:34 : SRD TID.uniq doesn't match DB tid.uniq
0 1435 (0000 059B)
0 1434 (0000 059A)

Wed Apr 11, 2007 16:46:34 : DB @ Cylinder 0 100 (0000 000064) Sector 3150 (0C4E) length 119 (0077)
                        : TID :      0 1435 1024 (0000 059B 0400) ( TEST.data1 )

Wed Apr 11, 2007 16:46:34 : SRD TID.uniq doesn't match DB tid.uniq
0 1435 (0000 059B)
0 1434 (0000 059A)

Wed Apr 11, 2007 16:46:34 : DB @ Cylinder 0 100 (0000 000064) Sector 1056 (0420) length 69 (0045)
                        : TID :      0 1435 1024 (0000 059B 0400) ( TEST.data1 )

Wed Apr 11, 2007 16:46:34 : SRD TID.uniq doesn't match DB tid.uniq
0 1435 (0000 059B)
0 1434 (0000 059A)

Wed Apr 11, 2007 16:46:34 : DB @ Cylinder 0 100 (0000 000064) Sector 3544 (0DD8) length 69 (0045)
                        : TID :      0 1435 1024 (0000 059B 0400) ( TEST.data1 )

Wed Apr 11, 2007 16:46:34 : SRD TID.uniq doesn't match DB tid.uniq
0 1435 (0000 059B)
0 1434 (0000 059A)

Wed Apr 11, 2007 16:46:34 : CI @ Cylinder 0 266 (0000 00010A)

Wed Apr 11, 2007 16:46:34 : Table Ids are out of order
0 1435 1024 (0000 059B 0400) ( TEST.data1 )
0 1434 1024 (0000 059A 0400) ( test.trans )

```

```

Wed Apr 11, 2007 16:46:36
1 of 2 vprocs responded with no messages or errors.
Wed Apr 11, 2007 16:46:36
1 of 2 vprocs responded with messages or errors as specified above
Type 'ABORT' to stop the command before completion
Type 'INQUIRE' to check on progress of command

```

## Examples

### Example: SCANDISK Output

```

Ferret ==>
scandisk
Tue Feb 28, 1995 15:16:50 :Scandisk has been started on all AMP Vprocs in the S
COPE.
Vproc 0 response
DB @ Cylinder 0 2 (0000 0002) Sector 16 (0010) length 1 (0001)
DB ref count doesn't match DBD row Count

Tue Feb 28, 1995 15:16:50 : The scandisk found problems
Vproc 1 response
Tue Feb 28, 1995 15:16:50 : The scandisk found nothing wrong

```

### Example: SCANDISK Output for LSI Interrupted Write in a CI

The following is an example of output that SCANDISK generates when it finds an LSI interrupted write pattern in a CI. This error is specific to LSI brand storage.

```

Ferret ==>
scandisk ci
Mon May 06, 2002 15:12:20 :Scandisk has been started on all AMP Vprocs in the
SCOPE.
vproc 0 (0000) response
Mon May 06, 2002 15:55:21 : CI @ Cylinder 0 7 (0000 000007)
Mon May 06, 2002 15:55:21 : LSI interrupted write pattern found in CI.
0120 MAY 05 05:02:35 LUN 1111, Start Block 00004545, Blocks 0400

SRD
num      table id      firstdbd  dbdcount offset
      u0      u1      tai
-----
0001 0000 0494 0800      FFFF      0014 001E
Mon May 06, 2002 15:55:21 : Invalid DBD sector length of 14901 (3A35) found

```

```

Mon May 06, 2002 15:55:21 : Invalid DBD sector length of 26912 (6920) found
Mon May 06, 2002 15:55:21 : Invalid DBD sector length of 25972 (6574) found
Mon May 06, 2002 15:55:21 : Invalid DBD sector length of 12336 (3030) found
Mon May 06, 2002 15:55:21 : First rowid out of order dbds 18 (0012) and 19 (0013)

```

### Example: SCANDISK Output for LSI Interrupted Write in a DB

The following is an example of output that SCANDISK generates when it finds an LSI interrupted write pattern in a DB. This error is specific to LSI brand storage.

```

Ferret ==>
scandisk db
Tue Feb 28, 1995 15:16:50 :Scandisk has been started on all AMP Vprocs in
the SCOPE.
vproc 0 (0000) response
Mon May 06, 2002 15:12:20
1 of 1 vprocs responded with no messages or errors.
Type 'ABORT' to stop the command before completion
Type 'INQUIRE' to check on progress of command
Reading
vproc 0 (0000) response
Mon May 06, 2002 15:11:11 : CI @ Cylinder 0 7 (0000 000007)
Mon May 06, 2002 15:11:11 : LSI interrupted write pattern found in DB.
0120 MAY 05 05:02:35 LUN 1111, Start Block 00004545, Blocks 0400
Mon May 06, 2002 15:11:11 : rows -1 (FFFFFFFF) and 0 (0000) are out of order

```

### Example: Using SCANDISK to Scan the WAL Log

```

scope wal
scandisk

```

### Example: Using SCANDISK to Scan One Data Subtable

```

scope table 'employee.emp' p
scandisk

```

### Example: Using the INQUIRE Option to Display SCANDISK Progress

In the following example, an INQUIRE command is sent per minute. Therefore, you get a display of SCANDISK progress every minute.

```
Ferret ==>  
> scandisk inq 1 m  
  
> Mon Jul 17, 2012 05:02:45  
SCANDISK STATUS :  
Slowest vproc 2 is 44% done  
Fastest vproc 3 is 64% done  
The scandisk is about 55% done
```

## SCOPE

The SCOPE command defines the scope for subsequent COMPRESS, DEFRAGMENT, PACKDISK, SCANDISK, SHOWAMPRECOVERY, SHOWBLOCKS, SHOWCOMPRESS, SHOWCYLALLOC, SHOWFSP, SHOWSPACE, SHOWWHERE, and UNCOMPRESS commands. It defines the class of tables, range of tables, vprocs/maps (AMPs), and cylinders, or the WAL log to be used as parameters with these Ferret commands.

Each SCOPE command defines a new scope and is not a continuation of the last one.

## Syntax Rules

The Ferret SCOPE syntax is unusual. The following rules apply.

| IF you specify a ...           | THEN ...                                                                                    |
|--------------------------------|---------------------------------------------------------------------------------------------|
| <i>single_syntax_element</i>   | commas and spaces are not required. Parentheses are optional.                               |
| <i>list_of_syntax_elements</i> | each syntax element must be separated by either a comma or space. Parentheses are required. |

The following are valid examples of the SCOPE command:

```
scope vproc 1
scope vproc (1)
scope vproc (1,3)
scope vproc (1, 3)
```

The following are invalid examples of the SCOPE command:

```
scope vproc 1,3
scope vproc 1 3
```

## Syntax

### Note:

Not all combinations of acceptable syntax are shown.

```
SCOPE {
  { class | cylinder | table | vproc | map } [,...] |
  ALL
}
```



**class**

```

CLASS {
  ( { JRNL |
    { PERMANENT | P } |
    { TEMPORARY | TEMP } |
    PSPOOL |
    SPOOL
    } [[,]...]
  ) |
  ALL
}

```

**cylinder**

```
{ CYLINDER | CYL } { ( cylid [[,]...] ) | cylid | ALL }
```

**table**

```
TABLE { ( tid [[,]...] ) | tid | ALL }
```

**vproc**

```

VPROC {
  vproc_number |
  ( { vproc_number | vproc_number TO vproc_number } [[,]...] ) |
  ALL
}

```

**map**

```
MAP { "mapname" | ( "mapname" [[,]...] ) | ALL }
```

**Syntax Elements****CLASS**

The class of tables for the scope of a subsequent command.

**Note:**

Classes are subtable ranges, so if a command requires tables in the recorded scope, you may specify a CLASS on the SCOPE command.

| CLASS     | Description                                                                                          |
|-----------|------------------------------------------------------------------------------------------------------|
| JRNL      | Specifies the permanent journal tables containing non-visible user data.                             |
| PERMANENT | Specifies the permanent tables containing visible user data.                                         |
| TEMPORARY | Specifies the temporary worktables (global temporary table instances) containing non-permanent data. |
| PSPOOL    | Specifies the work tables containing non-permanent data that persist across restarts.                |
| SPOOL     | Specifies the intermediate worktables containing non-permanent data.                                 |
| ALL       | Specifies all the table classes in the configuration.                                                |

**CYLINDER**

The cylinders that are to be acted upon only by a subsequent DEFRAGMENT command. No other command uses the CYLINDER SCOPE.

**Note:**

A SCOPE command with CYLINDER arguments must also include VPROC arguments to be valid.

| CYLINDER     | Description                                                         |
|--------------|---------------------------------------------------------------------|
| <i>cylid</i> | Specifies the cylinder ID number, a 16-character hexadecimal value. |
| ALL          | Specifies all the cylinders in the vproc.                           |

**VPROC**

The range of AMP vproc ID numbers, or all AMPs in the current configuration that are to be acted upon by a subsequent command.

| VPROC               | Description                                                                                                                                                                                                                   |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>vproc_number</i> | specifies a single AMP vprocID number or a range of numbers. Valid AMP vproc ID numbers are from 0 to 16199. There is no default. If you type a vproc identifier that is assigned to a non-AMP vproc, Ferret issues an error. |

| VPROC | Description                                                                                    |
|-------|------------------------------------------------------------------------------------------------|
|       | <b>Note:</b><br>Ferret also informs you when you select a vproc number assigned to a down AMP. |
| ALL   | Specifies all the AMP vprocs in the configuration. VPROC ALL is synonymous with MAP ALL        |

## MAP

Similar to VPROC, however it scopes to the AMPs included in the specified contiguous maps. Map names must be enclosed in double-quotation marks. MAP ALL is synonymous with VPROC ALL.

---

### Note:

If you try to set SCOPE to a sparse map, the effective Ferret scope is set to the range of AMPs in the parent contiguous map of the sparse map.

---

## TABLE

The subtables that are to be acted upon by a subsequent command.

| TABLE      | Description                                                                                                                                                                                                              |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>tid</i> | Specifies the subtable to process.<br><br><b>Note:</b><br>The <i>typeandindex</i> component of <i>tid</i> is required. For more information on <i>tid</i> formatting, refer to <a href="#">Using Ferret Parameters</a> . |
| ALL        | Specifies all the subtables in the configuration.                                                                                                                                                                        |

## WAL

The WAL log.

## ALL

to reset SCOPE to the default startup settings of Ferret. The initial scope will consist, as appropriate to each command, of all tables, all cylinders, all vprocs, and the WAL log.

## Usage Notes

The following table shows how scopes are interpreted.

| Scope Type | Interpretation                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Table      | <p>Specified subtables, which can be selected subtables, all subtables, or classes.</p> <p><b>Note:</b></p> <p>Although disk space allocated for TJ and WAL records is charged against the table 0 26, no actual TJ or WAL records are found in the subtables of this table. Instead, these records are in the WAL log. The only row that exists in any subtable of the table 0, 26 is the table header in subtable 0.</p> <p>Table scopes imply cylinder scopes, and all-table scopes imply free CIs.</p> |
| Cylinder   | <p>All the specified cylinders.</p> <p>Cylinder scopes can be specified by implication as subtable scopes. For the DEFRAGMENT command, this means all cylinders containing the specified subtables. An explicit cylinder specification is only meaningful to the DEFRAGMENT command. If you specify a free cylinder in <code>CYLINDER <i>cylid</i></code>, Ferret SCANDISK will respond with an indication that the cylinder did not need to be defragmented.</p>                                          |
| Vproc      | All subtables, all free CIs, and the WAL log.                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Map        | The AMPs associated with the specified maps.                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| WAL        | <p>The entire WAL log.</p> <p><b>Note:</b></p> <p>Although disk space allocated for TJ and WAL records is charged against the table 0 26, no actual TJ or WAL records are found in the subtables of this table. Instead, these records are in the WAL log. The only row that exists in any subtable of the table 0, 26 is the table header in subtable 0.</p>                                                                                                                                              |

The Ferret commands utilize SCOPE as follows, where classes are collections of tables and result in Table on AMP scopes.

| Command    | Description                                                                                                                                                                                                                           |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DEFRAGMENT | The command uses Table on AMP scopes, or Cylinder on AMP or MAP scopes. The scope can contain tables only, both tables and vprocs or maps, or both vprocs /maps and cylinders. For more information, see <a href="#">DEFRAGMENT</a> . |
| PACKDISK   | The command uses AMP scopes, or Table on AMP scopes. The scope selected must include either vprocs/maps or tables, but not both. For more information, see <a href="#">PACKDISK</a> .                                                 |
| SCANDISK   | The command uses AMP scopes, or Table on AMP scopes, and WAL scope. The scope can contain tables only, vprocs/maps only, both tables and vprocs/maps, or the WAL log. For more information, see <a href="#">SCANDISK</a> .            |
| SHOWBLOCKS | The command uses Table on AMP scopes and WAL scope. The scope can contain tables only or the WAL log. For more information, see <a href="#">SHOWBLOCKS</a> .                                                                          |
| SHOWFSP    | The command uses AMP scopes, or Table on AMP scopes.                                                                                                                                                                                  |

| Command   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|           | The scope can include one or more tables, one or more vprocs/maps, or the entire system. For more information, see <a href="#">SHOWFSP</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                |
| SHOWSPACE | <p>The command uses AMP scopes, or Table on AMP scopes, and WAL scope.</p> <p><b>Note:</b></p> <p>Although disk space allocated for TJ and WAL records is charged against the table 0 26, no actual TJ or WAL records are found in the subtables of this table. Instead, these records are in the WAL log. The only row that exists in any subtable of the table 0, 26 is the table header in subtable 0.</p> <p>The scope can contain tables only, vprocs/maps only, both tables and vprocs/maps, or the WAL log. For more information, see <a href="#">SHOWSPACE</a>.</p> |

All scopes are not applicable to all commands. When a command is executed and the scope is not applicable to the command, either the command is rejected or the inapplicable portions of the scope are ignored.

The SHOWDEFAULTS command displays the various components of a recorded scope, which are interpreted individually by each of the commands: COMPRESS, DEFRAGMENT, PACKDISK, SCANDISK, SHOWBLOCKS, SHOWCOMPRESS, SHOWCYLALLOC, SHOWFSP, SHOWSPACE, SHOWWHERE, and UNCOMPRESS.

The scopes appear in the SHOWDEFAULTS output as tables on vprocs, cylinders on vprocs, vprocs, maps, or the WAL log. For more information, see [SHOWDEFAULTS](#).

## Examples

### Example: SCOPE Command Examples

The following command examples are representative of how the SCOPE command is normally used:

| Command                                           | Action                                  |
|---------------------------------------------------|-----------------------------------------|
| Scope vproc ALL                                   | Select all the AMPs on the system.      |
| Scope vproc 4, vproc 6<br>or<br>Scope vproc (4,6) | Select vprocs 4 and 6.                  |
| Scope Table 400H 0 400H                           | Select table 400H 0 400H on all vprocs. |

| Command                                                  | Action                                                                                                                                                        |
|----------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Scope Class (P, JRNL)<br>or<br>Scope Class P, Class JRNL | Select all the Permanent and Journal tables on all vprocs.                                                                                                    |
| Scope Class P, Class JRNL, vproc 4                       | Select all the Permanent and Journal tables on vproc 4.                                                                                                       |
| Scope Cyl<br>(000100000000003C 0001000000000043)         | Select cylinders 000100000000003C through 0001000000000043.<br><b>Note:</b><br>All cylinders specified in a single SCOPE command must belong to the same AMP. |

### Example: Scoping Ferret to Fallback Tables

Assume that table T4 is a table in database XYZ and has a table number of 0 1198.

- One of the following commands would place all T4 fallback subtables under scope.
  - SCOPE "TABLE XYZ.T4 F"\*
  - SCOPE TABLE 0 1198 F\*
- One of the following commands would place all T4 subtables under scope.
  - SCOPE TABLE "XYZ.T4" \*
  - SCOPE TABLE 0 1198 \*

### Example: Resetting Ferret Scope to the Default

The following example resets the scope to the default Ferret startup settings. Some of the output is omitted to condense the example.

```
scope all

showd

Scope for the Defrag command is :
  All Cylinders
  On all AMP vprocs

Scope for the Packdisk command is :
  All of the AMP vprocs
```

Scope for the Scandisk command is :  
All of the AMP vprocs

Scope for the Showblocks command is :  
All tables  
WAL Log  
On all AMP vprocs

Scope for the Showspace command is :  
All of the AMP vprocs

Scope for the Showwhere command is :  
All tables  
On all AMP vprocs

### Example: Scoping Ferret to One AMP

The following example shows a whole AMP scope, which includes the WAL log.

```
SCOPE VPROC 1
```

The SCOPE has been set

```
Ferret => SHOWD
```

The current setting of the Input Radix is Decimal  
The current setting of the Output Radix is Decimal

Scope for the Defrag command is :  
All Cylinders  
On AMP vproc(s) 1

Scope for the Packdisk command is :  
AMP vproc(s) 1

Scope for the Scandisk command is :  
AMP vproc(s) 1

Scope for the Showblocks command is :  
All tables  
WAL Log  
On AMP vproc(s) 1

Scope for the Showspace command is :  
AMP vproc(s) 1

Scope for the ShowFSP command is :  
AMP vproc(s) 1

Scope for the Showwhere command is :  
AMP vproc(s) 1

### Example: Scoping Ferret to a Map

```
scope map "td_map1"
```

The SCOPE has been set

Ferret ==>

```
> showd
```

```
showd
```

The current setting of the Input Radix is Decimal  
The current setting of the Output Radix is Decimal  
Ferret is running with Script Mode -- Disabled

Scope for the Defrag command is :  
Invalid, because no tables or cylinders are specified

Scope for the Packdisk command is :  
MAP TD\_MAP1 ( vproc(s) 0-3 )

Scope for the Scandisk command is :  
MAP TD\_MAP1 ( vproc(s) 0-3 )

Scope for the Showblocks command is :  
Invalid because there are no tables or log records selected

Scope for the Showspace command is :  
MAP TD\_MAP1 ( vproc(s) 0-3 )

Scope for the ShowFSP command is :  
MAP TD\_MAP1 ( vproc(s) 0-3 )



```

Scope for the Showwhere command is :
  MAP TD_MAP1 ( vproc(s) 0-3 )

Scope for the Showcylalloc command is :
  MAP TD_MAP1 ( vproc(s) 0-3 )

Scope for the Force command is :
  Ignores the map scope set by scope command

Scope for the Showcompress command is :
  MAP TD_MAP1 ( vproc(s) 0-3 )

Scope for the [un]Compress command is :
  Ignores the map scope set by scope command

```

### Example: Scoping Ferret to the WAL Log on All AMPs

The following example shows the scope set to the WAL log on all AMPs.

---

#### Note:

Only SCANDISK, SHOWBLOCKS, SHOWSPACE, and SHOWWHERE work with this scope.

---

```

SCOPE WAL

The SCOPE has been set

Ferret => SHOWD

The current setting of the Input Radix is Decimal
The current setting of the Output Radix is Decimal

Scope for the Defrag command is :
  On all AMP vprocs

Scope for the Packdisk command is :
  On all AMP vprocs

Scope for the Scandisk command is :
  WAL Log
  On all AMP vprocs

Scope for the Showblocks command is :

```

```
WAL Log
On all AMP vprocs
```

Scope for the Showspace command is :

```
WAL Log
On all AMP vprocs
```

Scope for the ShowFSP command is :

```
Invalid, because it contains WAL
```

Scope for the Showwhere command is :

```
WAL Log
On all AMP vprocs
```

### Example: Scoping Ferret to Exclude the WAL Log

The following example shows the scope set to all the tables on all amps, but not the WAL log.

```
scope class all
```

```
showd
```

Scope for the Defrag command is :

```
Table(s) 0 0 0 TO 65535 65535 65535
On All AMP vprocs
```

Scope for the Packdisk command is :

```
Table(s) 0 0 0 TO 65535 65535 65535
On All AMP vprocs
```

Scope for the Scandisk command is :

```
Table(s) 0 0 0 TO 65535 65535 65535
On All AMP vprocs
```

Scope for the Showblocks command is :

```
Table(s) 0 0 0 TO 65535 65535 65535
On All AMP vprocs
```

Scope for the Showspace command is :

```
Table(s) 0 0 0 TO 65535 65535 65535
On All AMP vprocs
```

Scope for the ShowFSP command is :

```
Table(s) 0 0 0 TO 65535 65535 65535  
On All AMP vprocs
```

Scope for the Showwhere command is :

```
Table(s) 0 0 0 TO 65535 65535 65535  
On All AMP vprocs
```

## SHOWAMPRECOVERYBLC

The SHOWAMPRECOVERYBLC command shows which compress and uncompress operations are in progress during a system recovery after a restart.

Space allocations are automatically updated following compress and uncompress operations.

### Syntax

```
{ SHOWAMPRECOVERYBLC | SHOWAMPR }
```

Example: Displaying in-progress compression during system recovery

The following is a portion of the output of the SHOWAMPRECOVERYBLC command when compress is in progress during recovery.

```
Ferret ==>
showampr

The following BLC operation is in progress by AMP recovery
```

| Database Name       | Table Name               | Map Name | Table ID |
|---------------------|--------------------------|----------|----------|
| 1. ShowCmdsshowampr | T8_table_WithoutFallback | TD_MAP2  | 0 3274   |
| 0 TABCOMPRESS       |                          |          |          |
| *. ALL of the above |                          |          |          |

```

Do you want to cancel the above BLC operation....? (Y/N)
> Y
Y

Enter the number of the operation to cancel, or type * to cancel all
> *
*

Cancelling the currently in progress BLC operation on all above tables
Note: Additional Compress / Uncompress operations may still be pending as part of recovery
Ferret ==>
```



## SHOWBLOCKS

The SHOWBLOCKS command displays statistics about data block size, number of rows per data block, and information about the compression status of data blocks and tables for all the tables defined by the SCOPE command. SHOWBLOCKS can also display WAL log statistics.

---

### Note:

The CreateFsysInfoTable and PopulateFsysInfoTable macros provide a way to generate SHOWBLOCKS-like file system information to an SQL table. See also the information about file system information macros and functions in *Teradata Vantage™ - SQL Operators and User-Defined Functions*, B035-1210.

---

### Syntax

```
{ SHOWBLOCKS | SHOWB }
  [ /S | /M | /L ]
  [ COMPRESSTLA = { AUTOTEMP | DEFAULT | MANUAL | ALWAYS | NEVER } ]
  [ PERVPROC ]
```

---

### Note:

The online help lists the display options (/S, /M, /L) as /dispt.

---

### Syntax Elements

#### /S

For each primary data subtable, display the following:

- A histogram of block sizes
- The minimum, average, and maximum block size per subtable
- Block compression information (status, estimated compression ratio, estimated uncompressed)

This is the default display.

#### /M

Display what the /S option shows, and additionally display the table name, indicate if it is a foreign table, show the name of the map the table uses for data distribution, and display statistics for all subtables. For each subtable, display the BLOCKCOMPRESSION value set when the table was created or altered.

#### /L

For each subtable, for each block size, display the following:

- Table name

- Indicator if table is a foreign table
- Map that the table uses for data distribution
- Number of blocks
- The minimum, average, and maximum number of rows per data block size
- Block compression information (status, estimated compression ratio, estimated percent uncompressed)

Displays the statistics for all subtables.

## COMPRESSTLA

SHOWBLOCKS output is filtered to show only information for tables with BLOCKCOMPRESSION set to the specified value, one of the following: AUTOTEMP, DEFAULT, MANUAL, ALWAYS, or NEVER.

BLOCKCOMPRESSION can be set for a table in a CREATE TABLE or ALTER TABLE statement.

## PERVPROC

Displays data block statistics per AMP vproc within the current scope.

---

### Note:

The CreateFsysInfoTable and PopulateFsysInfoTable macros do not support the PERVPROC option.

---

The Ferret ABORT command can be used to halt this operation during command execution.

## Usage Notes

If the scope is set to both vproc and map, SHOWBLOCKS displays information for those specified vprocs that are in the specified map.

The output of the long display is one line for every size data block from every subtable of every table in the scope.

The output can be lengthy; therefore, consider using the OUTPUT command to redirect the output to a file.

This command can be used with the DATABLOCKSIZE option of the CREATE TABLE and ALTER TABLE SQL statements to determine the best data block size for tables based on performance requirements.

Vantage can isolate some file system errors to a specific data or index subtable, or to a contiguous range of rows ("region") in a data or index subtable. In these cases, Vantage marks only the affected subtable or region down. This improves system performance and availability by allowing transactions that do not require access to the down subtable or rows to proceed, without causing a database crash that would require a system restart. If SHOWBLOCKS encounters down regions, it skips these regions, and displays the percentage of total space that was skipped.



---

**Note:**

Although disk space allocated for TJ and WAL records is charged against table 0 26, no actual TJ or WAL records are found in the subtables of this table. Instead, these records are in the WAL log. The only row that exists in any subtable of table 0, 26 is the table header in subtable 0.

---

## Example: SHOWBLOCKS short display

The following is a portion of the output of the showblocks command using the /s short display option:

| Table ID | Compression Status | Estimated Compression Ratio | Estimated % of Blocks Un-compressed | Distribution of data block sizes (by range of number of sectors) |      |     |      |       |       |       |       |       |        |         |         |         |         | Data block size statistics (sectors) |     |     | Total Number of Data Blocks | Total Number of Cylinders |     |   |    |
|----------|--------------------|-----------------------------|-------------------------------------|------------------------------------------------------------------|------|-----|------|-------|-------|-------|-------|-------|--------|---------|---------|---------|---------|--------------------------------------|-----|-----|-----------------------------|---------------------------|-----|---|----|
|          |                    |                             |                                     | 1-1                                                              | 2-3  | 4-7 | 8-15 | 16-31 | 32-47 | 48-63 | 64-79 | 80-95 | 96-111 | 112-127 | 128-159 | 160-191 | 192-223 | 224-255                              | Min | Avg |                             |                           | Max |   |    |
| 0 256    | N                  | 50%<br>88.14%               | 20%<br>81.69%                       | 100%                                                             | 100% | 2%  | 88%  | 12%   |       |       |       |       |        |         |         |         |         | 9                                    | 13  | 19  | 8                           | 8*                        |     |   |    |
| 0 265    | N                  |                             |                                     |                                                                  |      |     |      |       |       |       |       |       |        |         |         |         |         |                                      |     |     | 1                           | 1                         | 1   | 3 | 3* |
| 0 266    | N                  |                             |                                     |                                                                  |      |     |      |       |       |       |       |       |        |         |         |         |         |                                      |     |     |                             | 2                         | 2   | 3 | 5  |
| 0 1804   | C                  |                             |                                     |                                                                  |      |     | 34%  |       |       |       |       |       |        |         |         |         |         |                                      | 2   | 6   | 15                          | 4974                      | 12* |   |    |
| 0 1805   | PC                 |                             |                                     |                                                                  |      |     | 18%  |       |       |       |       |       |        |         |         |         |         |                                      | 4   | 94  | 121                         | 4008                      | 20* |   |    |
| 0 2087   | U                  |                             |                                     |                                                                  | 99%  | 1%  |      |       |       |       |       |       | 26%    | 56%     |         |         |         | 1                                    | 13  | 15  | 689                         | 8*                        |     |   |    |
| WAL LOG  | N                  |                             |                                     | 7%                                                               | 83%  | 3%  | 1%   | 5%    | 1%    |       |       |       |        |         |         |         |         | 1                                    | 4   | 255 | 2475                        | 25                        |     |   |    |

\* Reported in units of Large Cylinders  
(A Large Cylinder is 6 times the size of a small cylinder)  
"Compression Status" : C = Fully Compressed  
PC = Partially Compressed  
U = Fully Uncompressed  
N = Not Compressible

## Example: SHOWBLOCKS medium display

The following is a portion of the output of the showblocks command using the /m medium display option:

Showblocks has been started on all AMP vprocs in the SCOPE.  
Type 'ABORT' to stop the command before completion

| Table ID | Compression Status | Estimated Compression Ratio                              | Estimated % of Blocks Un-compressed | Distribution of data block sizes (by range of number of sectors) |      |       |        |         |         |         |         |         |         |         |          |           |           | Data block size statistics (sectors) |     |     | Total Number of Data Blocks | Total Number of Cylinders |
|----------|--------------------|----------------------------------------------------------|-------------------------------------|------------------------------------------------------------------|------|-------|--------|---------|---------|---------|---------|---------|---------|---------|----------|-----------|-----------|--------------------------------------|-----|-----|-----------------------------|---------------------------|
|          |                    |                                                          |                                     | 1-8                                                              | 9-24 | 25-64 | 65-120 | 121-168 | 169-216 | 217-256 | 257-360 | 361-456 | 457-512 | 513-760 | 761-1024 | 1025-1304 | 1305-1632 | 1633-2048                            | Min | Avg |                             |                           |
| 0        | 0                  | DBC.FIRSTPERMTABLE (DEFAULT) (Map: TD_DATADITIONARYMAP)  |                                     |                                                                  |      |       |        |         |         |         |         |         |         |         |          |           |           |                                      |     |     |                             |                           |
|          | 0                  | N                                                        |                                     | 100%                                                             |      |       |        |         |         |         |         |         |         |         |          |           | 1         | 1                                    | 1   | 4   | 4*                          |                           |
| 0        | 7                  | DBC.RCConfiguration (DEFAULT) (Map: TD_DATADITIONARYMAP) |                                     |                                                                  |      |       |        |         |         |         |         |         |         |         |          |           |           |                                      |     |     |                             |                           |
|          | 0                  | N                                                        |                                     | 100%                                                             |      |       |        |         |         |         |         |         |         |         |          |           | 2         | 2                                    | 2   | 4   | 4*                          |                           |
| 0        | 25                 | DBC.Global (DEFAULT) (Map: TD_DATADITIONARYMAP)          |                                     |                                                                  |      |       |        |         |         |         |         |         |         |         |          |           |           |                                      |     |     |                             |                           |
|          | 0                  | N                                                        |                                     | 100%                                                             |      |       |        |         |         |         |         |         |         |         |          |           | 2         | 2                                    | 2   | 4   | 4*                          |                           |
| 1024     |                    | N                                                        |                                     | 100%                                                             |      |       |        |         |         |         |         |         |         |         |          |           | 1         | 1                                    | 1   | 1   | 1*                          |                           |
| 2048     |                    | N                                                        |                                     | 100%                                                             |      |       |        |         |         |         |         |         |         |         |          |           | 1         | 1                                    | 1   | 1   | 1*                          |                           |
| 0        | 26                 | DBC.TransientJournal (DEFAULT) (Map: TD_GLOBALMAP)       |                                     |                                                                  |      |       |        |         |         |         |         |         |         |         |          |           |           |                                      |     |     |                             |                           |
|          | 0                  | N                                                        |                                     | 100%                                                             |      |       |        |         |         |         |         |         |         |         |          |           | 2         | 2                                    | 2   | 4   | 4*                          |                           |
| 0        | 27                 | DBC.Owners (DEFAULT) (Map: TD_DATADITIONARYMAP)          |                                     |                                                                  |      |       |        |         |         |         |         |         |         |         |          |           |           |                                      |     |     |                             |                           |
|          | 0                  | N                                                        |                                     | 100%                                                             |      |       |        |         |         |         |         |         |         |         |          |           | 2         | 2                                    | 2   | 4   | 4*                          |                           |
| 1024     |                    | N                                                        |                                     | 100%                                                             |      |       |        |         |         |         |         |         |         |         |          |           | 1         | 1                                    | 2   | 3   | 3*                          |                           |
| 2048     |                    | N                                                        |                                     | 100%                                                             |      |       |        |         |         |         |         |         |         |         |          |           | 1         | 1                                    | 2   | 3   | 3*                          |                           |
| 0        | 30                 | DBC.ErrorMsgs (DEFAULT) (Map: TD_DATADITIONARYMAP)       |                                     |                                                                  |      |       |        |         |         |         |         |         |         |         |          |           |           |                                      |     |     |                             |                           |
|          | 0                  | N                                                        |                                     | 100%                                                             |      |       |        |         |         |         |         |         |         |         |          |           | 2         | 2                                    | 2   | 4   | 4*                          |                           |
| 1024     |                    | N                                                        |                                     |                                                                  |      |       |        |         |         | 100%    |         |         |         |         |          |           | 239       | 244                                  | 248 | 4   | 4*                          |                           |

[illegible]

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

```
"Compression Status" :  C = Fully Compressed
                        U = Fully Uncompressed
                        N = Not Compressible
```

### Example: SHOWBLOCKS medium display scoped to a foreign table

For more information about foreign tables, see CREATE FOREIGN TABLE in *Teradata Vantage™ - SQL Data Definition Language Syntax and Examples*, B035-1144.

```
Ferret ==>
> scope table 0 3123 0
```

The SCOPE has been set

```
Ferret ==>
> showblocks /m
```

[illegible]

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

```
"Compression Status" : C = Fully Compressed
                        U = Fully Uncompressed
                        N = Not Compressible
```

Example: SHOWBLOCKS long display

The following is a portion of the output of the showblocks command using the // long output option.

| Table ID |                 | Compr-<br>ession<br>Status                    | Estimated<br>Compres-<br>sion<br>Ratio | Estimated %<br>of Blocks<br>Un-compressed | Data<br>Block<br>size<br>(sectors) | Number of<br>Data Blocks<br>of indicated<br>size | Cumulative<br>Percent of<br>Total<br>Data Blocks | Statistics of<br>Number of Rows<br>Per Data Block |     |     | Total<br>Number<br>of |
|----------|-----------------|-----------------------------------------------|----------------------------------------|-------------------------------------------|------------------------------------|--------------------------------------------------|--------------------------------------------------|---------------------------------------------------|-----|-----|-----------------------|
|          |                 |                                               |                                        |                                           |                                    |                                                  |                                                  | Min                                               | Avg | Max | Cylinders             |
| 0        | 0               | DBC.FIRSTPERMTABLE(Map: TD_DATADITIONARYMAP)  |                                        |                                           |                                    |                                                  |                                                  |                                                   |     |     |                       |
|          | 0               | N                                             |                                        |                                           | 1                                  | 4                                                | 100.00%                                          | 1                                                 | 1   | 1   | 4*                    |
| 0        | 7               | DBC.RCConfiguration(Map: TD_DATADITIONARYMAP) |                                        |                                           |                                    |                                                  |                                                  |                                                   |     |     |                       |
|          | 0               | N                                             |                                        |                                           | 2                                  | 4                                                | 100.00%                                          | 1                                                 | 1   | 1   | 4*                    |
| 0        | 25              | DBC.Global(Map: TD_DATADITIONARYMAP)          |                                        |                                           |                                    |                                                  |                                                  |                                                   |     |     |                       |
|          | 0               | N                                             |                                        |                                           | 2                                  | 4                                                | 100.00%                                          | 1                                                 | 1   | 1   | 4*                    |
|          | 1024            | N                                             |                                        |                                           | 1                                  | 1                                                | 100.00%                                          | 1                                                 | 1   | 1   | 1*                    |
|          | 2048            | N                                             |                                        |                                           | 1                                  | 1                                                | 100.00%                                          | 1                                                 | 1   | 1   | 1*                    |
| 0        | 26              | DBC.TransientJournal(Map: TD_GLOBALMAP)       |                                        |                                           |                                    |                                                  |                                                  |                                                   |     |     |                       |
|          | 0               | N                                             |                                        |                                           | 2                                  | 4                                                | 100.00%                                          | 1                                                 | 1   | 1   | 4*                    |
| 0        | 27              | DBC.Owners(Map: TD_DATADITIONARYMAP)          |                                        |                                           |                                    |                                                  |                                                  |                                                   |     |     |                       |
|          | 0               | N                                             |                                        |                                           | 2                                  | 4                                                | 100.00%                                          | 1                                                 | 1   | 1   | 4*                    |
|          | 1024            | N                                             |                                        |                                           | 1                                  | 2                                                | 66.67%                                           | 1                                                 | 2   | 2   | 3*                    |
|          |                 | N                                             |                                        |                                           | 2                                  | 1                                                | 100.00%                                          | 34                                                | 34  | 34  |                       |
|          | 2048            | N                                             |                                        |                                           | 1                                  | 2                                                | 66.67%                                           | 1                                                 | 2   | 2   | 3*                    |
|          |                 | N                                             |                                        |                                           | 2                                  | 1                                                | 100.00%                                          | 34                                                | 34  | 34  |                       |
| 49152    | 0               | DBC.FIRSTSPOOLTABLE(Map: TD_MAP1)             |                                        |                                           |                                    |                                                  |                                                  |                                                   |     |     |                       |
|          | 0               | N                                             |                                        |                                           | 1                                  | 4                                                | 100.00%                                          | 1                                                 | 1   | 1   | 4*                    |
| WAL LOG  | WAL LOG.WAL LOG |                                               |                                        |                                           |                                    |                                                  |                                                  |                                                   |     |     |                       |
| WAL LOG  |                 | N                                             |                                        |                                           | 1                                  | 48                                               | 85.71%                                           | 0                                                 | 2   | 4   | 8*                    |
|          |                 | N                                             |                                        |                                           | 2                                  | 1                                                | 87.50%                                           | 0                                                 | 6   | 6   |                       |
|          |                 | N                                             |                                        |                                           | 3                                  | 6                                                | 98.21%                                           | 0                                                 | 7   | 9   |                       |
|          |                 | N                                             |                                        |                                           | 5                                  | 1                                                | 100.00%                                          | 0                                                 | 1   | 1   |                       |

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



## SHOWCOMPRESS

Lists subtables that have block-level compressed data blocks.

### Syntax

```
{ SHOWCOMPRESS | SHOWCOMP } [ /S | /L ]
```

#### Note:

The online help lists the display options (/S and /L) as /dispt.

### Syntax Elements

#### /S

For compressed or compressible tables, displays table ID, table name, and subtable ID.

This is the default display.

#### /L

Displays compression state, compression algorithm name, compression level, exact compression ratio, exact percentage of blocks compressed, exact percentage of data compressed, exact number of used GB, table ID, table name, and subtable ID.

Uncompressed subtable information is displayed only when the table has compressed or compressible sub-tables.

#### Note:

This display option requires longer to display than the /S option because all of the data blocks of the tables in scope must be examined.

The output shows blocks that are compressed, uncompressed, and "disqualified". Disqualified blocks are those that did not qualify for compression for one of the following reasons:

- Some data blocks fall below BLC compression threshold levels, which are set in the Compression group of DBS Control fields. For more information about these compression settings, see [DBS Control \(dbscontrol\)](#).
- Some data blocks are too small to be compressed.
- Some special subtables are never compressed.

### Usage Notes

Block-level compression (BLC) compresses table data at the File System level to save storage space. SHOWCOMPRESS lists all tables that have data blocks compressed using BLC.

SHOWCOMPRESS shows compression data only for table types eligible for BLC, even if the Ferret scope is set to encompass a wider set of table types.

BLC can be controlled at many levels of Vantage:

- The Compression and Storage field groups in the DBS Control utility determine whether Vantage applies BLC automatically to different categories of tables. You can use these settings to have data compressed automatically when the tables are loaded or updated, to have some categories of tables never be compressed, or to tell Vantage to compress only infrequently accessed ("cold") data using temperature-based block-level compression (TBBLC).
- You can manually set BLC for individual tables using the BLOCKCOMPRESSION option of the CREATE TABLE and ALTER TABLE SQL statements.
- You can use the BLOCKCOMPRESSION query band to override other BLC settings for an entire Teradata session or as part of a data load script.
- Use the Ferret [COMPRESS](#) and [UNCOMPRESS](#) commands to compress and uncompress tables on-demand.

SHOWCOMPRESS shows table-level BLC compression information. For block-level BLC compression information, use the [SHOWBLOCKS](#) command.

Example: SHOWCOMPRESS /L output

In the following example:

- Table tab1 has the primary and fallback subtables compressed with IPPZLIB algorithm.
- Table tab2 has the primary subtable compressed with the IPPZLIB algorithm, and the fallback subtable compressed with the ZLIB algorithm.
- Table tab3 has primary and fallback subtables both compressed with the IPPZLIB algorithm. They are compressed to different compression levels (degrees).

```
Ferret ==>
showcompress /L
```

SHOWCOMPRESS has been started on all AMP vprocs.

| Table ID | Tablename (BLC option, BLC ALG, BLC LEVEL)      | Mapname | Compression state | Compression algorithm | Compression level | Exact compression ratio | Exact % of blocks | Exact % of data | Exact used GB |
|----------|-------------------------------------------------|---------|-------------------|-----------------------|-------------------|-------------------------|-------------------|-----------------|---------------|
| 0 2930   | EXACT_COMPSTATS.tab1 (MANUAL, DEFAULT, DEFAULT) | TD_MAP1 |                   |                       |                   |                         |                   |                 |               |
|          |                                                 |         | COMPRESSED        | IPPZLIB               | 6                 | 98.70%                  | 98.58%            | 90.42%          | 0.001         |
|          |                                                 |         | DISQUALIFIED      | N/A                   | N/A               | 0.00%                   | 0.57%             | 4.60%           | 0.000         |
|          |                                                 |         | UNCOMPRESSED      | N/A                   | N/A               | 0.00%                   | 0.85%             | 4.98%           | 0.000         |
|          |                                                 |         | UNCOMPRESSED      | N/A                   | N/A               | 0.00%                   | 100.00%           | 100.00%         | 0.000         |
|          |                                                 |         | COMPRESSED        | IPPZLIB               | 6                 | 98.96%                  | 100.00%           | 100.00%         | 0.001         |
|          |                                                 |         | UNCOMPRESSED      | N/A                   | N/A               | 0.00%                   | 100.00%           | 100.00%         | 0.000         |
|          |                                                 |         | UNCOMPRESSED      | N/A                   | N/A               | 0.00%                   | 100.00%           | 100.00%         | 0.000         |
|          |                                                 |         | COMPRESSED        | IPPZLIB               | 6                 | 77.84%                  | 100.00%           | 100.00%         | 0.000         |
|          |                                                 |         | COMPRESSED        | IPPZLIB               | 6                 | 74.26%                  | 100.00%           | 100.00%         | 0.000         |
|          |                                                 |         | COMPRESSED        | IPPZLIB               | 6                 | 98.97%                  | 100.00%           | 100.00%         | 0.001         |
|          |                                                 |         | DISQUALIFIED      | N/A                   | N/A               | 0.00%                   | 100.00%           | 100.00%         | 0.000         |
|          |                                                 |         | DISQUALIFIED      | N/A                   | N/A               | 0.00%                   | 100.00%           | 100.00%         | 0.000         |
| 0 2932   | EXACT_COMPSTATS.tab2 (MANUAL, DEFAULT, DEFAULT) | TD_MAP1 |                   |                       |                   |                         |                   |                 |               |
|          |                                                 |         | COMPRESSED        | ZLIB                  | 1                 | 98.67%                  | 49.25%            | 45.37%          | 0.001         |
|          |                                                 |         | COMPRESSED        | IPPZLIB               | 6                 | 98.69%                  | 49.32%            | 45.09%          | 0.001         |
|          |                                                 |         | DISQUALIFIED      | N/A                   | N/A               | 0.00%                   | 0.57%             | 4.59%           | 0.000         |
|          |                                                 |         | UNCOMPRESSED      | N/A                   | N/A               | 0.00%                   | 0.85%             | 4.96%           | 0.000         |
|          |                                                 |         | UNCOMPRESSED      | N/A                   | N/A               | 0.00%                   | 100.00%           | 100.00%         | 0.000         |
|          |                                                 |         | COMPRESSED        | IPPZLIB               | 6                 | 98.96%                  | 100.00%           | 100.00%         | 0.001         |
|          |                                                 |         | UNCOMPRESSED      | N/A                   | N/A               | 0.00%                   | 100.00%           | 100.00%         | 0.000         |
|          |                                                 |         | UNCOMPRESSED      | N/A                   | N/A               | 0.00%                   | 100.00%           | 100.00%         | 0.000         |
|          |                                                 |         | COMPRESSED        | IPPZLIB               | 6                 | 77.35%                  | 100.00%           | 100.00%         | 0.000         |
|          |                                                 |         | COMPRESSED        | IPPZLIB               | 6                 | 73.69%                  | 100.00%           | 100.00%         | 0.000         |
|          |                                                 |         | COMPRESSED        | ZLIB                  | 1                 | 98.96%                  | 100.00%           | 100.00%         | 0.001         |
|          |                                                 |         | DISQUALIFIED      | N/A                   | N/A               | 0.00%                   | 100.00%           | 100.00%         | 0.000         |
| 0 2934   | EXACT_COMPSTATS.tab3 (MANUAL, DEFAULT, DEFAULT) | TD_MAP1 |                   |                       |                   |                         |                   |                 |               |
|          |                                                 |         | DISQUALIFIED      | N/A                   | N/A               | 0.00%                   | 100.00%           | 100.00%         | 0.000         |
|          |                                                 |         | COMPRESSED        | ZLIB                  | 1                 | 76.35%                  | 100.00%           | 100.00%         | 0.000         |
|          |                                                 |         | COMPRESSED        | ZLIB                  | 1                 | 71.92%                  | 100.00%           | 100.00%         | 0.000         |



## 12: Ferret Utility (ferret)

|      |  |  |              |         |     |        |         |         |       |
|------|--|--|--------------|---------|-----|--------|---------|---------|-------|
|      |  |  | COMPRESSED   | IPPZLIB | 1-6 | 98.97% | 40.00%  | 47.73%  | 0.000 |
|      |  |  | DISQUALIFIED | N/A     | N/A | 0.00%  | 40.00%  | 29.55%  | 0.000 |
|      |  |  | UNCOMPRESSED | N/A     | N/A | 0.00%  | 20.00%  | 22.73%  | 0.000 |
| 0    |  |  | UNCOMPRESSED | N/A     | N/A | 0.00%  | 100.00% | 100.00% | 0.000 |
| 1024 |  |  | COMPRESSED   | IPPZLIB | 6   | 98.97% | 100.00% | 100.00% | 0.000 |
| 1028 |  |  | UNCOMPRESSED | N/A     | N/A | 0.00%  | 100.00% | 100.00% | 0.000 |
| 1032 |  |  | UNCOMPRESSED | N/A     | N/A | 0.00%  | 100.00% | 100.00% | 0.000 |
| 1792 |  |  | DISQUALIFIED | N/A     | N/A | 0.00%  | 100.00% | 100.00% | 0.000 |
| 1794 |  |  | DISQUALIFIED | N/A     | N/A | 0.00%  | 100.00% | 100.00% | 0.000 |
| 2048 |  |  | COMPRESSED   | IPPZLIB | 1   | 98.97% | 100.00% | 100.00% | 0.000 |
| 2052 |  |  | DISQUALIFIED | N/A     | N/A | 0.00%  | 100.00% | 100.00% | 0.000 |
| 2056 |  |  | DISQUALIFIED | N/A     | N/A | 0.00%  | 100.00% | 100.00% | 0.000 |
| 2816 |  |  | DISQUALIFIED | N/A     | N/A | 0.00%  | 100.00% | 100.00% | 0.000 |
| 2818 |  |  | DISQUALIFIED | N/A     | N/A | 0.00%  | 100.00% | 100.00% | 0.000 |

**Related Information**

| For more information on...                                             | See...                                                                                   |
|------------------------------------------------------------------------|------------------------------------------------------------------------------------------|
| Block-level compression                                                | <i>Teradata Vantage™ - Database Design</i> , B035-1094.                                  |
| BLOCKCOMPRESSION table attribute, CREATE TABLE, ALTER TABLE statements | <i>Teradata Vantage™ - SQL Data Definition Language Syntax and Examples</i> , B035-1144. |
| Query bands for BLC and TBLC                                           | <i>Teradata Vantage™ - SQL Data Definition Language Detailed Topics</i> , B035-1184.     |
| Compression and Storage field groups in DBS Control                    | <a href="#">DBS Control (dbscontrol)</a> .                                               |

## 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

## SHOWDEFAULTS

Shows the default settings and the current context. The command displays the current default radix for input and output, the current input, output, and error file names, and the SCOPE defined. SHOWDEFAULTS also displays WAL log context information.

### Syntax

```
{ SHOWDEFAULTS | SHOWD } [ /L ]
```

### Syntax Elements

/L

When the SCOPE is set to a table that uses a sparse map, /L displays the complete list of AMP vprocs in the sparse map.

### Example: SHOWDEFAULTS output

```
The current setting of the Input Radix is Decimal
The current setting of the Output Radix is Decimal
Ferret is running with Script Mode -- Disabled
```

```
Scope for the Defrag command is :
  All Cylinders
  On All AMP vprocs
```

```
Scope for the Packdisk command is :
  All of the AMP vprocs
```

```
Scope for the Scandisk command is :
  All of the AMP vprocs
  WAL Log
  All Tables
```

```
Scope for the Showblocks command is :
  All Tables
  WAL Log
  All of the AMP vprocs
```

```
Scope for the Showspace command is :
  All Tables
  WAL Log
  All of the AMP vprocs
```

Scope for the ShowFSP command is :

- All Tables

- All of the AMP vprocs

Scope for the Showwhere command is :

- All Tables

- WAL Log

- All of the AMP vprocs

Scope for the Showcylalloc command is :

- All of the AMP vprocs

Scope for the Force command is :

- All of the AMP vprocs

Scope for the Showcompress command is:

- Class ALL | PERM | JRNL

- All of the AMP vprocs

Scope for the [un]Compress command is :

- All of the AMP vprocs

## SHOWFSP

The SHOWFSP command displays information about table storage space utilization. SHOWFSP provides the following information on a per-vproc (per-AMP) basis:

- Table name
- Free space percentage (FSP), the amount of unused storage space on the cylinders used by the table
- Number of cylinders currently occupied by tables meeting SHOWFSP filtering criteria
- Number of cylinders that would be occupied by each of these tables after a PACKDISK command

SHOWFSP shows the number of cylinders that will be recovered or consumed if PACKDISK is run to achieve a given percentage of free space on table cylinders.

The scope of SHOWFSP depends on the most recent options set by the SCOPE command, and can include one or more specific tables, one or more specific vprocs, or all tables in the system.

### Syntax

```
{ SHOWFSP | SHOWF }
  [ /Y ]
  [ -A ]
  [ -C cyls ]
  [ -D fsp ]
  [ -M fsp ]
  [ -R fsp ]
  [ -V ]
```

### Syntax Elements

#### /Y

Prevents Ferret from displaying a confirmation message.

#### -A

All tables option displays all tables meeting the criteria of -C, -D, and -M options. If you do not specify any of those options, they will use their default values. The -R option is always automatically overridden and set to the most negative integer known to Ferret.

A table must occupy at least one full cylinder or be the first table on a cylinder to be considered for display.

The -A option causes PACKDISK to consider both of these cases:

- Where PACKDISK would tighten the packing (and thus free up cylinders)
- Where PACKDISK would loosen the packing (and thus consume free cylinders).

If you do not specify the -A option, PACKDISK considers only tables that would tighten the packing.

The number of cylinders affected by the PACKDISK are displayed as a positive or negative value, where positive indicates cylinders are freed and negative indicates cylinders are consumed.

You can qualify tables further by specifying the -A option in conjunction with the other options.

### **-C *cyls***

The number of cylinders per AMP that must be exceeded by a table to be considered for display.

The -C *cyls* option follows this priority order:

- If -C *cyls* is specified, use it unless *cyls* is 0, in which case *cyls* defaults to 1 automatically.
- If -C *cyls* is not specified, it automatically defaults to 0.

You can further qualify tables by specifying the -C option in conjunction with the other options.

### **-D *fsp***

The desired FSP after packing.

The default follows this priority order:

- If -D *fsp* is specified, use it.
- If -D *fsp* is not specified and a table-level FSP exists, use the table-level FSP.

Otherwise, use the system-wide default FSP as defined in the DBS Control GDO.

---

#### **Note:**

This option allows SHOWFSP to display tables that would free up cylinders on each AMP if you ran PACKDISK with this FSP. If specified in conjunction with the -A option, SHOWFSP also considers tables that consume free cylinders. You can qualify tables further by specifying the -D *fsp* option in conjunction with the other options.

---

### **-M *fsp***

minimum current FSP a table must exceed to qualify for display.

The default is 0.

---

#### **Note:**

This option allows SHOWFSP to ignore tables whose current average FSP is under a certain FSP value.

---

The average is calculated as a floating point value, and the option is input as an integer. Therefore, often the FSP appears to act as a minimum. For example, an average FSP of 10.1 is greater than a minimum FSP of 10, whereas an average FSP of 10 is not greater than the

minimum FSP of 10. Therefore, a table with an average FSP of 10.1 qualifies, whereas a table with an average FSP of 10 would not qualify.

You can qualify tables further by specifying the `-M fsp` option in conjunction with the other options.

### **-R *cy/s***

the number of recoverable cylinders that must be exceeded to qualify the table for display.

The default follows this priority order:

- If the `-A` option is specified, set `-R cy/s` to the most negative integer known to Ferret.
- If `-R cy/s` is specified, use it unless `cy/s` is 0, in which case `cy/s` is defaulted to 1 automatically.
- If `-R cy/s` is not specified, it is defaulted to 0 automatically.

You can qualify tables further by specifying the `-R cy/s` option in conjunction with the other options, except for the `-A` option, in which case the `-R cy/s` option is handled in the manner noted above.

### **-V**

verbose mode.

The default is Off.

This is for debugging purposes and produces extremely detailed and voluminous output. Teradata does not recommend this option for normal use.

## **Usage Notes**

All arguments and displays are on a per-AMP basis. Without parameters, SHOWFSP displays all tables with recoverable cylinders, assuming the defaults used by PACKDISK. For more details as to how the defaults of each option are determined, see their respective sections above.

SHOWFSP provides an approximation of the number of cylinders that can be freed or consumed by PACKDISK. The actual number of cylinders freed or consumed by a PACKDISK operation may be slightly different.



## Example: SHOWFSP output

The following example shows the basic output of SHOWFSP.

```
Wed Mar 01, 2017 20:33:07 : ShowFsp has been started
                          On all AMP vprocs
```

```
vproc 0 (0000) response
```

```
There are 4 tables larger than 0 cylinders on amp 0
```

| Database Name | Cylinders | Table Name | Map Name            | fsp % | Recoverable Current |
|---------------|-----------|------------|---------------------|-------|---------------------|
| DBC           | 1*        | TVFields   | TD_DATADITIONARYMAP | 85    |                     |

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

```
vproc 2 (0002) response
```

```
There are 4 tables larger than 0 cylinders on amp 2
```

| Database Name | Cylinders | Table Name      | Map Name            | fsp % | Recoverable Current |
|---------------|-----------|-----------------|---------------------|-------|---------------------|
| SYSLIB        | 1*        | FixCurrentSpace | TD_DATADITIONARYMAP | 64    |                     |

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

```
vproc 3 (0003) response
```

```
There are 4 tables larger than 0 cylinders on amp 3
```

| Database Name | Cylinders | Table Name | Map Name | fsp % | Recoverable Current |
|---------------|-----------|------------|----------|-------|---------------------|
|---------------|-----------|------------|----------|-------|---------------------|

```

Cylinders  Cylinders
-----
SYSLIB      FixCurrentSpace      TD_DATADICTIONARYMAP      64
1*          2*

```

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

1 of 4 vprocs responded with no tables fitting the criteria  
 3 of 4 vprocs responded with the above tables fitting the criteria  
 ShowFsp has completed

**Examples: SHOWFSP command syntax examples**

These examples demonstrate how you can use SHOWFSP to answer different questions.

The following SHOWFSP command shows all tables that would free up or consume any amount of cylinders per AMP if executing a PACKDISK command with an FSP of 10%.

```
showfsp -a -d 10
```

The following SHOWFSP command shows all tables that would free up more than one cylinder per AMP if executing a PACKDISK with an FSP of 10%.

```
showfsp -d 10 -r 1
```

The following SHOWFSP command shows all tables that would free up or consume any amount of cylinders per AMP if executing a PACKDISK command with an FSP of 5%. It considers only tables which currently occupy more than 10 cylinders per AMP and have a current FSP greater than 5%.

```
showfsp -a -d 5 -c 10 -m 5
```

The following SHOWFSP command shows all tables that would free up more than three cylinders per AMP if executing a PACKDISK command with an FSP of 7%. It considers only tables with a current FSP of more than 10%, and those that currently occupy more than three cylinders per AMP.

```
showfsp -d 7 -c 3 -m 10 -r 3
```

## SHOWSPACE

The SHOWSPACE command shows storage space utilization. It displays the following information based on the currently specified Ferret scope. For more information about Ferret command scope, see [SCOPE](#).

- Number of cylinders allocated for permanent, journal, temporary, spool, DEPOT, and WAL log data.
- Average utilization per cylinder for permanent, journal, temporary, DEPOT, and spool cylinders.
- Number and percentage of available free cylinders.

### Syntax

```
{ SHOWSPACE | SHOWS } [ /S | /M | /L ]
```

### Syntax Elements

**/S**

Displays a one-line summary of space information for all AMPs in the current Ferret scope. Use this option for systems with large numbers of AMPs. This is the default display option.

**/M**

**/L**

Shows space information for the AMPs in the current Ferret scope, displaying one line for each AMP.

### Usage Notes

Although disk space allocated for TJ and WAL records is charged against the table 0 26, no actual TJ or WAL records are found in the subtables of this table. Instead, these records are in the WAL log. The only row that exists in any subtable of the table 0, 26 is the table header in subtable 0.

Example: Using SHOWSPACE /S to display storage space utilization summary

The following output is from a large cylinder system.

```
Ferret ==>
showspace /s
```

SHOWSPACE results for Scoped AMP(s)

| Vproc<br>Num | DSU | total<br>Avail<br>Cyls | Perm Data Cyls           |                                 |      | Wal Cyls                        |      | Depot Cyls               |      | Spool Cyls               |                                 |      | Temp Cyls                |                                 |      | Jrnl Cyls                |                                 |      | Pspool Cyls              |                                 |      | Free Cyls                       |       |
|--------------|-----|------------------------|--------------------------|---------------------------------|------|---------------------------------|------|--------------------------|------|--------------------------|---------------------------------|------|--------------------------|---------------------------------|------|--------------------------|---------------------------------|------|--------------------------|---------------------------------|------|---------------------------------|-------|
|              |     |                        | Avg<br>utl<br>per<br>Cyl | % of<br>total<br>Avail<br>Space | #Cyl | % of<br>total<br>Avail<br>Space | #Cyl | Avg<br>utl<br>per<br>Cyl | #Cyl | Avg<br>utl<br>per<br>Cyl | % of<br>total<br>Avail<br>Space | #Cyl | Avg<br>utl<br>per<br>Cyl | % of<br>total<br>Avail<br>Space | #Cyl | Avg<br>utl<br>per<br>Cyl | % of<br>total<br>Avail<br>Space | #Cyl | Avg<br>utl<br>per<br>Cyl | % of<br>total<br>Avail<br>Space | #Cyl | % of<br>total<br>Avail<br>Space | #Cyl  |
| ALL          |     | 38988                  | 59%                      | 0%                              | 16*  | 0%                              | 60   | 0%                       | 12   | 0%                       | 4%                              | 72*  | 0%                       | 0%                              | 4*   | 0%                       | 0%                              | 4*   | 0%                       | 0%                              | 4*   | 98%                             | 38296 |

Average Free Cylinders Across All AMPs in the Current Scope: 90%

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

Example: Using SHOWSPACE /M to display per-AMP storage space utilization

The following output is from a large cylinder system.

```
Ferret ==>
showspace /m
```

SHOWSPACE

results for Scoped AMP(s)

| Vproc<br>Num | DSU | total<br>Avail<br>Cyls | Perm Data Cyls           |                                 |      | Wal Cyls                        |      | Depot Cyls               |      | Spool Cyls               |                                 |      | Temp Cyls                |                                 |      | Jrnl Cyls                |                                 |      | Pspool Cyls                         |      |                                     | Free Cyls |       |
|--------------|-----|------------------------|--------------------------|---------------------------------|------|---------------------------------|------|--------------------------|------|--------------------------|---------------------------------|------|--------------------------|---------------------------------|------|--------------------------|---------------------------------|------|-------------------------------------|------|-------------------------------------|-----------|-------|
|              |     |                        | Avg<br>utl<br>per<br>Cyl | % of<br>total<br>Avail<br>Space | #Cyl | % of<br>total<br>Avail<br>Space | #Cyl | Avg<br>utl<br>per<br>Cyl | #Cyl | Avg<br>utl<br>per<br>Cyl | % of<br>total<br>Avail<br>Space | #Cyl | Avg<br>utl<br>per<br>Cyl | % of<br>total<br>Avail<br>Space | #Cyl | Avg<br>utl<br>per<br>Cyl | % of<br>total<br>Avail<br>Space | #Cyl | % of<br>total<br>Avail<br>Cyl Space | #Cyl | % of<br>total<br>Avail<br>Cyl Space | #Cyl      |       |
| 0            |     | 10810                  | 79%                      | 10%                             | 179* | 0%                              | 6    | 0%                       | 2*   | 0%                       | 0%                              | 4*   | 0%                       | 0%                              | 1*   | 0%                       | 0%                              | 1*   | 0%                                  | 0%   |                                     | 90%       | 9697  |
| 1            |     | 10810                  | 78%                      | 10%                             | 181* | 0%                              | 5    | 0%                       | 2*   | 0%                       | 0%                              | 4*   | 0%                       | 0%                              | 1*   | 0%                       | 0%                              | 1*   | 0%                                  | 0%   |                                     | 90%       | 9686  |
| TOTAL        |     | 21620                  |                          |                                 | 360* |                                 | 11*  |                          | 4*   |                          |                                 | 8*   |                          |                                 | 2*   |                          |                                 | 2*   |                                     |      | 1*                                  |           | 19383 |

Average Free Cylinders Across All AMPs in the Current Scope: 90%

\* Reported in units of Large Cylinders

(A Large Cylinder is 6 times the size of a small cylinder)



## 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.

## SHOWWHERE Example

### Example: Using SHOWWHERE to display cylinder allocation and temperature

```
Ferret ==>
showwhere
```

SHOWWHERE result for entire system

| Vproc<br>Num | # of<br>Cyls | Type        | Grade  | %    | Temperature |       |       | VH |
|--------------|--------------|-------------|--------|------|-------------|-------|-------|----|
|              |              |             |        |      | %HOT        | %WARM | %COLD |    |
|              | 48           | PERM        | FAST   | 100% | 38%         | 54%   | 8%    | 0% |
|              | 0            | PERM        | MEDIUM | 0%   | 0%          | 0%    | 0%    | 0% |
|              | 0            | PERM        | SLOW   | 0%   | 0%          | 0%    | 0%    | 0% |
|              | 17           | WAL         | FAST   | 100% | 100%        | 0%    | 0%    | 0% |
|              | 0            | WAL         | MEDIUM | 0%   | 0%          | 0%    | 0%    | 0% |
|              | 0            | WAL         | SLOW   | 0%   | 0%          | 0%    | 0%    | 0% |
|              | 24           | DEPOT       | FAST   | 100% | 100%        | 0%    | 0%    | 0% |
|              | 0            | DEPOT       | MEDIUM | 0%   | 0%          | 0%    | 0%    | 0% |
|              | 0            | DEPOT       | SLOW   | 0%   | 0%          | 0%    | 0%    | 0% |
|              | 160          | WAL POOL    | FAST   | 100% | 100%        | 0%    | 0%    | 0% |
|              | 0            | WAL POOL    | MEDIUM | 0%   | 0%          | 0%    | 0%    | 0% |
|              | 0            | WAL POOL    | SLOW   | 0%   | 0%          | 0%    | 0%    | 0% |
|              | 0            | SPOOL       | FAST   | 0%   | 0%          | 0%    | 0%    | 0% |
|              | 0            | SPOOL       | MEDIUM | 0%   | 0%          | 0%    | 0%    | 0% |
|              | 0            | SPOOL       | SLOW   | 0%   | 0%          | 0%    | 0%    | 0% |
|              | 8            | SPOOL       | LOCAL  | 100% | 100%        | 0%    | 0%    | 0% |
|              | 0            | SPOOL POOL  | FAST   | 0%   | 0%          | 0%    | 0%    | 0% |
|              | 0            | SPOOL POOL  | MEDIUM | 0%   | 0%          | 0%    | 0%    | 0% |
|              | 0            | SPOOL POOL  | SLOW   | 0%   | 0%          | 0%    | 0%    | 0% |
|              | 136          | SPOOL POOL  | LOCAL  | 100% | 100%        | 0%    | 0%    | 0% |
|              | 0            | GLOBAL TEMP | FAST   | 0%   | 0%          | 0%    | 0%    | 0% |
|              | 1            | GLOBAL TEMP | MEDIUM | 13%  | 0%          | 0%    | 100%  | 0% |
|              | 7            | GLOBAL TEMP | SLOW   | 88%  | 0%          | 0%    | 100%  | 0% |
|              | 8            | JRNL        | FAST   | 100% | 0%          | 25%   | 75%   | 0% |
|              | 0            | JRNL        | MEDIUM | 0%   | 0%          | 0%    | 0%    | 0% |
|              | 0            | JRNL        | SLOW   | 0%   | 0%          | 0%    | 0%    | 0% |
|              | 8            | PSPPOOL     | FAST   | 100% | 100%        | 0%    | 0%    | 0% |
|              | 0            | PSPPOOL     | MEDIUM | 0%   | 0%          | 0%    | 0%    | 0% |
|              | 0            | PSPPOOL     | SLOW   | 0%   | 0%          | 0%    | 0%    | 0% |
| TOTAL        | 2310         |             | FAST   | 92%  | 91%         | 7%    | 2%    | 0% |
|              | 6            |             | MEDIUM | 0%   | 0%          | 0%    | 100%  | 0% |
|              | 42           |             | SLOW   | 2%   | 0%          | 0%    | 100%  | 0% |
|              | 144          |             | LOCAL  | 6%   | 0%          | 0%    | 100%  | 0% |

Reported in units of 32768 sector Cylinders



## TABLEID

The TABLEID command displays the table number of the specified table when given the database name and table name.

### Syntax

```
TABLEID {
  "databasename.tablename" |
  "databasename"."tablename" |
  'databasename.tablename' |
  'databasename'.'tablename'
}
```

### Syntax Elements

#### *databasename*

The name of the database containing the table for which the table number will be displayed.

#### *tablename*

The name of the table for which the table number will be displayed.

---

#### Note:

The database name and table name must be delimited by apostrophes or double quotation marks, as indicated in the syntax diagram.

---

### Usage Notes

A table is identified in the Data Dictionary by a table number (tvm.tvmid). Each table number is unique across the whole system, rather than local to a database. Therefore, a table number uniquely identifies a table in the system.

The TABLEID command displays the table number of the table specified by *databasename* and *tablename*. The output of the TABLEID command is a numeric subtable identifier (*tid*), which consists of three numbers:

- The first two comprise the table number. This pair of numbers is used to uniquely identify a table in the system.
- The third is the *typeandindex* value, which specifies a kind of subtable, such as a table header, data subtable, or a particular index subtable. TABLEID always returns a *typeandindex* value of zero (0), which specifies the table header.

For more information on how to interpret a *tid*, see [Using Ferret Parameters](#).

The following rules apply when specifying *databasename* and *tablename*:

- The period (.) is required to separate the database name from the table name.

- You must use either apostrophes ( ' ) or double quotation marks ( " ) when typing a database name and table name. The results are the same.
- You can specify a fully qualified table name using any one of the methods suggested in the syntax diagram with the following exceptions:

- The object name itself includes an apostrophe, in which case you must enclose the object name within double quotation marks.

Valid examples include the following:

- `tableid "xyz.mark's table"`
- `tableid "xyz"."mark's table"`

Invalid examples include the following:

- `tableid 'xyz.mark's table'`
- `tableid "xyz".'mark's table'`

- The object name has a period, in which case you must type the fully qualifying tablename in the form of "database"."tablename" or 'database'.'tablename'.

Valid examples include the following:

- `tableid "xyz.0's"."mark's table.2.53.00"`
- `tableid 'xyz'.'table.0'`

Invalid examples include the following:

- `tableid "xyz.0's.mark's table.2.53.00"`
- `tableid 'xyz.0's.mark's table.2.53.00'`

### Example: Using TABLEID to display the table ID number

The following example shows output generated by TABLEID:

```
Ferret ==> tableid "mydatabase.mytable"
The table id for MYDATABASE.MYTABLE is
1 1217 0 (0x0001 0x4C1 0x0000)
```

#### Note:

You could get the same results with the following commands:

```
tableid 'mydatabase.mytable'
tableid "mydatabase"."mytable"
tableid 'mydatabase'.'mytable'
```

## TIME

See [DATE/TIME](#).

## UNCOMPRESS

Uncompresses specified types of data in specified databases and tables, or shows an estimate of what results would be if tables were uncompressed. If data of one of the specified types is subsequently added to the table or database, it will not be compressed.

Decompression for a table uses the algorithm specified by the BLOCKCOMPRESSIONALGORITHM and BLOCKCOMPRESSIONLEVEL options that were specified in the CREATE TABLE or ALTER TABLE statement. If either of these options has not been specified for the table, or if the value was specified as DEFAULT, the data is decompressed using the algorithm and level specified by the [CompressionAlgorithm](#) and [CompressionLevel](#) settings in DBS Control.

---

### Note:

BLOCKCOMPRESSIONLEVEL is effective only for the ZLIB compression algorithm.

---

### Syntax

```
UNCOMPRESS [ /Y ] table_unique
  [ PRIMARY | FALLBACK | FALLBACKANDCLOBS | { WITHOUT | ONLY } CLOBS ]
  [ ESTIMATE | E ]
```

### Syntax Elements

**/Y**

Prevents Ferret from displaying a confirmation message.

***table\_unique***

Unique identifier of the table for which data blocks will be uncompressed.

The table to be uncompressed 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'
  - "database\_name.\*"
  - 'database\_name.\*'
  - "database\_name"."\*"
  - 'database\_name'.'\*'

The asterisk denotes all tables in the given database.

- Specify the unique numeric identifier of the table, which consists of two numeric values separated by a space. The table identifier consists of 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 following options specify the type of data to uncompress.

**PRIMARY**

All primary base table data and primary LOB data that is eligible for compression within the specified database or table.

**FALLBACK**

All fallback base table data and fallback LOB data that is eligible for compression within the specified database or table.

**FALLBACKANDCLOBS**

All fallback base table data, fallback LOB data that is eligible for compression, and primary LOB data eligible for compression within the specified database or table.

**WITHOUT CLOBS**

All data except LOB data that is eligible for compression within the specified database or table.

**ONLY CLOBS**

Only the primary and fallback LOB data that is eligible for compression within the specified table. Other table DBs are not affected.

The following option does not uncompress any data:

**ESTIMATE****E**

Estimates data block (DB) sizes that would exist after an uncompress operation is performed, and estimates the CPU usage (time per DB) that would be required. Does not uncompress any data.

Multiply the Table Size Factor shown in the output by the current table size to determine the estimated table size after an uncompress operation.

**Note:**

Estimates are most accurate when the table DB size is at or near the size defined by DATABLOCKSIZE (optionally specified when the table was created or altered), or defined by the PermDBSize setting in DBS Control if no DATABLOCKSIZE was specified for the table.

**Usage Notes**

The UNCOMPRESS command is part of the block-level compression (BLC) feature of Teradata.

BLC enables data compression at the data block (DB) level of the Teradata file system. Compression reduces the amount of storage required for a given amount of data. The BlockLevelCompression field of DBS Control enables and disables BLC.

BLC can be enabled or disabled system-wide, or for various categories of tables. For Permanent data, these include subtables for primary, fallback, eligible LOB data, and join and hash indexes. Secondary indexes on primary rows and fallback copies of secondary indexes are never compressed.

Tables that have their BLOCKCOMPRESSION value set to ALWAYS in the CREATE TABLE or ALTER TABLE statement cannot be uncompressed using the UNCOMPRESS command.

The UNCOMPRESS command consumes storage space as it uncompresses data. To ensure sufficient space is available after an uncompress command completes, use the UncompressReservedSpace setting in DBS Control to specify the amount of space to reserve. The decompression operation stops when the available free space falls below the reserved amount. The table undergoing decompression at the time may remain in a partially compressed state. This will not affect the functioning of the table or database. To resume decompression, free up some storage space, or lower the value of UncompressReservedSpace then reissue the UNCOMPRESS command.

Ferret logs the beginning and end of UNCOMPRESS operations. If the database is reset during the operation, the decompression operation continues as part of the recovery process after the database restarts. However, decompression will continue and complete only on the table that was in the process of being decompressed at the time of the reset. The rest of the database will remain in the state it was prior to the reset.

Only one instance of the UNCOMPRESS command is allowed to run at any time on the system. If you receive an error when you issue the COMPRESS or UNCOMPRESS command, and you are certain there is no other instance of either command currently running, use the RESETBLCTSKCNT command to reset the count of running COMPRESS and UNCOMPRESS tasks to zero.

The command progress percentage reported by the INQUIRE command can change unexpectedly if the table being uncompressed is being modified concurrently by another process.

The PermDBSize and JournalDBSize field settings in DBS Control apply to the uncompressed size of these types of DBs.

If you receive an error when you issue the COMPRESS or UNCOMPRESS command, and you are certain there is no other instance of either command currently running, use the RESETBLCTSKCNT command to reset the count of running COMPRESS and UNCOMPRESS tasks to zero.

### Related Information

| For more information on...       | See...                                                                               |
|----------------------------------|--------------------------------------------------------------------------------------|
| DBS Control compression settings | <a href="#">DBS Control (dbscontrol)</a> .                                           |
| COMPRESS command                 | <a href="#">COMPRESS</a> .                                                           |
| Query Banding                    | <i>Teradata Vantage™ - SQL Data Definition Language Detailed Topics</i> , B035-1184. |
| RADIX command                    | <a href="#">RADIX</a> .                                                              |
| RESETBLCTSKCNT command           | <a href="#">RESETBLCTSKCNT</a> .                                                     |
| TABLEID command                  | <a href="#">TABLEID</a> .                                                            |

# Gateway Control (gtwcontrol)

---

**Note:**

The utility described is not available for as-a-service customers.

---

The Gateway Control utility, gtwcontrol, modifies the default values in the fields of the gateway control Globally Distributed Object (GDO). GDOs are binary files that store Advanced SQL Engine configuration settings. They are distributed to and used by every node in the system. The PDE layer of Advanced SQL Engine ensures that the GDO is consistent across all virtual processors.

There can be multiple gateways per node if the gateways belong to different host groups and listen on different IP addresses.

## Runs From

- Database Window or comparable interface to the Teradata console subsystem, such as cnstern
- Linux command line

For general information on starting the utilities from different interfaces, see [Starting the Utilities](#).

## Syntax

```
gtwcontrol { option | debugging_option } [...]
```

This command displays a list of available options:

```
gtwcontrol -h
```

### *option*

---

**Note:**

Options are case-sensitive.

---

```
{ -a ExternalAuthentication |
  --auditnetsecurity[={yes|no|ct}] |
  -b { socketbuffersize | default | auto } |
  -c connectiontimeout |
  -d |
  -e Eventcnt |
  -F [ OFF | ON ] |
  -f Logfilesize |
```



```

-g Hostnumber |
-h |
-i Initiallothreads |
-j EnableChannelBinding |
-k keepalivetimeout |
-L [ OFF | ON ] |
-m Maximumlothreads |
--monitorlib suboption [,...] |
-n EnableDeprecatedMessages |
-o default |
-p LocalPEPPreferredPercent |
-r IoThreadCheck |
-s Sessions |
--seccpynotsupported suboption [,...] |
--shutdowntimeout Timeoutvalue |
-t Timeoutvalue |
--TLS [disable|enable|require|nolegacy][,trace=yes|no|all] |
-u SendConnectRespNoSecurity |
-v Vprocnumber |
-x RequireConfidentiality |
-z |
-Z
}

```

### ***debugging\_option***

---

#### **Note:**

Use only for debugging the gateway under the direction of Teradata Support Center personnel.

---

#### **Note:**

Options are case-sensitive.

---

```

{ -1 logonname |
  -A | -C | -D | -H | -I | -J | -K | -M |
  -N | -O | -R | -S | -T | -U | -W | -X | -Y
}

```

Unless otherwise noted, changes to gtwcontrol settings take effect immediately.

## Syntax Elements

### **-a ExternalAuthentication**

Enables or disables external authentication, as follows:

| <b>ExternalAuthentication</b> | <b>Description</b>                                              |
|-------------------------------|-----------------------------------------------------------------|
| off                           | Rejects external authentication and accepts traditional logons. |
| on (Teradata default)         | Accepts both external authentication and traditional logons.    |
| only                          | Accepts external authentication and rejects traditional logons. |

For additional information on External Authentication, see *Teradata Vantage™ - Advanced SQL Engine Security Administration*, B035-1100.

### **--auditnetsecurity[={yes|no|ct}]**

Intended for use in security audits.

Changes to this setting affect only sessions that log on after the change.

Allows the gateway to log the level of encryption used by client interfaces that communicate with the gateway, as follows:

| <b>Option</b> | <b>Description</b>                                                                                                                                                                                                                                            |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| yes           | The gateway logs the security level of the first message after the Connect message that logs the session on. Thereafter, the gateway logs any change in the security level of incoming messages.<br>This is the default if you do not specify yes, no, or ct. |
| no            | The gateway does not log the security level of incoming messages.<br>This is the initial setting for this option and the Teradata default. (The gtwcontrol -Z option resets this option to no.)                                                               |
| ct            | The gateway logs the security level of incoming messages that are cleartext security level (messages that are not explicitly a Confidentiality level).                                                                                                        |

This information is recorded in the gateway logs.

### **-b { socketbuffersize | default | auto }**

Specifies the SND and RCV buffer sizes, as follows:

| <b>Option</b>    | <b>Description</b>                                                                                                                            |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| socketbuffersize | Specifies the buffer size in bytes.<br>The valid range is 65588 through 2147483647 bytes.                                                     |
| default          | Specifies to use the default setting. The gateway chooses the default setting that is appropriate for most circumstances. This setting is set |

| Option | Description                                                                                                                                                                                                                                                                                          |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|        | <p>automatically by the Linux auto-tuning feature, but depends on the database software release.</p> <p><b>Note:</b><br/>Unless you are thoroughly familiar with TCP/IP and SND/RCV buffer sizing, you should only change this setting under the direction of Teradata Support Center personnel.</p> |
| auto   |                                                                                                                                                                                                                                                                                                      |

**-c *connectiontimeout***

Controls the logon message timeout in seconds. The Gateway terminates any session for which a message in the logon sequence is not received in a timely manner. The turnaround time for any message during the logon should be less than the value in the *connectiontimeout* setting.

The value ranges from 5 to 3600 seconds.

The Teradata default is 60 seconds.

**-d**

Displays current setting of the Gateway GDO.

**-e *Eventcnt***

Specifies the number of event trace entries.

The Teradata default is 500.

**-F [ OFF | ON ]****Note:**

This option is deprecated. Do not use.

Toggles "append domain names" for authentication schemes in which domain names are required to define user identities uniquely.

The Teradata default is OFF.

For information about authentication methods, see *Teradata Vantage™ - Advanced SQL Engine Security Administration*, B035-1100.

**-f *Logfilesize***

Specifies the maximum log file size.

The valid range is 1000 through 2147483647.

The Teradata default is 5000000.

### **-g *Hostnumber***

Specifies a host group to which the host-specific settings in this invocation of gtwcontrol will be applied. If you do not specify this option, the host settings are applied to all host groups.

*Hostnumber* is an integer from 0 through 1023 that identifies a host group.

The host-specific options are: -a, -b, -c, -i, -k, -m, -r, -s, -t, -A, -F, -C and -T.

### **-h**

Displays help on gtwcontrol options.

### **-i *Initiallothreads***

Specifies the number of threads of each type that are started initially for the processing of LAN messages. When adjusting the number of threads to match the load, the number of threads of each type will never be reduced below this number.

There are two types of threads:

- One handles traffic from the client (that is, TCP/IP connections).
- One handles traffic from the database (that is, the PDE msgsystem).

The Teradata default is 25.

### **-j *EnableChannelBinding***

---

#### **Note:**

This option is intended for use with Teradata Unity.

---

Enables binding TDGSS-API authentication mechanisms to secure channels at lower network layers for those mechanisms that support channel binding. (PROXY is the only mechanism that currently supports channel binding.) Channel binding verifies the endpoints of the lower level network layers to eliminate man-in-the-middle attacks. In the case of the PROXY mechanism, channel binding also makes it more difficult to use stolen certificates to pretend to be a legitimate endpoint.

*EnableChannelBinding* can be yes or no.

For more information on TDGSS, see *Teradata Vantage™ - Advanced SQL Engine Security Administration*, B035-1100.

**-k *keepalivetimeout***

Specifies how long the connection between the gateway and a client remains idle before the operating system begins probing to see if the connection has been lost.

*keepalivetimeout* specifies the time in minutes, and can be any integer from 1 through 120.

When a connection has been idle for the specified number of minutes, the gateway's operating system will send a keepalive message over the connection to see if there is a response from the client's operating system. If there is no response, the gateway's operating system repeats the probe several times.

If there continues to be no response from the client's operating system, the gateway's operating system closes the connection, disconnecting the session using it.

The specific number of probes and the time between probes vary by operating system type. Some systems allow these values to be changed when networking is configured. If these values have not been changed, it typically takes about 10 minutes from the first probe until a dead connection is closed. If the *keepalivetimeout* value is 5, the actual time until the connection is closed is approximately 15 minutes.

If the value is changed, a database restart is necessary for it to take effect.

The Teradata default is 10 minutes.

**-L [ OFF | ON ]**

Toggles enable logons.

The Teradata default is ON.

**-m *MaximumIthreads***

Specifies the maximum number of threads per type. When adjusting the number of threads to match the load, the number of threads of each type will never be increased above this number.

There are two types of threads:

- One handles traffic from the client (that is, TCP/IP connections).
- One handles traffic from the database (that is, the PDE msgsystem).

The Teradata default is 50.

**--monitorlib *suboption* [...]**

Used to control a loadable library for database monitoring. Such libraries are provided by third-party providers of Database Activity Monitoring tools.

| <i>suboption</i> | Description |                               |
|------------------|-------------|-------------------------------|
| load=[yes   no]  | yes         | Loads the monitoring library. |

| <i>suboption</i>     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |               |                                                                                                                                                                                                          |     |                                                                                                                                                                                               |        |                                                                                                                                                                                                                                                    |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                      | <table> <tr> <td>no (default)</td><td>Disables the library if it has been loaded.<br/>If the library has been disabled by setting this value to no, it will not be reloaded by setting this value to yes until after the next database restart.</td></tr> </table>                                                                                                                                                                                                                                                                                                                                                                                                                                                               | no (default)  | Disables the library if it has been loaded.<br>If the library has been disabled by setting this value to no, it will not be reloaded by setting this value to yes until after the next database restart. |     |                                                                                                                                                                                               |        |                                                                                                                                                                                                                                                    |
| no (default)         | Disables the library if it has been loaded.<br>If the library has been disabled by setting this value to no, it will not be reloaded by setting this value to yes until after the next database restart.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |               |                                                                                                                                                                                                          |     |                                                                                                                                                                                               |        |                                                                                                                                                                                                                                                    |
| copy=[no yes verify] | <p>Determines the method used by the gateway to pass database data, as follows:</p> <table> <tr> <td>no (default)</td><td>Passes the original data buffer directly to the monitoring tools.</td></tr> <tr> <td>yes</td><td>Creates a dynamic data buffer, copies the data from the original data buffer to the new buffer, and sends the new data buffer to the monitoring tools of the third-party monitoring provider.</td></tr> <tr> <td>verify</td><td>Implies copy=yes, and causes the gateway to compare the data in the new and original data buffers (after sending the new data buffer to the monitoring tools), to verify that the monitoring tools did not change any data in the new data buffer.</td></tr> </table> | no (default)  | Passes the original data buffer directly to the monitoring tools.                                                                                                                                        | yes | Creates a dynamic data buffer, copies the data from the original data buffer to the new buffer, and sends the new data buffer to the monitoring tools of the third-party monitoring provider. | verify | Implies copy=yes, and causes the gateway to compare the data in the new and original data buffers (after sending the new data buffer to the monitoring tools), to verify that the monitoring tools did not change any data in the new data buffer. |
| no (default)         | Passes the original data buffer directly to the monitoring tools.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |               |                                                                                                                                                                                                          |     |                                                                                                                                                                                               |        |                                                                                                                                                                                                                                                    |
| yes                  | Creates a dynamic data buffer, copies the data from the original data buffer to the new buffer, and sends the new data buffer to the monitoring tools of the third-party monitoring provider.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |               |                                                                                                                                                                                                          |     |                                                                                                                                                                                               |        |                                                                                                                                                                                                                                                    |
| verify               | Implies copy=yes, and causes the gateway to compare the data in the new and original data buffers (after sending the new data buffer to the monitoring tools), to verify that the monitoring tools did not change any data in the new data buffer.                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |               |                                                                                                                                                                                                          |     |                                                                                                                                                                                               |        |                                                                                                                                                                                                                                                    |
| trace=[yes no all]   | <p>Controls the diagnostic trace facility of the monitoring library, as follows:</p> <table> <tr> <td>yes (default)</td><td>Causes the monitoring library to log only error messages.</td></tr> <tr> <td>no</td><td>Causes the monitoring library to log both error and warning messages.</td></tr> <tr> <td>all</td><td>Logs all messages, including errors and warnings, and any other types of messages the monitoring library can provide.</td></tr> </table>                                                                                                                                                                                                                                                                | yes (default) | Causes the monitoring library to log only error messages.                                                                                                                                                | no  | Causes the monitoring library to log both error and warning messages.                                                                                                                         | all    | Logs all messages, including errors and warnings, and any other types of messages the monitoring library can provide.                                                                                                                              |
| yes (default)        | Causes the monitoring library to log only error messages.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |               |                                                                                                                                                                                                          |     |                                                                                                                                                                                               |        |                                                                                                                                                                                                                                                    |
| no                   | Causes the monitoring library to log both error and warning messages.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |               |                                                                                                                                                                                                          |     |                                                                                                                                                                                               |        |                                                                                                                                                                                                                                                    |
| all                  | Logs all messages, including errors and warnings, and any other types of messages the monitoring library can provide.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |               |                                                                                                                                                                                                          |     |                                                                                                                                                                                               |        |                                                                                                                                                                                                                                                    |

### **-n EnableDeprecatedMessages**

Enables deprecated, descriptive logon failure error messages, as follows:

| <i>EnableDeprecatedMessages</i> | Description                                                                                                      |
|---------------------------------|------------------------------------------------------------------------------------------------------------------|
| no (default)                    | Causes Vantage to return only generic logon failure error messages to users who attempt to logon unsuccessfully. |
| yes                             | Returns less secure, more descriptive logon failure error messages.                                              |

Database errors that are returned to users during unsuccessful logon attempts often provide information regarding the cause of the logon failure. This information could pose a security risk by helping unauthorized users gain entry to the system.

Regardless of this setting, more detailed information about logon failures is always logged to the system logs and to the DBC.eventlog system table, which system administrators can use to determine the reasons for specific logon failures. Administrators can also inspect these logs for repeated unsuccessful logon attempts that might indicate attempts to breach system security.

## **-o default**

### **Note:**

The -o option cannot be used with the -g or -v option.

Indicates that the other options specified in this invocation of gtwglobal should be saved as a set of user-defined default values. These defaults take precedence over the Teradata gateway control defaults, and will be used for new host groups and gateway vprocs when the system is reconfigured.

### **Note:**

Host groups and vprocs that existed before the reconfiguration retain their previous settings. To apply the custom defaults to all existing host groups and vprocs, use the -z option.

gtwcontrol -o default can be run multiple times to set individual default values or groups of values. Subsequent runs do not cancel previous runs.

To clear the user-defined defaults and restore the Teradata defaults, use this option with the -Z option.

## **-p LocalPEPreferredPercent**

Determines the Vantage preference or bias for assigning a new session to a local PE (a PE on the node containing the gateway that accepted the logon request) or assigning the session to a remote PE (a PE on a different node).

*LocalPEPreferredPercent* is a measure of how much difference in relative available capacity (as a percentage) is tolerable when deciding to choose a local PE. Higher values result in a greater preference given to assigning new sessions to local PEs. *LocalPEPreferredPercent* is in the range [0, 100]:

| <i>LocalPEPreferredPercent</i> | Description                                                                                                                                                                                                                                  |
|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0 (default)                    | Gives local PEs preference if the remaining available session capacity is identical for the least busy local and remote PEs. If a remote PE has more session capacity available than the local PE, the session is assigned to the remote PE. |
| 100                            | Gives local PEs preference, even if remote PEs have more available session capacity.                                                                                                                                                         |

| <i>LocalPEPreferredPercent</i> | Description                                                                                                                                              |
|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 through 99                   | Gives increasing levels of preference to local PE assignment, so a local PE may get the session even if a remote PE has more session capacity available. |

**-r *IoThreadCheck***

Determines the frequency in minutes that the gateway checks to see if all the threads are busy.

If they are all busy, a new thread of the appropriate type is started unless it will exceed the maximum number of threads set by the -m option.

If more than one thread has not run during the IoThreadCheck period, the gateway stops a thread, unless it will leave fewer threads than are specified by the -i option.

There are two types of threads:

- One handles traffic from the client (that is, TCP/IP connections).
- One handles traffic from the database (that is, the PDE msgsystem).

The Teradata default is 10 minutes.

**-s *Sessions***

Specifies maximum sessions per gateway.

The valid range is 1 through 2147483647.

The Teradata default is 600.

**--seccynotsupported *suboption* [,...]**

Changes to this setting do not affect sessions logged on at the time of the change.

This option allows the gateway to accept logons from older client software or proxies that do not support Teradata network security policy, even when security policy applies. Additionally, it allows you to have the gateway log messages that identify these older clients or proxies. You can use these log messages to help identify older clients that should be replaced or upgraded.

Proxies are special clients that use the TDGSS PROXY authentication mechanism to communicate with Vantage on behalf of other clients. Currently, Teradata® Unity™ is the only proxy.

| <i>suboption</i>            | Description                                                                                                                                |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| logon=[no all client proxy] | no (default)<br>The gateway does not allow logons using clients or proxies that are unable to support security policy when policy applies. |
|                             | all<br>The gateway allows logons using clients or proxies that are unable to support security policy.                                      |



| suboption                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                     |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                           | client                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | The gateway allows logons using clients that are unable to support security policy, but does not allow logons using proxies that are unable to support security policy when policy applies.                                                                                                                                                                                         |
|                           | proxy                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | The gateway allows logons using proxies that are unable to support security policy when policy applies. Because such proxies provide no information about the security policy capabilities of the clients that connect through them, logons using these clients are allowed, whether the client supports security policy or not.                                                    |
|                           | <p><b>Note:</b></p> <p>For logon=all or logon=proxy, clients can logon through proxies that do not support security policy. These proxies cannot guarantee that the clients follow policy, nor can they transmit policy to clients that could otherwise follow it. For this reason, all clients that log on through such proxies must be manually configured to be within policy, even if they are otherwise capable of following policy automatically. In practice, the gateway can identify security violations by client sessions logged on through such a proxy and log them off, but not until after a single out-of-policy message has already been sent.</p> <p>For logon=all or logon=client, a client that has not been manually configured to be within policy can send a single out-of-policy message per session before the security violation is caught and the session is logged off.</p> |                                                                                                                                                                                                                                                                                                                                                                                     |
| log=[no all client proxy] | no<br>(default)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | The gateway does not log a message to the gateway log files to identify older clients or proxies that are unable to support security policy.                                                                                                                                                                                                                                        |
|                           | all                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | The gateway logs a message in a gateway log file when an attempt is made to log on using a client or proxy that is unable to support security policy.                                                                                                                                                                                                                               |
|                           | client                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | <p>The gateway logs a message in a gateway log file when an attempt is made to log on using a client that is unable to support security policy.</p> <p>The gateway does not log a message if a logon attempt uses a proxy that is unable to support security policy, because such a proxy is incapable of telling the gateway when the client does not support security policy.</p> |
|                           | proxy                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | The gateway logs a message in a gateway log file when an attempt is made to log on through a proxy that is unable to support security policy.                                                                                                                                                                                                                                       |

**--shutdowntimeout *Timeoutvalue***

Sets the amount of time a client is allowed to take after the gateway does a partial TCP/IP socket close until the client must complete the close. The gateway does an abortive close to preemptively free the socket if the client does not complete the close in time.

*Timeoutvalue* is a value from 5 through 3600 seconds. The Teradata default is 60 seconds.

**Note:**

The default value is suitable for most situations. Before you change this setting, consult with Teradata Support Center personnel.

**-t *Timeoutvalue***

Determines how long a disconnected session has to reconnect in minutes. If the client has not reconnected within the specified time period, the client is logged off automatically.

**Note:**

During this time period, the session still counts against the number of sessions allocated to a PE.

The Teradata default is 20 minutes.

**--TLS [disable|enable|require|nolegacy] [,trace=no|yes|all]**

Configures TLS and turns on diagnostic trace.

**Note:**

Changes to the `disable|enable|require|nolegacy` setting take effect after the next database restart.

| Option  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| enable  | <p>Default. The gateway listens to both the HTTPS port (default is 443) and the legacy port (default is 1025) and accepts a new TLS connection through the HTTPS port and a new legacy connection through the legacy port.</p> <p><b>Note:</b></p> <p>If the TLS flag is enabled, but there is not a valid certificate-private key pair installed on the node, the gateway will not be able to listen to the HTTPS port until a valid certificate-private key pair is installed.</p> |
| disable | The gateway does not listen to the HTTPS port. It only listens to the legacy port and accepts a new legacy connection through the legacy port.                                                                                                                                                                                                                                                                                                                                       |
| require | The gateway listens to both the HTTPS port and the legacy port and only accepts a new TLS connection through the HTTPS port. The gateway returns                                                                                                                                                                                                                                                                                                                                     |

| Option               | Description                                                                                                                                                                                                                                                                                                    |              |                                            |     |                                                         |     |                                                                                      |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|--------------------------------------------|-----|---------------------------------------------------------|-----|--------------------------------------------------------------------------------------|
|                      | an error to the client application if it receives a legacy connection request from the legacy port.                                                                                                                                                                                                            |              |                                            |     |                                                         |     |                                                                                      |
| nolegacy             | The gateway only listens to the HTTPS port and accepts a new TLS connection through the HTTPS port. The gateway no longer listens to the legacy port.                                                                                                                                                          |              |                                            |     |                                                         |     |                                                                                      |
| trace=no   yes   all | <table> <tr> <td>no (default)</td><td>The gateway only logs severe error events.</td></tr> <tr> <td>yes</td><td>The gateway logs both error and informational messages.</td></tr> <tr> <td>all</td><td>The gateway logs all messages, including errors, informational, and diagnostic logs.</td></tr> </table> | no (default) | The gateway only logs severe error events. | yes | The gateway logs both error and informational messages. | all | The gateway logs all messages, including errors, informational, and diagnostic logs. |
| no (default)         | The gateway only logs severe error events.                                                                                                                                                                                                                                                                     |              |                                            |     |                                                         |     |                                                                                      |
| yes                  | The gateway logs both error and informational messages.                                                                                                                                                                                                                                                        |              |                                            |     |                                                         |     |                                                                                      |
| all                  | The gateway logs all messages, including errors, informational, and diagnostic logs.                                                                                                                                                                                                                           |              |                                            |     |                                                         |     |                                                                                      |

**-u SendConnectRespNoSecurity**

Specifies whether the gateway sends connection responses encrypted or cleartext, as follows:

| <i>SendConnectRespNoSecurity</i> | Description                                                  |
|----------------------------------|--------------------------------------------------------------|
| no (default)                     | The logon response is encrypted.                             |
| yes                              | The logon response is in cleartext (unencrypted plain text). |

**Note:**

Teradata recommends that you use the default setting unless you use third-party activity-monitoring software that requires access to the contents of the connection responses.

**-v Vprocnumber**

Specifies a vproc to which the vproc-specific settings in this invocation of gtwcontrol will be applied. If you do not specify this option, the vproc-specific settings apply to all vprocs.

*Vprocnumber* is an integer from 0 through 30719 that identifies a vproc.

The vproc-specific options are: -C, -D, -E, -H, -J, -K, -M, -O, -R, -S, -W, and -Y.

**-x RequireConfidentiality****Note:**

Changes to this setting affect only sessions initiated after the change. To ensure that encryption is enforced on all sessions, Teradata recommends that the Teradata system be in a quiescent state (no users logged on) when -x is changed to yes.

Determines whether the gateway requires that input messages be encrypted. The output from the gateway matches the security level of the input it receives, as follows:

| <i>RequireConfidentiality</i> | <i>Description</i>                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| no (default)                  | Does not require that input messages be encrypted.                                                                                                                                                                                                                                                                                                                                                                                        |
| yes                           | Requires input messages to be encrypted. The messages will automatically be encrypted by a client that supports the Enforce Network Security Policy feature, see Security Administration. Gateway will automatically force a session off if a message is received that is not encrypted.<br>The following message types will be accepted, even if they are not encrypted: test, abort, assign, reassign, methods, SSO, logoff, or config. |

**-Z**

Sets gateway control to apply the user-defined defaults created with the `-o default` option to all current host groups and vprocs.

**-Z**

Sets gateway control to apply the original Teradata defaults to all current host groups and vprocs.

If a set of user-defined defaults, created with the `-o default` option exist, they will still be applied to new host groups and vprocs after a reconfiguration. To reset these user-defined defaults to the original Teradata defaults, so new hosts and vprocs will use the original Teradata defaults, use the `-Z` option in conjunction with the `-o default` option:

```
gtwcontrol -o default -Z
```

---

**Note:**

The following options should be used only for debugging the gateway under the direction of Teradata Support Center personnel.

---

**-1 logonname**

For remote gateway global access.

**-A**

Toggles assign tracing. The Teradata default is OFF.

**-C**

Toggles connection tracing. The Teradata default is OFF.

**-D**

Toggles no gtwdie. The Teradata default is OFF.

**-E**

Toggles event trace. The Teradata default is OFF.

The E event trace does not log the actions.

**-H**

Toggles connect heap trace. The Teradata default is OFF.

**-I**

Toggles interactive mode. The Teradata default is OFF.

**-J**

Toggles log LAN errors. The Teradata default is OFF.

Logs any LAN-related errors even when properly handled by the gateway.

**-K**

Toggles session ctx lock trace. The Teradata default is OFF.

This option shows the session locking to make the session context multiprocessor safe.

**-M**

Toggles message tracing. The Teradata default is OFF.

**-N**

Toggles logging of security mechanism selection by TDNEGO. Used for troubleshooting if TDNEGO is choosing the wrong security mechanism. The Teradata default is OFF.

**-O**

Toggles output LAN header on errors. The Teradata default is OFF.

Causes an error message to be written to the gateway log file.

**-R**

Toggles xport log all. The Teradata default is OFF.

By default, the xport trace does not log every LAN operation. The xport log all options causes all LAN operations to be logged.

This option only takes effect if the X trace is on.

**-S**

Toggles the action log. The Teradata default is OFF.

The S option turns on the action trace. The S option only takes effect if the E trace is on.

**-T**

Toggles allow gateway testing. The Teradata default is OFF.

**-U**

Toggles tdgss trace. The Teradata default is OFF.

**Note:**

The -U option causes tdgss-related errors to be logged into the gateway log file for the purpose of diagnosing problems.

**-W**

Toggles wait for debugger to attach. The Teradata default is OFF.

**-X**

Toggles xport trace. The Teradata default is OFF.

**-Y**

Toggles handle trace. The Teradata default is OFF.

## Usage Notes

For convenience, multiple single-hyphen options that do not require values can be passed after a single hyphen. For example:

```
gtwcontrol -dACEX --auditnetsecurity
```

When options require values, each option must have a hyphen:

```
gtwcontrol -a off -k 20
```

For example, this command sets the maximum number of sessions for host group 1 to 600:

```
gtwcontrol -g 1 -s 600
```

## Gateway Host Groups

Each gateway runs in its own vproc. All gateways that belong to the same host group (HG) defined in the vconfig file have a single Assign Task that manages a set of PEs. The set of PEs that the Assign Task manages is determined by using the host number (HN) in the vconfig file and mapping the HN to the HN defined during a system configuration or reconfiguration. The Assign Task is responsible for assigning a new session to the Parsing Engine (PE) having the least number of sessions associated with that PE.

### Example: Gateway Host Groups

This example represents a system running multiple gateways on each node.

| NODE 1                               | NODE 2                               |
|--------------------------------------|--------------------------------------|
| PE 16383    HN = 1                   | PE 16382    HN = 1                   |
| PE 16384    HN = 2                   | PE 16381    HN = 2                   |
| Gateway (8192) HGID 1<br>192.168.1.1 | Gateway (8191) HGID 1<br>192.168.1.2 |
| Gateway (8190) HGID 2<br>192.168.1.3 | Gateway (8189) HGID 2<br>192.168.1.4 |

Each gateway must have a separate set of IP addresses. The gateway running in vproc 8192 only looks for network connections on IP 192.168.1.1 port 1025. The gateway in vproc 8190 only looks for network connections on IP 192.168.1.3 port 1025. This configuration does not require separate LAN cards, but it does require that the IP addresses be unique.

In this scenario, the network administrator creates the following host names/IP entries:

```
hgid1cop1 192.168.1.1
hgid1cop2 192.168.1.2
hgid2cop1 192.168.1.3
hgid2cop2 192.168.1.4
```

If you submit very similar SQL requests and have a separate host group to process those requests, you will get a better cache hit rate. By controlling where the PEs and gateways for a host group are located, and controlling which jobs go to which host groups, you can balance the gateway and PE workload better.

Configuring multiple host groups such that each node has gateways in multiple groups, as shown in the example, helps maintain connectivity if a node goes down. In such a situation, PEs from all host groups will remain available to accept connections.

## Gateway Log Files

The gateway log files are located in /var/opt/teradata/tdtemp/gtw.

## Log File Naming Conventions

Gateway Control uses the following conventions to generate log file names.

```
Gtw_vvvvv_YYYYMMDDhhmmss.log
```

| Name element | Meaning                                                                                           |
|--------------|---------------------------------------------------------------------------------------------------|
| Gtw          | Literal Gtw characters. All Gateway Control log files have names beginning with Gtw.              |
| vvvvv        | Gateway vproc number.                                                                             |
| YYYY         | The four-digit year.                                                                              |
| MM           | Number of the month within the year.                                                              |
| DD           | Number of the day within the month.                                                               |
| hh           | Hour of the day (24-hour clock).                                                                  |
| mm           | Minute of the hour.                                                                               |
| ss           | Second of the minute.                                                                             |
| log          | Literal log characters. All Gateway Control log files have names ending with file extension .log. |

Example:

```
Gtw_08192_20060607154331.log
```

## Changing Maximum Sessions Per Gateway

The number of supported sessions is based on the following:

- The number of allotted Parsing Engines (PEs)
- Usage (resources consumed)
- Maximum number of sessions per gateway setting

If your site runs jobs that are CPU or I/O intensive, you might find that a lower session limit gives better performance. You cannot set the maximum sessions to a negative number.

To view the current maximum sessions, type:

```
gtwcontrol -d
```

To set the maximum sessions to 1000, type:

```
gtwcontrol -g 1 -s 1000
```



where 1 is the number of the host group, and 1000 is the desired maximum number of sessions.

The new limit is effective immediately.

If more sessions are active than the new maximum sessions limit allows, then no new sessions are started.

No new sessions can log on until the number of sessions is below the maximum sessions limit.

# Gateway Global (gtwglobal)

---

**Note:**

The utility described is not available for as-a-service customers.

---

The Gateway Global utility, gtwglobal, allows you to monitor and control the sessions of Advanced SQL Engine workstation-connected users. The gateway software runs as a separate operating system task and is the interface between the network and the Advanced SQL Engine.

Client programs that communicate through the gateway to the Advanced SQL Engine can be resident on the Advanced SQL Engine system, or they can be installed and running on network-attached workstations. Client programs that run on mainframe-attached hosts bypass the gateway completely.

Advanced SQL Engine supports multiple gateways per node. The gateways must belong to different host groups and listen on different IP addresses.

The number of supported sessions is based on the following:

- The number of allotted Parsing Engines (PEs)
- Usage (resources consumed)
- Maximum number of sessions per gateway setting in Gateway Control

Each logical network attachment requires at least one PE. Each PE can support up to 120 sessions.

When all the PEs in the DBS configuration are offline, Gateway Global exits, and the following message appears:

```
gtwglobal cannot proceed as no PEs are online.
```

When all of the PEs in any host group are offline, Gateway Global skips the group and continues to process information for the host groups with at least one PE online.

The number of sessions per gateway is defined using the gtwcontrol utility. For information on configuring gateway options, see [Gateway Control \(gtwcontrol\)](#).

---

**Note:**

Gateway errors are handled in the same manner as other Advanced SQL Engine system errors.

---

## Runs From

- Database Window or comparable interface to the Teradata console subsystem, such as cnstern
- Linux command line
- Teradata Viewpoint Remote Console portlet

For general information on starting the utilities from different interfaces, see [Starting the Utilities](#). For information on Viewpoint, see *Teradata® Viewpoint User Guide*, B035-2206.

## Syntax

```
gtwglobal [ -u ]
```

### Syntax Elements

**-u**

User names are treated as UTF-8 character encoding for input and output. Use this startup option if you have a display capable of showing UTF-8 characters.

## Usage Notes

### User Names

By default, Gateway Global treats user names as ISO 8859-1 (Latin-1) encoding. Characters that cannot be represented in 7-bit ASCII, or that are non-printable are represented using the Unicode delimited identifier format.

User names that contain no characters that would require the use of Unicode delimited identifier syntax, but that contain characters that would require quoting in SQL (such as names containing a space character), can be represented by simple double-quote character (") delimited identifiers, for example "john doe".

Unicode delimited identifiers and identifiers enclosed in double quotation marks are described in *Teradata Vantage™ - SQL Fundamentals*, B035-1141.

### Client Disconnections and Reconnections

Client sessions that are disconnected from the gateway due to communications failures can immediately reconnect.

During the reconnection, both the old lost connection and new reconnecting connection can briefly appear simultaneously, having the same session number in the output of Gateway Global commands `DISPLAY GTW`, `DISPLAY USER`, and `DISPLAY STATS USER`.

You can distinguish the old from the new connection by looking at the status displayed in the output: The new connection shows a status of `REASSIGN`, `TAKEOVER`, or `EXITED`, while the old connection shows a status of `TAKENOVER` or `LOGOFF` for the `DISPLAY GTW` and `DISPLAY USER` commands, or just the requested statistics for the `DISPLAY STATS USER` command.

### Gateway Status

Gateway Global commands [DISPLAY GTW](#) and [DISPLAY USER](#) report gateway status information, which appears as a column in the output. For disconnected sessions, status is additionally shown in the output of the [DISPLAY SESSION](#) and `DISPLAY STATS USER` commands.

The following table lists and briefly describes these statuses.

| Status    | Description                                                                                                                                                                                                                                                                    | Persistence                                                                                                                                                                                                                                     |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CONNECTED | Session is connected to the gateway.                                                                                                                                                                                                                                           | While session is connected.                                                                                                                                                                                                                     |
| DISCON    | The connection for a session was lost. Gateway is waiting for reconnection.                                                                                                                                                                                                    | Until the client reconnects or until the session is logged off after the session timeout period, which can be specified using the gtwcontrol -t command. For more information on gtwcontrol, see <a href="#">Gateway Control (gtwcontrol)</a> . |
| EXITED    | The reconnecting connection has done an implicit logoff (an orderly close of its connection) after having authenticated itself, but before the old connection has completed its takeover processing. The session will be logged off after the takeover processing is complete. | Brief, rarely seen.                                                                                                                                                                                                                             |
| FORCED    | Session has been forced to close by Gateway Global, Performance Monitoring or Production Control (PMPC), or by a security violation.                                                                                                                                           | Session timeout period, which can be specified using the gtwcontrol -t command. For more information on gtwcontrol, see <a href="#">Gateway Control (gtwcontrol)</a> .                                                                          |
| GONE      | Session forced off by PMPC.                                                                                                                                                                                                                                                    | Session timeout period, which can be specified using the gtwcontrol -t command. For more information on gtwcontrol, see <a href="#">Gateway Control (gtwcontrol)</a> .                                                                          |
| KILLED    | Session has been killed by the Gateway Global KILL SESSION or KILL USER command.                                                                                                                                                                                               | Until logoff is confirmed. Brief, rarely seen.                                                                                                                                                                                                  |
| LOGOFF    | Reported for a session while it is being logged off due to an earlier implicit logoff (an orderly close of its connection) by a new connection that was taking over the session.                                                                                               | From the time the session completes its takeover processing and begins logging off until the logoff is complete.                                                                                                                                |
| REASSIGN  | Reported for a new connection when it has requested that an existing session be reassigned to it, but it hasn't yet established the right to take over the session.                                                                                                            | Between when a reconnection attempt starts and the time it authenticates itself and is allowed to take over for a lost connection. Brief, rarely seen.                                                                                          |
| TAKENOVER | Reported for the original (disconnected) connection while the session is in the process of being taken over by a new connection during an authenticated reconnect.                                                                                                             | Until the original connection completes takeover processing or the session is logged off. Brief, rarely seen.                                                                                                                                   |
| TAKEOVER  | Reported for an authenticated new connection while it's taking over an existing session from its original (disconnected) connection                                                                                                                                            | Until the original connection completes takeover processing, the session is logged off, or the new connection does an implicit logoff or disconnects. Brief, rarely seen.                                                                       |

## Gateway Global Commands

Gateway Global presents a command-line environment that allows the entry of Gateway Global commands, which perform the following functions:

- Display network and session information
- Administer users and sessions
- Perform routine and special diagnostics
- Reload TLS configuration and certificate/key pair.

### Specifying a Host

Some commands require you to specify a host group before you can initiate the command action. To specify a host group, use the SELECT HOST command. SELECT HOST allows input of a specific host group number to define the scope used by a subsequent command function.

#### Note:

Selecting host 0 resets the host group selection.

The following sections summarize the functions of each command, and indicate whether the command requires the use of SELECT HOST.

### Displaying Network and Session Information

DISPLAY commands allow you to display your network configuration, sessions, and vprocs associated with the gateway, plus information about specific sessions.

| Command                            | Function                                                                                                                                                        | Requires SELECT HOST |
|------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|
| <a href="#">DISPLAY DISCONNECT</a> | Displays a list of sessions that have disconnected.                                                                                                             | Yes                  |
| <a href="#">DISPLAY FORCE</a>      | Displays sessions that have been successfully killed or aborted using Performance Monitoring and Production Control (PMPC) abort process.                       | Yes                  |
| <a href="#">DISPLAY GTW</a>        | Displays all sessions connected to the gateway.                                                                                                                 | No                   |
| <a href="#">DISPLAY NETWORK</a>    | Displays network configuration information.                                                                                                                     | No                   |
| <a href="#">DISPLAY SESSION</a>    | Displays the following information on a specific session: User name, IP address, TCP socket number, state, event, action, partition, and authentication method. | Yes                  |
| <a href="#">DISPLAY STATS</a>      | Displays the RSS statistics for the gateway vproc.                                                                                                              | No                   |
| <a href="#">DISPLAY TIMEOUT</a>    | Displays the timeout value.                                                                                                                                     | Yes                  |

| Command                      | Function                                                                                      | Requires<br>SELECT<br>HOST |
|------------------------------|-----------------------------------------------------------------------------------------------|----------------------------|
| <a href="#">DISPLAY USER</a> | Displays the session number, PE number, User name, IP address, and current connection status. | Yes                        |

## Administering Users and Sessions

These commands allow you to control gateway traffic and access to the database.

| Command                        | Function                                                 | Requires<br>SELECT<br>HOST |
|--------------------------------|----------------------------------------------------------|----------------------------|
| <a href="#">DISABLE LOGONS</a> | Disables all logons to the database through the gateway. | Yes                        |
| <a href="#">ENABLE LOGONS</a>  | Enables logons to the database through the gateway.      | Yes                        |
| <a href="#">KILL SESSION</a>   | Terminates a specific session.                           | Yes                        |
| <a href="#">KILL USER</a>      | Terminates all sessions of a specific user.              | Yes                        |
| <a href="#">SET TIMEOUT</a>    | Sets a timeout value.                                    | Yes                        |

## Reloading TLS Configuration and Certificate/Key Pair

These commands allow you to update the TLS configuration file and load a new certificate/key pair.

| Command                   | Function                                                                                                                                                                                                                                                            | Requires<br>SELECT<br>HOST |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|
| REREAD<br>TLS CONFIG      | Updates the TLS configuration, such as cipher suites, from the gateway TLS file.<br>If you make changes in the TLS configuration file <code>/usr/tgtw/etc/gtwtls.cfg</code> , you can use REREAD TLS Config to reload that configuration file with the new changes. | Yes                        |
| REREAD TLS<br>CERTIFICATE | Updates the TLS certificate and private key from the new gateway TLS certificate and key files.<br>If you installed a new certificate/pair, you can use REREAD TLS Certificate to load them.                                                                        | Yes                        |

## Performing Special Diagnostics

The TRACE commands allow you to debug internal gateway errors or anomalies.

| Command                      | Function                         | Requires<br>SELECT<br>HOST |
|------------------------------|----------------------------------|----------------------------|
| <a href="#">ENABLE TRACE</a> | Records internal gateway events. | Yes                        |

| Command                       | Function                                                                                          | Requires<br>SELECT HOST |
|-------------------------------|---------------------------------------------------------------------------------------------------|-------------------------|
| <a href="#">DISABLE TRACE</a> | Turns off the recording of event tracing and writing to the gateway log files.                    | Yes                     |
| <a href="#">FLUSH TRACE</a>   | Directs the gateway to write the contents of its internal trace buffers to the gateway log files. | Yes                     |

## Logging Sessions Off Using KILL

When you issue a KILL command, any outstanding requests are first aborted. The specified session is logged off. The following table describes the KILL command behavior.

| IF the session is currently.<br>.. | THEN...                                                                                                           |
|------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| disconnected                       | the assign task attempts a log off by setting the KILL flag to indicate that an attempt has been made to log off. |
| logged on                          | a kill message is sent to the connect task to abort any outstanding requests and then to log off.                 |

### Note:

Aborting an outstanding request could take a significant amount of time. Therefore, killing a specific session or all of a user's sessions does not necessarily free up the resources of those sessions immediately. For more information on KILL commands, see [KILL SESSION](#) and [KILL USER](#).

## Getting Help

To list all the Gateway Global commands, type `help` at the command line. The help command displays a menu of all Gateway Global commands alphabetically, and shows the syntax for each command. The letter `h` can be used as a synonym for `help`.

## DISABLE LOGONS

The DISABLE LOGONS command prevents users from logging on to the database on the network using the gateway for the selected host group.

### Syntax

```
{ DISABLE | DISA } { LOGONS | LOGO }
```

### Usage Notes

The default setting of the logon flag is enabled. However, after a Teradata system restart, all host groups that were disabled remain disabled.

Any application that attempts to log on after DISABLE LOGONS results in the following error message:

8033: Logon is disabled.

---

#### Note:

Sessions already logged on are not affected by the DISABLE LOGONS command.

---

Before using the DISABLE LOGONS command, you must select the host group from which the session is running using the SELECT HOST command. For information, see [SELECT HOST](#).

For additional information, see [ENABLE LOGONS](#).

### Example: Disabling logons

To prevent network users from logging on through the gateway to the database, type:

```
disa logo
```



## DISABLE TRACE

The DISABLE TRACE command turns off the recording of event tracing and writing to the gateway log files.

### Syntax

```
{ DISABLE | DISA } { TRACE | TRAC }
```

### Usage Notes

The default setting of the trace flag is disabled.

---

#### Note:

If tracing has been enabled and you do not use the DISABLE TRACE command to turn it off, the file system may become full.

---

Before using the DISABLE TRACE command, you must select the host group from which the session is running using the SELECT HOST command. For information, see [SELECT HOST](#).

For additional information, see [ENABLE TRACE](#).

For information on the gateway log files, see [Gateway Control \(gtwcontrol\)](#).

### Example: Disabling event tracing and writing to gateway logs

To turn off the recording of event tracing and writing to the gateway log files, type:

```
disa trac
```

No report is generated.

## DISPLAY DISCONNECT

The DISPLAY DISCONNECT command returns a list of sessions that have disconnected and not yet reconnected.

### Syntax

```
{ DISPLAY | DI } { DISCONNECT | DISC }
```

### Usage Notes

The context for these sessions is maintained in the gateway control assign task. If the client associated with the session fails to reconnect in the time allotted, the session is logged off.

Before using the DISPLAY DISCONNECT command, you must select the host group from which the session is running using the SELECT HOST command. For information, see [SELECT HOST](#).

### Example: Displaying disconnected sessions

To display disconnected sessions, type:

```
di disc
```

The following report appears.

Host Group 52 has 5 disconnected sessions:

| Session | PE    | User | TimeToLive |
|---------|-------|------|------------|
| 1050    | 16383 | DBC  | 1185 secs  |
| 1051    | 16383 | DBC  | 1185 secs  |
| 1052    | 16383 | DBC  | 1185 secs  |
| 1053    | 16383 | DBC  | 1185 secs  |
| 1054    | 16383 | DBC  | 1185 secs  |

## DISPLAY FORCE

The DISPLAY FORCE command displays sessions that have been forced off by Gateway Global, Performance Monitoring and Production Control (PMPC), or a security violation.

### Syntax

```
{ DISPLAY | DI } { FORCE | FO }
```

### Usage Notes

The Teradata system displays this information:

- Host group number
- Number of sessions
- Session ID number
- User name associated with the session
- The length of time before the session information is discarded

The gateway retains the session information for the standard timeout period after the client has been logged off. The gateway returns an 8055 error when the client reconnects:

Session forced off by PMPC or gtwglobal

After the timeout period expires, the gateway discards the session information.

Before using the DISPLAY FORCE command, you must select the host group from which the session is running using the SELECT HOST command. For information, see [SELECT HOST](#).

### Example: Displaying sessions that have been killed or aborted

To display a session that has been killed or aborted, type:

```
DI FO
```

The following appears.

```
Host Group 1 has 2 forced sessions
```

```
Session User TimeToLive
1002    DBC  916 secs
1004    DBC  922 secs
```

## DISPLAY GTW

The DISPLAY GTW command displays all sessions connected to the gateway.

### Syntax

```
{ DISPLAY | DI } GTW { gtwid | ALL }
```

### Syntax Elements

#### *gtwid*

vproc number of the gateway containing the sessions you want to display.

#### ALL

Keyword that displays all sessions regardless of selected Host Group.

---

#### Note:

To list all possible values for *gtwid*, use [DISPLAY NETWORK](#).

---

### Usage Notes

For a specific *gtwid*, Gateway Global shows:

- Gateway vproc number
- Node the gateway vproc is running on, in the format *cabinetID-moduleID*
- Number of sessions on the gateway
- The following information for each session:
  - Session ID number
  - PE vproc number
  - Node the PE vproc is running on, in the format *cabinetID-moduleID*
  - Local? can be Yes or No. Yes indicates that the PE and gateway vprocs are running on the same node.
  - User name associated with the session
  - Internet address of the originating host for the session
  - Status
  - SP (Client Security Policy)
  - PP (Proxy Security Policy)
  - RNP (Recoverable Network Protocol)

If the ALL keyword is specified, Gateway Global shows:

- Name of the database
- Total number of sessions for all gateways

- The following information for each session:
  - Host group number
  - Gateway vproc number
  - Node the gateway vproc is running on, in the format *cabinetID-moduleID*
  - Session ID number
  - PE vproc number
  - Node the PE vproc is running on, in the format of *cabinetID-moduleID*
  - Local? can be Yes or No. Yes indicates the PE and gateway vprocs are running on the same node.
  - User name associated with the session
  - Internet address of the originating host for the session
  - Status
  - SP (Client Security Policy)
  - PP (Proxy Security Policy)
  - RNP (Recoverable Network Protocol)

### Example: Displaying sessions connected to a specified gateway

To display a gateway (for example, gateway 22528), type:

```
di gtw 22528
```

The following appears:

GTW 22528 on node 001-01 has 18 sessions

| Session | PE    | Node   | Local? | User | IP Adr     | Status    | SP | PP | RNP |
|---------|-------|--------|--------|------|------------|-----------|----|----|-----|
| 1001    | 30719 | 001-01 | Yes    | TEST | 192.0.2.30 | CONNECTED | N  |    | No  |
| 1003    | 30719 | 001-01 | Yes    | TEST | 192.0.2.30 | CONNECTED | N  |    | No  |
| 1005    | 30719 | 001-01 | Yes    | TEST | 192.0.2.30 | CONNECTED | N  |    | No  |
| 1006    | 30719 | 001-01 | Yes    | TEST | 192.0.2.30 | CONNECTED | N  |    | No  |
| 1007    | 30719 | 001-01 | Yes    | TEST | 192.0.2.30 | CONNECTED | N  |    | No  |
| 1008    | 30719 | 001-01 | Yes    | TEST | 192.0.2.30 | CONNECTED | N  |    | No  |
| 1009    | 30719 | 001-01 | Yes    | TEST | 192.0.2.30 | CONNECTED | N  |    | No  |
| 1010    | 30719 | 001-01 | Yes    | TEST | 192.0.2.30 | CONNECTED | N  |    | No  |
| 1011    | 30719 | 001-01 | Yes    | TEST | 192.0.2.30 | CONNECTED | N  |    | No  |
| 1012    | 30719 | 001-01 | Yes    | TEST | 192.0.2.30 | CONNECTED | N  |    | No  |
| 1013    | 30719 | 001-01 | Yes    | TEST | 192.0.2.30 | CONNECTED | N  |    | No  |
| 1014    | 30719 | 001-01 | Yes    | TEST | 192.0.2.30 | CONNECTED | N  |    | No  |
| 1015    | 30719 | 001-01 | Yes    | TEST | 192.0.2.30 | CONNECTED | N  |    | No  |
| 1016    | 30719 | 001-01 | Yes    | TEST | 192.0.2.30 | CONNECTED | N  |    | No  |
| 1017    | 30719 | 001-01 | Yes    | TEST | 192.0.2.30 | CONNECTED | N  |    | No  |
| 1018    | 30719 | 001-01 | Yes    | TEST | 192.0.2.30 | CONNECTED | N  |    | No  |

|      |       |        |     |      |            |           |   |    |
|------|-------|--------|-----|------|------------|-----------|---|----|
| 1019 | 30719 | 001-01 | Yes | TEST | 192.0.2.30 | CONNECTED | N | No |
| 1020 | 30719 | 001-01 | Yes | TEST | 192.0.2.30 | CONNECTED | N | No |

**Example: Displaying sessions on all host groups**

To display all of the gateway sessions on all host groups, type:

```
di gtw all
```

The following appears:

System WUSRH122051-JEA has 1 connected session

| HG  | GTW   | Node   | Session | PE    | Node   | Local? | User  | IP         | Adr   | Status    | SP | PP | RNP |
|-----|-------|--------|---------|-------|--------|--------|-------|------------|-------|-----------|----|----|-----|
| --- | ----- | -----  | -----   | ----- | -----  | -----  | ----- | -----      | ----- | -----     | -- | -- | --- |
| 1   | 22528 | 001-01 | 1005    | 16383 | 001-01 | Yes    | TEST  | 192.0.2.30 |       | CONNECTED | N  |    | No  |

## DISPLAY NETWORK

The DISPLAY NETWORK command displays information about your network configuration and its associated gateway.

### Syntax

```
{ DISPLAY | DI } { NETWORK | NE } [ { LONG | LON } | host_no ]
```

### Syntax Elements

#### *host\_no*

Host group number (in decimal).

#### LONG

Display of the gateway statistics for the particular network.

### Usage Notes

The default setting for the DISPLAY NETWORK command is the short form report, which gives this information for the specified host group, or for each host group in Vantage:

- Host group numbers
- Total number of sessions on the host group
- Total number of gateways assigned to host group
- Total number of PEs assigned to host group
- Total number of active sessions on the host group
- Total number of disconnected sessions on the host group
- Total number of forced off (killed) sessions on the host group
- The following information about each gateway in the host group:
  - Vproc number of the gateway
  - Node the gateway vproc is running on, in the format *cabinetID-moduleID*
  - Total number of sessions connected to the gateway
  - Logon information (either Enabled or Disabled)
  - Tracing information (either Enabled or Disabled)

The long form displays the following information about each PE in a host group in addition to the information displayed in the short form report for the network:

- Vproc number of the PE
- Node the PE vproc is running on, in the format *cabinetID-moduleID*
- Total number of sessions connected to the PE

If *host\_no* is used to request a specific host group number, the long form cannot be requested.

**Example: Displaying network and gateway configuration information**

To display information about your network configuration and its associated gateway in the short form, type:

```
di ne
Host 2 has 0 session(s) over 1 GTW(s) and 1 PE(s)
( 0 Active / 0 Disconnected / 0 Forced)

Gateway Node    Sessions Logon    Trace
-----
8193    001-02 0          Enable  Disable

Host 1 has 20 session(s) over 1 GTW(s) and 2 PE(s)
( 18 Active / 0 Disconnected / 2 Forced)

Gateway Node    Sessions Logon    Trace
-----
8192    001-01 18          Enable  Disable
```

**Example: Displaying network and gateway configuration information (long format)**

To display information about your network configuration and its associated gateway in the long form, type:

```
di ne long

Host 1 has 0 session(s) over 1 GTW(s) and 2 PE(s)
( 0 Active / 0 Disconnected / 0 Forced)

PE      Node    Sessions
-----
30719   001-01 0
30718   001-01 0

Gateway Node    Sessions Logon    Trace
-----
22528   001-01 0          Enable  Disable
```



## DISPLAY SESSION

The DISPLAY SESSION command displays information for a specified session.

### Syntax

```
{ DISPLAY | DI } { SESSION | SE }
  [ LONG | LON ] { ses_lst | { USER | US } user_name }
```

### Syntax Elements

#### LONG LON

Displays additional detailed information for the specified sessions.

For backward compatibility, LONG is also valid after *ses\_lst*, but it is only valid before USER.

#### *ses\_lst*

One or more session numbers indicating the sessions for which information is to be displayed.

Multiple sessions can be specified as:

- a range of contiguous session numbers, represented by the lowest and highest session in the range separated by a hyphen: *nnnn-nnnn*

Multiple ranges can be separated by commas: *nnnn-nnnn,nnnn-nnnn*

- A list of discontinuous session numbers separated by commas: *nnnn,nnnn,nnnn*
- Any combination of ranges and lists of session numbers: *nnnn-nnnn,nnnn,nnnn*

where *nnnn* is a session number in decimal format. To list currently connected sessions and see possible values for *nnnn*, use the [DISPLAY GTW](#) command.

#### *user\_name*

The name of the user account for which you want to display session information.

### Usage Notes

- Before using the DISPLAY SESSION command, you must select the host group from which the session is running using the SELECT HOST command. For information, see [SELECT HOST](#).
- By default, the Teradata system displays the short-form report containing the following information:
  - User Name
  - IP address
  - TCP socket number
  - State

- Event
  - Action
  - Partition
  - Authentication method
  - Client Security Policy
  - Proxy Security Policy
  - Recoverable Network Protocol
- The Authentication field has the values shown in the following table:

| The value...          | Indicates...                                                                                                                                                                                                  |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Database              | that the database provided the authentication method.<br><br><b>Note:</b><br>This was the normal logon method before the implementation of single sign on.                                                    |
| TDGSS-returned method | an authentication method returned by the Teradata Generic Security Service (TDGSS).<br>For more information on TDGSS, see <i>Teradata Vantage™ - Advanced SQL Engine Security Administration</i> , B035-1100. |

- The long-form report gives a detailed connect display for a selected session. It includes all of the information displayed by the default short format, and adds the following information:
  - The Account associated with the User Name, if any.
  - Teradata system mailbox information.
  - I/O statistics for the session.
  - The authentication negotiating mechanism used, if any.

The long-form report is useful for field service personnel when diagnosing gateway problems.

- For information on the descriptions of the valid states, events, and actions for the DISPLAY SESSION command, see [Session States, Events, and Actions](#).
- To display a session, do the following:
  1. To select a host group (for example, host group 52), type:
 

```
se host 52
```

The Teradata system displays the following:

```
52>
```
  2. To display a session (for example, session 1040), type:
 

```
di se 1040
```

**Example: Displaying session information**

To display session 1040 in the default (or short) form, type:

```
> di se 1040
```

gtwglobal displays the following output:

```
Session 1040 connected to GTW 8193 is assigned to PE 16383
of host 1

User Name          IP Addr          Port
-----
DBC                192.0.2.30      27399

State              Event              Action
-----
CS_CLIENTWAITNOTRAN  CE_STARTMSGRSPNOTRAN  CA_SENDDBSRSP

Partition          Authentication
-----
DBC/SQL            DATABASE

Client Security Policy  Proxy Security Policy
-----
No Policy

Recoverable Network Protocol
-----
Not Supported
```

**Example: Displaying session information (long format)**

To display session 1000 in the long form, type:

```
> di se 1000 long
```

gtwglobal displays the following output:

```
di sess 1000 long

Session 1000 connected to GTW 22528 is assigned to PE 30719
of host 1

User Name          Account          IP Addr          Port
-----
DBC                127.0.0.1      55837
```

| State                                  | Event                 | Action               |
|----------------------------------------|-----------------------|----------------------|
| -----                                  | -----                 | -----                |
| CS_CLIENTWAITNOTRAN                    | CE_CONTINUEMSGRSPOK   | CA_SENDDBSRSP        |
| Partition                              | Authentication        | NegotiatingMechanism |
| -----                                  | -----                 | -----                |
| DBC/SQL                                | DATABASE              | TDNEGO               |
| Client Security Policy                 | Proxy Security Policy |                      |
| -----                                  | -----                 |                      |
| No Policy                              |                       |                      |
| Recoverable Network Protocol           |                       |                      |
| -----                                  |                       |                      |
| Not Supported                          |                       |                      |
| StrMbx: 01 00 77ff 00000002 0000 00 00 |                       |                      |
| CntMbx: 02 00 77ff 00000000 040d 00 00 |                       |                      |
| AbtMbx: 02 00 77ff 00000000 030d 00 00 |                       |                      |
| HostMessageReads :                     | 18                    |                      |
| HostBlockReads :                       | 9                     |                      |
| HostReadBytes :                        | 2384                  |                      |
| DbsMessageReads :                      | 5                     |                      |
| HostMessageWrites :                    | 9                     |                      |
| HostBlockWrites :                      | 9                     |                      |
| HostWriteBytes :                       | 20663                 |                      |
| DbsMessageWrites :                     | 5                     |                      |

Example: Specifying a list of sessions that includes a range

|                                                                       |                       |               |
|-----------------------------------------------------------------------|-----------------------|---------------|
| > di se 1119-1120,1117                                                |                       |               |
| Session 1119 connected to GTW 22528 is assigned to PE 30718 of host 1 |                       |               |
| User Name                                                             | IP Addr               | Port          |
| -----                                                                 | -----                 | -----         |
| DBC                                                                   | 198.51.100.0          | 43727         |
| State                                                                 | Event                 | Action        |
| -----                                                                 | -----                 | -----         |
| CS_CLIENTWAITNOTRAN                                                   | CE_STARTMSGRSPNOTRAN  | CA_SENDDBSRSP |
| Partition                                                             | Authentication        |               |
| -----                                                                 | -----                 |               |
| DBC/SQL                                                               | DATABASE              |               |
| Client Security Policy                                                | Proxy Security Policy |               |
| -----                                                                 | -----                 |               |
| No Policy                                                             |                       |               |

Recoverable Network Protocol

-----  
Not Supported

Session 1120 connected to GTW 22528 is assigned to PE 30719  
of host 1

| User Name | IP Addr       | Port  |
|-----------|---------------|-------|
| ETL03     | 198.51.100.23 | 43728 |

| State               | Event                 | Action        |
|---------------------|-----------------------|---------------|
| CS_CLIENTWAITNOTRAN | CE_STARTMSGGRSPNOTRAN | CA_SENDDBSRSP |

| Partition | Authentication |
|-----------|----------------|
| DBC/SQL   | DATABASE       |

| Client Security Policy | Proxy Security Policy |
|------------------------|-----------------------|
| No Policy              |                       |

Recoverable Network Protocol

-----  
Not Supported

Session 1117 connected to GTW 22528 is assigned to PE 30718  
of host 1

| User Name | IP Addr       | Port  |
|-----------|---------------|-------|
| ETL02     | 198.51.100.21 | 43725 |

| State               | Event                 | Action        |
|---------------------|-----------------------|---------------|
| CS_CLIENTWAITNOTRAN | CE_STARTMSGGRSPNOTRAN | CA_SENDDBSRSP |

| Partition | Authentication |
|-----------|----------------|
| DBC/SQL   | DATABASE       |

| Client Security Policy | Proxy Security Policy |
|------------------------|-----------------------|
| No Policy              |                       |

Recoverable Network Protocol

-----  
Not Supported

### Example: Displaying all current sessions of a specified user

```
> di se us dbc
```

User DBC has 2 sessions

Session 1116 connected to GTW 22528 is assigned to PE 30719  
of host 1

| User Name | IP Addr        | Port  |
|-----------|----------------|-------|
| DBC       | 198.51.100.255 | 43724 |

```

State                      Event                      Action
-----
CS_CLIENTWAITNOTRAN       CE_STARTMSGGRSPNOTRAN   CA_SENDDBSRSP

Partition      Authentication
-----
DBC/SQL        DATABASE

Client Security Policy      Proxy Security Policy
-----
No Policy

Recoverable Network Protocol
-----
Not Supported

Session 1117 connected to GTW 22528 is assigned to PE 30718
of host 1

User Name              IP Addr      Port
-----
DBC                    198.51.100.255  43725

State                      Event                      Action
-----
CS_CLIENTWAITNOTRAN       CE_STARTMSGGRSPNOTRAN   CA_SENDDBSRSP

Partition      Authentication
-----
DBC/SQL        DATABASE

Client Security Policy      Proxy Security Policy
-----
No Policy

Recoverable Network Protocol
-----
Not Supported

```

**Example: Displaying the TLS version and Cipher Suite**

```
> di se 1000
```

Session 1000 connected to GTW 22528 is assigned to PE 30719 of host 1

| User Name                    | IP Addr               | Port                           |
|------------------------------|-----------------------|--------------------------------|
| DBC                          | 10.25.64.17           | 55376                          |
| State                        | Event                 | Action                         |
| CS_CLIENTWAITNOTRAN          | CE_STARTMSGRSPNOTRAN  | CA_SENDDBSRSP                  |
| Partition                    | Authentication TLS    | Cipher Suite                   |
| DBC/SQL                      | DATABASE              | TLSv1.2 TLS_AES_256_GCM_SHA384 |
| Client Security Policy       | Proxy Security Policy |                                |
| No Policy                    |                       |                                |
| Recoverable Network Protocol |                       |                                |

If the current session is a legacy session, TLS version and Cipher will not display. The columns will appear as:

| Partition | Authentication TLS | Cipher Suite |
|-----------|--------------------|--------------|
| DBC/SQL   | DATABASE           | Not Used     |

## DISPLAY STATS

The DISPLAY STATS command allows you to display the statistics for the gateway vproc.

### Syntax

```
{ DISPLAY | DI } { STATS | ST }
{
  { REPEAT | REP } secs reps { USER | US } user_name |
  gtwid |
  ALL
}
```

### Syntax Elements

#### REPEAT

Automatically reruns the DISPLAY STATS command at a specified interval a specified number of times.

##### Note:

The DISPLAY STATS command can be repeated automatically only when used to display USER statistics.

#### secs

Time, in seconds, between runs of DISPLAY STATS.

#### reps

Number of times to repeat the DISPLAY STATS command.

#### user\_name

The name of the user account for which you want to display connected session statistics.

#### gtwid

Specified gateway vproc whose statistics are displayed.

#### ALL

Selected host group.

If you select a host group, gtwglobal returns the statistics for all the gateway vprocs in the selected host group.

If you do not select a host group, gtwglobal returns the statistics for all gateway vprocs.



**Example: Displaying gateway vproc statistics**

```
di st all
```

```
Gtw Vprocid:      8193
HostMessageReads:  382
HostBlockReads:    125
HostReadBytes:     3829382
DbsMessageReads:   108
HostMessageWrites: 125
HostBlockWrites:   125
HostWriteBytes:    1167
DbsMessageWrites:  104
Gtw Vprocid:      8192
HostMessageReads:  2220
HostBlockReads:    1107
HostReadBytes:     92919
DbsMessageReads:   1103
HostMessageWrites: 1107
HostBlockWrites:   1107
HostWriteBytes:    3950263
DbsMessageWrites:  1103
```

Example: DISPLAY STATS command with automatic repeats

```
> di stats repeat 30 2 user ETL01
```

Repeat 1 of 2 with 30 second intervals started at 19-Oct-2016 11:22:07

User ETL01 has 3 sessions

| Clock | HG | Session | GTW | Node | PE | Node | Partition | IP Addr | HostReadBytes | Since Last | HostWriteBytes |
|-------|----|---------|-----|------|----|------|-----------|---------|---------------|------------|----------------|
|-------|----|---------|-----|------|----|------|-----------|---------|---------------|------------|----------------|

|          |   |         |       |        |       |        |          |               |          |          |  |
|----------|---|---------|-------|--------|-------|--------|----------|---------------|----------|----------|--|
| 11:22:07 | 1 | 55801   | 22528 | 001-01 | 30719 | 001-02 | TPTLOAD  | 203.0.113.255 | 36363    | 36363    |  |
| 17604    |   | 17604   |       |        |       |        |          |               |          |          |  |
| 11:22:07 | 1 | 55802   | 22528 | 001-01 | 30718 | 001-01 | DBC/SQL  | 198.51.100.0  | 14768023 | 14768023 |  |
| 9067819  |   | 9067819 |       |        |       |        |          |               |          |          |  |
| 11:22:07 | 1 | 55804   | 22528 | 001-01 | 30718 | 001-01 | MULTLOAD | 198.51.100.0  | 1183     | 1183     |  |
| 798      |   | 798     |       |        |       |        |          |               |          |          |  |

Repeat 2 of 2 with 30 second intervals started at 19-Oct-2016 11:22:37

User ETL01 has 3 sessions

| Clock | HG | Session | GTW | Node | PE | Node | Partition | IP Addr | HostReadBytes | Since Last | HostWriteBytes |
|-------|----|---------|-----|------|----|------|-----------|---------|---------------|------------|----------------|
|-------|----|---------|-----|------|----|------|-----------|---------|---------------|------------|----------------|

|          |   |       |       |        |       |        |          |               |          |     |  |
|----------|---|-------|-------|--------|-------|--------|----------|---------------|----------|-----|--|
| 11:22:37 | 1 | 55801 | 22528 | 001-01 | 30719 | 001-02 | TPTLOAD  | 203.0.113.255 | 36363    | 0   |  |
| 17604    |   | 0     |       |        |       |        |          |               |          |     |  |
| 11:22:37 | 1 | 55802 | 22528 | 001-01 | 30718 | 001-01 | DBC/SQL  | 198.51.100.0  | 14768193 | 170 |  |
| 9067879  |   | 60    |       |        |       |        |          |               |          |     |  |
| 11:22:37 | 1 | 55804 | 22528 | 001-01 | 30718 | 001-01 | MULTLOAD | 198.51.100.0  | 1183     | 0   |  |
| 798      |   | 0     |       |        |       |        |          |               |          |     |  |

Repeat 2 of 2 with 30 second intervals ended at 19-Oct-2016 11:22:37



## DISPLAY TIMEOUT

The DISPLAY TIMEOUT command allows you to display the logoff delay in minutes for disconnected sessions.

### Syntax

```
{ DISPLAY | DI } { TIMEOUT | TI }
```

### Example: Displaying the logoff delay for disconnected sessions

Before using the DISPLAY TIMEOUT command, you must select the host group from which the session is running using the SELECT HOST command. For more information, see [SELECT HOST](#).

To display the timeout logoff delay, type:

```
di ti
```

The following appears:

```
Host 1 Timeout Value: 20 minutes
```

## DISPLAY USER

The DISPLAY USER command returns a list of connected sessions whose names match the user name.

### Syntax

```
{ DISPLAY | DI } { USER | US } user_name
```

### Syntax Elements

*user\_name*

Name of the user.

### Usage Notes

Before using the DISPLAY USER command, you must select the host group from which the sessions are running using the SELECT HOST command. For more information, see [SELECT HOST](#).

The DISPLAY USER command displays this information:

- Vproc number of the gateway the session is connected to
- Node the gateway vproc is running on, in the format *cabinetID-moduleID*
- Session number
- Parsing Engine that the session is assigned to
- Node the PE vproc is running on, in the format of *cabinetID-moduleID*
- Local? can be Yes or No. Yes indicates the PE and gateway vprocs are running on the same node.
- User name
- IP address
- Status indicating the following:
  - Connected
  - Forced
  - Gone
  - Killed
- SP (Client Security Policy)
- PP (Proxy Security Policy)

### Example: Displaying sessions of a specified user

To display a information about a user, type:

```
di us test
```

The following appears:

```
User DBC has 1 session
```

| GTW   | Node   | Session | PE    | Node   | Local? | User  | IP            | Adr   | Status    | SP  | PP  | RNP |
|-------|--------|---------|-------|--------|--------|-------|---------------|-------|-----------|-----|-----|-----|
| ----- | -----  | -----   | ----- | -----  | -----  | ----- | -----         | ----- | -----     | --- | --- | --- |
| 22528 | 001-01 | 1238    | 30719 | 001-01 | Yes    | TEST  | 153.64.183.89 |       | CONNECTED | N   |     | No  |

## ENABLE LOGONS

The ENABLE LOGONS command allows users to log on to the database using the network through the gateway for the selected host group.

### Syntax

```
{ ENABLE | EN } { LOGONS | LOGO }
```

### Usage Notes

The default setting of the logons flag is enabled.

Before using the ENABLE LOGONS command, you must select the host group from which the session is running using the SELECT HOST command.

For additional information, see [SELECT HOST](#) and [DISABLE LOGONS](#).

### Example: Enabling network gateway logons

To allow users to log on to the database using the network through the gateway, type:

```
enable logons
```

## ENABLE TRACE

The ENABLE TRACE command turns on session tracing. The command records internal gateway events.

### Syntax

```
{ ENABLE | EN } { TRACE | TRAC }
```

### Usage Notes

Session tracing may contain some of these elements:

---

#### Note:

Entries vary depending on the events and action taken.

---

- Session number (if available)
- Associated file description (if available)
- Event
- One or more of these events:
  - Assigned
  - Logged on
  - Logged off
  - Logoff forced (Teradata system killed session)
  - Logon failed (usually caused by invalid password)
  - Disconnected
  - Reassigned
  - Reconnected
- Network address of the originating network-attached host

By default, tracing is disabled. After a Teradata system restart, tracing enabled earlier using the ENABLE TRACE command is continued.

---

#### Note:

If tracing has been enabled and you do not use the DISABLE TRACE command to turn it off, the file system may become full.

---

Before using the ENABLE TRACE command, you must select the host group from which the session is running using the SELECT HOST command. For information, see [SELECT HOST](#).

The trace file is a text file, which can be viewed by any tool that displays text. The trace information is the same trace information controlled by Gateway Control, and is logged to the same files.

For additional information, see [DISABLE TRACE](#). For information on the gateway log files, see [Gateway Control \(gtwcontrol\)](#).



**Example: Enabling session event tracing**

To turn on session event tracing, type:

```
enable trace
```

## FLUSH TRACE

The gateway records all significant events to an internal circular buffer, such that as new events are recorded, older events are overwritten in the buffer. The FLUSH TRACE command directs the gateway to write every event currently in the buffer to the gateway log file. This preserves a record of the most recent events, which would otherwise be lost from the buffer.

FLUSH TRACE is useful for investigating gateway problems by logging the most recent gateway activities if tracing has not already been enabled with the ENABLE TRACE command, or with the Gateway Control tracing options. For more information on these options, see [Gateway Control \(gtwcontrol\)](#).

### Syntax

```
{ FLUSH | FL } { TRACE | TRAC }
```

### Usage Notes

The FLUSH TRACE command can be used when diagnosing gateway anomalies.

For additional information, see [ENABLE TRACE](#) and [DISABLE TRACE](#).

Before using the FLUSH TRACE command, you must select the host group from which the session is running using the SELECT HOST command. For information, see [SELECT HOST](#).

For information on the gateway log files, see [Gateway Control \(gtwcontrol\)](#).

### Example: Writing buffered gateway events to the gateway log file

To direct the gateway to write the contents of its internal trace buffers to the gateway log file, type:

```
flush trace
```

## HELP

The HELP command allows you to get help when using gtwglobal commands.

### Syntax

```
{ HELP | H }
```

## KILL SESSION

The KILL SESSION command terminates the specified session on the database by aborting any active request on that session and logging the session off.

---

**Note:**

Aborting an outstanding request could take a significant amount of time. Therefore, killing a session or a user's sessions does not necessarily free up the resources of those sessions immediately.

---

### Syntax

```
{ KILL | KI } { SESSION | SE } ses_no
```

### Syntax Elements

**ses\_no**

Session number (in decimal).

---

**Note:**

To list the possible values for *ses\_no*, use [DISPLAY GTW](#) or [DISPLAY NETWORK](#).

---

### Usage Notes

Before using the KILL SESSION command, you must select the host group from which the session is running using the SELECT HOST command. For information, see [SELECT HOST](#).

The KILL SESSION command specifies that the session identified by *ses\_no* be terminated. This command leaves an audit trail in the error log.

### Example: Terminating a specified session

To terminate session 1000 on the network through the gateway, type:

```
kill se 1000
```

The following appears:

Session 1000 scheduled to be killed.

## KILL USER

The KILL USER command terminates all logged on sessions for the specified user name restricted to the host group of the gateway. KILL USER aborts any active requests on those sessions and logs the sessions off.

### Syntax

```
{ KILL | KI } { USER | US } user_name
```

### Syntax Elements

*user\_name*

Name of the user.

---

#### Note:

To list the possible values for *user\_name*, use [DISPLAY GTW](#) or [DISPLAY SESSION](#).

---

### Usage Notes

Aborting an outstanding request can take a significant amount of time. Killing a session or a user's sessions does not necessarily free up the resources of those sessions immediately.

Before using the KILL USER command, select the host group from which the session is running using the SELECT HOST command. For more information, see [SELECT HOST](#).

The KILL USER command leaves an audit trail in the gateway log files, which shows the vproc that the kill was issued on and the session number of the user being killed. Because the attempt might be denied by Vantage, only the attempt is logged:

```
Gtwglobal issued kill command:
gtwassign.c @3205 (867): Fri Apr 18 13:39:01 2008
VprocId: 8193 Session Number: 1009
```

#### Example: Terminating all sessions for a specified user

```
kill us perm01
```

```
User PERM01 has 1 session killed
1005
```

# SELECT HOST

The SELECT HOST command selects the host number of a group of gateways, and the PEs that have been assigned to those gateways. Subsequent Gateway Global commands will affect only those selected gateways until a new SELECT HOST command is issued. In this context, the specified collection of gateways and PE comprises a host group.

## Syntax

```
{ SELECT | SE } { HOST | HO } host_no
```

## Syntax Elements

*host\_no*  
Host group number.

### Example: Specifying a host for subsequent Gateway Global commands

To select host group 1, type:

```
select host 1
```

| IF the Host Group number is... | THEN the following message appears...        |
|--------------------------------|----------------------------------------------|
| found                          | Host 1 has been selected.                    |
| not found                      | Invalid Host 1 is entered. Please try again! |

To deselect a host group, type:

```
select host 0
```

## SET TIMEOUT

The SET TIMEOUT command allows you to set the time in minutes to delay a logoff for disconnected sessions.

### Syntax

```
SET { TIMEOUT | TI } time_value
```

### Syntax Elements

*time\_value*

Amount of time to delay logoff in minutes.

### Usage Notes

The SET TIMEOUT command can be used to delay logoff for disconnected sessions when working on network or database problems.

Before using the SET TIMEOUT command, you must select the host group from which the sessions are running using the SELECT HOST command. For information, see [SELECT HOST](#).

### Example: Setting the time between a session disconnection and logoff

To set a timeout delay for one hour, type:

```
SET TI 60
```

```
Timeout value on host 1 set to 60
```

## Lock Display (lokdisp)

---

**Note:**

The utility described is not available for as-a-service customers.

---

The Lock Display utility, lokdisp, provides information about all real-time database locks.

Advanced SQL Engine uses locks for concurrency control, ensuring data involved in a particular transaction remains consistent for the duration of that transaction, while the database remains available to multiple concurrent users and queries. For more information on Advanced SQL Engine locks, see *Teradata Vantage™ - SQL Request and Transaction Processing*, B035-1142.

Locks are applied automatically as appropriate by Advanced SQL Engine. Optionally, you can specify the lock to be applied to a database object using the SQL LOCKING request modifier. For more information, see *Teradata Vantage™ - SQL Data Manipulation Language*, B035-1146. Locks are applied to different levels of database objects as required by the particular operation requiring the lock. Locks can be applied with the following granularities:

- Database
- Table
- Partition (row-partition)
- Range of rows
- Row (locks rows having a particular rowhash value, which may be more than a single row)

Lock Display can obtain lock information from the following:

- A specific AMP
- A group of AMPs
- All AMPs

---

**Note:**

For a Advanced SQL Engine system with many AMPs and a heavy load, be careful when selecting the number of AMPs from which to obtain information. The fewer the AMPs selected, the lower the volume of lock information generated, and the more manageable the output. Teradata recommends obtaining information from all AMPs only when the lock information for a specific transaction is needed.

---

Lock information can also be monitored and displayed using the Lock Viewer portlet in Teradata Viewpoint. For more information, see *Teradata® Viewpoint User Guide*, B035-2206.

### Runs From

- Database Window or comparable interface to the Teradata console subsystem, such as cnstern



- Linux command line
- Teradata Viewpoint Remote Console portlet

For general information on starting the utilities from different interfaces, see [Starting the Utilities](#). For information on Viewpoint, see *Teradata® Viewpoint User Guide*, B035-2206.

## Lock Modes

You can specify the following lock modes to a database object:

- Access
- Read
- Write
- Exclusive

### Lock-Mode Contention

The lock-mode contention among all outstanding lock requests determines which lock requests can be granted concurrently against a database object. This is illustrated by the contention matrix shown below.

| Lock Mode          | Implicit Access | Implicit Read | Implicit Write | Implicit Exclusive | Access | Read | Write | Exclusive |
|--------------------|-----------------|---------------|----------------|--------------------|--------|------|-------|-----------|
| Implicit Access    | X               | X             | X              | X                  | X      | X    | X     |           |
| Implicit Read      | X               | X             | X              | X                  | X      | X    |       |           |
| Implicit Write     | X               | X             | X              | X                  | X      |      |       |           |
| Implicit Exclusive | X               | X             | X              | X                  |        |      |       |           |
| Access             | X               | X             | X              |                    | X      | X    | X     |           |
| Read               | X               | X             |                |                    | X      | X    |       |           |
| Write              | X               |               |                |                    | X      |      |       |           |
| Exclusive          |                 |               |                |                    |        |      |       |           |

### Explicit and Implicit Lock Modes

Explicit lock modes are the actual locks that secure access to a database object. Usually, they are acquired at the onset of the actual transaction processing.

Implicit lock modes are pre-emptive locks used by the Teradata Lock Manager in the process of acquiring an explicit lock. Use of pre-emptive locks suggests that an explicit lock is not acquired in a single step but rather in a sequence of steps.

For example, consider acquiring an explicit table-level lock. Prior to acquiring the explicit table-level lock, an implicit database-level lock and an implicit table-level lock must be acquired first.

The lock mode is noted in the mode field of the Lock Display output. An implicit lock is identified by the appearance of an asterisk (\*). An explicit lock is implied when an asterisk is absent.

### Lock Request Status

A lock request against a database object can be granted or blocked depending on the following:

- The lock mode contention of all outstanding lock requests
- The success of acquiring all locks implied by the lock granularity of the request

Lock Display output shows separate sections for Granted and Blocked lock requests. In the case of a blocked request, the level of the database object is shown by marking the associated field in the output with the hash symbol (#). For example, if the blocked request involves a table, then the table field is marked with the # symbol.

## Locking and Row Partitions

To allow the Lock Manager to request locks at row-partition granularities, Vantage employs a value called a *rowkey*, which combines a row partition identifier (the internal partition number or IPN) with a rowhash value. The rowkey allows the Lock Manager to consider row partitions in determining the type of lock to use for a given transaction. The Lock Manager can thereby limit locks, when appropriate, to rows in a single row-partition, allowing concurrent access to the non-locked partitions of the table.

Partition-aware locks are identified in the output of Lock Display by having a PLLKind descriptor.

## Lock Display Utility Commands

Lock Display presents a command-line environment that allows the entry of the following Lock Display commands.

Commands are case-insensitive and may be abbreviated.

Lock Display interprets numeric input as hex (the default radix is hex). To enter decimal format numeric values, follow the number with a period character. For example, ten decimal must be entered as “10.”. An unmodified “10” is interpreted as hex value 0x10, equivalent to the decimal value sixteen.

---

#### Note:

Lock Display displays a series of question mark (?) characters if it does not find user, database, or table names in the cache or, in the case of DDL statements, in the Data Dictionary. For examples of this, see [DB](#) and [TABLE](#).

---

The following sections describe the commands and show examples of the output when various types of locks are present.

---

**Note:**

Because the TRAN command shows all transactions for which locks are being applied or blocked, it is described first, and the ROWKEY, ROWRANGE, PARTITION, and BLOCKERS commands, which display subsets of the TRAN command output, are described after TRAN.

---

## TRAN

The TRAN command displays currently running transactions with lock requests that have been granted or blocked.

### Syntax

```
{ TRAN | TR } [ ProclD Uniq1 Uniq2 | ALL ]
```

### Syntax Elements

#### *ProclD*

The virtual processor number of the parsing engine processor handling the transaction.

Since virtual processor numbers are designated as integer numbers, the corresponding value for this option normally is specified in decimal notation.

This value is the first component of a transaction ID.

#### *Uniq1*

A value that is normally specified as four hexadecimal digits.

This value is the second component of a transaction ID.

#### *Uniq2*

A value that is normally specified as four hexadecimal digits.

This value is the third component of a transaction ID.

#### ALL

That all blocked transactions and their corresponding blocker transactions will be considered.

ALL is the default if you do not specify *ProclD*, *Uniq1*, and *Uniq2*, which together constitute a transaction ID.

### Usage Notes

A transaction is an internal database concept. A transaction can have more than one blocking transaction. For example, a transaction can have five lock requests, and five transactions can block those same lock requests. In other words, if you have five tables, then conceivably, five other transactions can have the locks on those same five tables.

Each logically grouped display represents one lock request. Only object names relevant to a given lock request are displayed.

For example, only the database name is displayed for a database lock, whereas both a database name and a table name are displayed for a table lock.

The following table shows the components of TRAN command output.

| Component...              | Specifies...                                                                                                                                                          |
|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tran                      | the currently running transactions with locks being applied.                                                                                                          |
| Host                      | the logical host ID (origin of the transaction).                                                                                                                      |
| Session                   | the session number for the transaction.                                                                                                                               |
| Mode                      | the type of lock mode: <ul style="list-style-type: none"> <li>• Access</li> <li>• Read</li> <li>• Write</li> <li>• Exclusive</li> </ul>                               |
| User                      | the logon-ID for whom the lock is being requested.                                                                                                                    |
| Database                  | the name of the database with granted or blocked locks.                                                                                                               |
| Table                     | the name of the table with granted or blocked locks.                                                                                                                  |
| PLLKind                   | the type of partition-level lock that is applied. This only appears for locks on row-partitioned tables.                                                              |
| PROXY LOCK<br>PSEUDO LOCK | These are special locks used internally by the Lock Manager component of Vantage for the purpose of serializing all-AMP table-level or all-AMP partition range locks. |

### Example: Displaying lock requests on running transactions

In the following examples RPT1 and RPT2 are row-partitioned tables, and NOTRPT1 and NOTRPT2 are a non-partitioned tables.

```
-> Please enter your selection from the list:
tran
- The following amps are available:
    0      1      2      3
-> Which amp(s) do you want to request on (S=Sampling/A=all/C=cancel/Q=quit):
1
----- AMP 1 REPORTS 10 LOCK ENTRIES -----
                GRANTED LOCK REQUEST(S):
Tran: 30719 00015DCA
Hash Locks      :           1
Range Locks     :           6
Host: 1025 Session: 0, 1003 Mode: Rd   User: EXAMPLE
Database: EXAMPLE Table: RPT1
Host: 1025 Session: 0, 1003 Mode: Rd   User: EXAMPLE
Database: EXAMPLE Table: RPT1
```

```

PLLKind: RowHash + All Partitions Subtable ID: 1024
PROXY LOCK
      : Partition1 :      0 Row Hash1: 65535,65535
      : Partition2 :      0 Row Hash2: 65535,65535

Host: 1025 Session: 0, 1003 Mode: WR User: EXAMPLE
Database: EXAMPLE Table: RPT1
PLLKind: RowKey Subtable ID: 1024
      : Partition1 : 0003 Row Hash1: 31158,40503
      : Partition2 : 0003 Row Hash2: 31158,40503

Host: 1025 Session: 0, 1003 Mode: WR User: EXAMPLE
Database: EXAMPLE Table: RPT1
PLLKind: IPN Partition Range Subtable ID: 1024
      : Partition1 : 0004 Row Hash1: 0, 0
      : Partition2 : 0004 Row Hash2: 0, 0

Host: 1025 Session: 0, 1003 Mode: WR User: EXAMPLE
Database: EXAMPLE Table: RPT1
PLLKind: RowKey Range Subtable ID: 1024
      : Partition1 : 000A Row Hash1: 1, 17
      : Partition2 : 000A Row Hash2: 2, 17

Host: 1025 Session: 0, 1003 Mode: WR User: EXAMPLE
Database: EXAMPLE Table: RPT2
PLLKind: RowHash + IPN Part'n Range Subtable ID: 1024
      : Partition1 : 0007 Row Hash1: 1, 17
      : Partition2 : 0007 Row Hash2: 1, 17

Host: 1025 Session: 0, 1003 Mode: WR User: EXAMPLE
Database: EXAMPLE Table: NOTRPT1
Row Range Lock Subtable ID: 1024
Row Hash1: 31158,40496 Row Hash2: 31158,40502

Host: 1025 Session: 0, 1003 Mode: WR User: EXAMPLE
Database: EXAMPLE Table: NOTRPT2
Row Hash Lock Subtable ID: 1024
Row Hash1: 31158,40503

BLOCKED LOCK REQUEST(S):

Tran: 30719 00015E1B
Host: 0 Session: 0, 0 Mode: WR User: ALL
Database: EXAMPLE Table: RPT1#
PLLKind: RowKey Subtable ID: 1024
      : Partition1 : 0003 Row Hash1: 31158,40503
      : Partition2 : 0003 Row Hash2: 31158,40503#

Tran: 30719 00015E1C
Host: 0 Session: 0, 0 Mode: WR User: ALL
Database: EXAMPLE Table: NOTRPT2
Row Hash Lock Subtable ID: 1024
Row Hash1: 31158,40503#

```

## ROWKEY

The ROWKEY command displays granted and blocked row-level locks.

To request row-level locks the Lock Manager uses the rowkey value, described in [Locking and Row Partitions](#), which allows the Lock Manager to request row-level locks that are limited to the rows in a single partition:

- For row-partitioned tables, if the IPN portion of the rowkey being locked is a non-zero value, the lock is applied only to rows in the specified partition. These rows have rowhash values matching the rowhash portion of the rowkey being locked.

If the IPN portion of the rowkey being locked is zero, the lock is applied to rows in all partitions of the table that match the rowhash portion of the rowkey.

- For non-row-partitioned tables, the partition number portion of the rowkey is always zero, and any rows in the table with rowhash values matching the rowhash portion of the rowkey are locked.

### Syntax

```
{ ROWKEY | ROWK } [
  DBname.Tablename TypeAndIndex [,] Partition RowHash1 RowHash2 |
  ALL
]
```

### Syntax Elements

#### ***DBname***

The name of a database.

#### ***Tablename***

The name of a table in *DBname*.

#### ***TypeAndIndex***

A subtable identifier.

A table is composed logically of one or more subtables. *TypeAndIndex* specifies one of these subtables. For example:

- 0 is the table header.
- hex 400 (decimal 1024) is a primary subtable.
- hex values 404, 408, and 40C (decimal values 1028, 1032, and 1036), and other +4 incremental values, are secondary index subtables.
- hex values 800, C00, and 1000 (decimal values 2048, 3072, and 4096), and other multiples of hex 400 (decimal 1024) are fallback subtables.

**Partition**

The internal partition number for the rowkey lock. Both decimal and hexadecimal numbers are accepted. A decimal number must be followed by a period.

*Partition* is zero for non-row-partitioned tables.

**RowHash1****RowHash2**

The first and second portions of the row hash, representing the row for which lock information is to be displayed.

Both decimal and hexadecimal numbers are accepted. A decimal number must be followed by a period.

**ALL**

That all tables that have a rowkey or rowhash lock applied are considered.

ALL is the default if you do not specify an object name.

**Usage Notes**

The following table shows the components of ROWKEY command output.

| Component... | Specifies...                                                                                                                            |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| Tran         | currently running transactions with locks being applied.                                                                                |
| Hash Locks   | the total number of rowhash locks applied by the transaction.                                                                           |
| Range Locks  | the total number of range locks applied by the transaction.                                                                             |
| RowKey Locks | the number of rowkey locks applied by the transaction.                                                                                  |
| Host         | the logical host ID (origin of the transaction).                                                                                        |
| Session      | the session number for the transaction.                                                                                                 |
| Mode         | the type of lock mode: <ul style="list-style-type: none"> <li>• Access</li> <li>• Read</li> <li>• Write</li> <li>• Exclusive</li> </ul> |
| User         | the logon-ID for whom the lock is being requested.                                                                                      |
| Database     | the name of the database with granted or blocked locks.                                                                                 |
| Table        | the name of the table with granted or blocked locks.                                                                                    |



| Component...              | Specifies...                                                                                                                                                                                                                                                                                                                                      |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PLLKind                   | the type of partition-level lock that is applied. This only appears for locks on row-partitioned tables.                                                                                                                                                                                                                                          |
| Subtable ID               | the identifier of the subtable to which the rowkey lock is applied                                                                                                                                                                                                                                                                                |
| Partition1 and Partition2 | the internal partition numbers of the rowkeys that begin and end the range of locked rows. For partition-level locks displayed by the ROWKEY command, Partition1 and Partition2 always match.<br>The internal partition number is part of the rowid of the physical row. It is displayed in hexadecimal as [[[hhhh.] hhhh.] [hhhh.hhhh]] or zero. |
| RowHash1 and RowHash2     | the first and second portions of the row hash, representing the row for which lock information is to be displayed.<br>Both decimal and hexadecimal numbers are accepted. A decimal number must be followed by a period.                                                                                                                           |

In the following examples table RPT1 is a row-partitioned table, and table NOTRPT2 is a non-partitioned table.

### Examples: ROWKEY command examples

This example shows the following granted lock requests:

- a proxy lock, used to serialize locking to prevent deadlocks
- a write lock on the row or rows having rowhash 31158,40503 in a row-partitioned table. Only the rows in partition number 3 are locked.
- a write lock on the row or rows having rowhash 31158,40503 in a non-partitioned table.

```
-> Please enter your selection from the list:
rowkey
- The following amps are available:

    0      1      2      3

-> Which amp(s) do you want to request on (S=Sampling/A=all/C=cancel/Q=quit):
1

----- AMP 1 REPORTS 5 LOCK ENTRIES -----

          GRANTED LOCK REQUEST(S):

Tran: 30719 00015DCA
Hash Locks      :          1
Range Locks     :          6
RowKey Locks    :          2

Host: 1025 Session:    0, 1003 Mode: Rd   User: EXAMPLE
Database: EXAMPLE Table: RPT1
PLLKind: RowHash + All Partitions   Subtable ID: 1024
PROXY LOCK

                        : Partition1 :                      0 Row
```

```

Hash1: 65535,65535
                                : Partition2 :                0 Row
Hash2: 65535,65535

Host: 1025 Session:      0, 1003 Mode: WR   User: EXAMPLE
Database: EXAMPLE Table: RPT1
PLLKind: RowKey          Subtable ID: 1024

                                : Partition1 :                0003 Row
Hash1: 31158,40503
                                : Partition2 :                0003 Row
Hash2: 31158,40503

Host: 1025 Session:      0, 1003 Mode: WR   User: EXAMPLE
Database: EXAMPLE Table: NOTRPT2
Row Hash Lock              Subtable ID: 1024
Row Hash1: 31158,40503

BLOCKED LOCK REQUEST(S):

Tran: 30719 00015E1B
Host:      0 Session:      0,      0 Mode: WR   User: ALL
Database: EXAMPLE Table: RPT1#
PLLKind: RowKey          Subtable ID: 1024

                                : Partition1 :                0003 Row
Hash1: 31158,40503
                                : Partition2 :                0003 Row
Hash2: 31158,40503
Tran: 30719 00015E1C
Host:      0 Session:      0,      0 Mode: WR   User: ALL
Database: EXAMPLE Table: NOTRPT2
Row Hash Lock              Subtable ID: 1024
Row Hash1: 31158,40503#

-> Please enter your selection from the list:
rowkey example.rpt1 400, 3 31158. 40503.
- The following amps are available:

      0      1      2      3

-> Which amp(s) do you want to request on (S=Sampling/A=all/C=cancel/Q=quit):
1

----- AMP 1 REPORTS 2 LOCK ENTRIES -----

GRANTED LOCK REQUEST(S):

Tran: 30719 00015DCA
Hash Locks      :                1
Range Locks     :                6
RowKey Locks    :                1

Host: 1025 Session:      0, 1003 Mode: WR   User: EXAMPLE
Database: EXAMPLE Table: RPT1
PLLKind: RowKey          Subtable ID: 1024

                                : Partition1 :                0003 Row
Hash1: 31158,40503
                                : Partition2 :                0003 Row

```

Hash2: 31158,40503

BLOCKED LOCK REQUEST(S):

Tran: 30719 00015E1B  
 Host: 0 Session: 0, 0 Mode: WR User: ALL  
 Database: EXAMPLE Table: RPT1#  
 PLLKind: RowKey Subtable ID: 1024

|                    |                |          |
|--------------------|----------------|----------|
| Hash1: 31158,40503 | : Partition1 : | 0003 Row |
|                    | : Partition2 : | 0003 Row |
| Hash2: 31158,40503 |                |          |

-> Please enter your selection from the list:  
 rowkey example.notrpt2 400, 0 31158. 40503.

- The following amps are available:

0 1 2 3

-> Which amp(s) do you want to request on (S=Sampling/A=all/C=cancel/Q=quit):  
 1

----- AMP 1 REPORTS 2 LOCK ENTRIES -----

GRANTED LOCK REQUEST(S):

Tran: 30719 00015DCA  
 Hash Locks : 1  
 Range Locks : 6  
 Host: 1025 Session: 0, 1003 Mode: WR User: EXAMPLE  
 Database: EXAMPLE Table: NOTRPT2  
**Row Hash Lock** Subtable ID: 1024  
 Row Hash1: 31158,40503

BLOCKED LOCK REQUEST(S):

Tran: 30719 00015E1C  
 Host: 0 Session: 0, 0 Mode: WR User: ALL  
 Database: EXAMPLE Table: NOTRPT2  
**Row Hash Lock** Subtable ID: 1024  
 Row Hash1: 31158,40503#

## ROWRANGE

The ROWRANGE command displays granted and blocked row range locks. A row range lock locks a range of rows (the rows identified by a range of contiguous rowhash values in non-row-partitioned tables or a range of contiguous rowkey values in row-partitioned tables). Rowkey values are described in [Locking and Row Partitions](#).

### Syntax

```
{ ROWRANGE | ROWR } [
  DBname.Tablename TypeAndIndex |
  ALL
]
```

### Syntax Elements

#### *DBname*

The name of a database.

#### *Tablename*

The name of a table in *DBname*.

#### *TypeAndIndex*

A subtable identifier.

A table is composed logically of one or more subtables. *TypeAndIndex* specifies one of these subtables. For example:

- 0 is the table header.
- hex 400 (decimal 1024) is a primary subtable.
- hex values 404, 408, and 40C (decimal values 1028, 1032, and 1036), and other +4 incremental values, are secondary index subtables.
- hex values 800, C00, and 1000 (decimal values 2048, 3072, and 4096), and other multiples of hex 400 (decimal 1024) are fallback subtables.

#### ALL

That all tables that have a rowrange-level lock request are considered.

ALL is the default if you do not specify the command parameters.

### Usage Notes

The following table shows the components of ROWRANGE command output.

| Component...              | Specifies...                                                                                                                                                                                                                                   |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tran                      | the currently running transactions with locks being applied.                                                                                                                                                                                   |
| Hash Locks                | the total number of rowhash locks applied by the transaction.                                                                                                                                                                                  |
| Range Locks               | the total number of range locks applied by the transaction.                                                                                                                                                                                    |
| Row Range Locks           | the number of row range locks applied by the transaction.                                                                                                                                                                                      |
| Host                      | the logical host ID (origin of the transaction).                                                                                                                                                                                               |
| Session                   | the session number for the transaction.                                                                                                                                                                                                        |
| Mode                      | the type of lock mode: <ul style="list-style-type: none"> <li>• Access</li> <li>• Read</li> <li>• Write</li> <li>• Exclusive</li> </ul>                                                                                                        |
| User                      | the logon-ID for whom the lock is being requested.                                                                                                                                                                                             |
| Database                  | the name of the database with granted or blocked locks.                                                                                                                                                                                        |
| Table                     | the name of the table with granted or blocked locks.                                                                                                                                                                                           |
| PLKind                    | the type of partition-level lock that is applied. This only appears for locks on row-partitioned tables.                                                                                                                                       |
| Subtable ID               | the identifier of the subtable to which the row range lock is applied                                                                                                                                                                          |
| Partition1 and Partition2 | the internal partition numbers of the rowkeys that begin and end the range of locked rows.<br>The internal partition number is part of the rowid of the physical row. It is displayed in hexadecimal as [[[hhhh.] hhhh.] [hhhh.hhhh]] or zero. |
| RowHash1 and RowHash2     | the first and second portions of the row hash, representing the row for which lock information is to be displayed.<br>Both decimal and hexadecimal numbers are accepted. A decimal number must be followed by a period.                        |

In the following examples RPT1 is a row-partitioned table, and NOTRPT1 is a non-partitioned table.

### Examples: ROWRANGE command examples

This example shows:

- a write lock on the range of rows beginning at rowhash 1,17 and ending at rowhash 2,17 in a row-partitioned table. Only the rows in partition number 000A are locked.
- a write lock on the range of rows beginning at rowhash 31158,40496 and ending at rowhash 31158,40502 in a non-partitioned table.

-> Please enter your selection from the list:  
rowrange

```

- The following amps are available:
    0      1      2      3

-> Which amp(s) do you want to request on (S=Sampling/A=all/C=cancel/Q=quit):
1

----- AMP 1 REPORTS 2 LOCK ENTRIES -----

          GRANTED LOCK REQUEST(S):

Tran: 30719 00015DCA
Hash Locks      :          1
Range Locks     :          6
Row Range Locks :          2
Host: 1025 Session: 0, 1003 Mode: WR   User: EXAMPLE
Database: EXAMPLE Table: RPT1
PLKind:  RowKey Range          Subtable ID: 1024
          : Partition1 :          000A Row Hash1:      1,   17
          : Partition2 :          000A Row Hash2:      2,   17

Host: 1025 Session: 0, 1003 Mode: WR   User: EXAMPLE
Database: EXAMPLE Table: NOTRPT1
Row Range Lock          Subtable ID: 1024
Row Hash1: 31158,40496 Row Hash2: 31158,40502

-> Please enter your selection from the list:
rowrange example.rpt1 400
- The following amps are available:
    0      1      2      3

-> Which amp(s) do you want to request on (S=Sampling/A=all/C=cancel/Q=quit):
1

----- AMP 1 REPORTS 1 LOCK ENTRIES -----

          GRANTED LOCK REQUEST(S):

Tran: 30719 00015DCA
Hash Locks      :          1
Range Locks     :          6
Row Range Locks :          1
Host: 1025 Session: 0, 1003 Mode: WR   User: EXAMPLE
Database: EXAMPLE Table: RPT1
PLKind:  RowKey Range          Subtable ID: 1024
          : Partition1 :          000A Row Hash1:      1,   17
          : Partition2 :          000A Row Hash2:      2,   17

-> Please enter your selection from the list:
rowrange example.notrpt1 400
- The following amps are available:
    0      1      2      3

-> Which amp(s) do you want to request on (S=Sampling/A=all/C=cancel/Q=quit):
1

```

```
----- AMP 1 REPORTS 1 LOCK ENTRIES -----  
  
          GRANTED LOCK REQUEST(S):  
  
Tran: 30719 00015DCA  
Hash Locks      :          1  
  
Range Locks     :          6  
  
Row Range Locks:          1  
  
Host: 1025 Session:      0, 1003 Mode: WR   User: EXAMPLE  
Database: EXAMPLE Table: NOTRPT1  
Row Range Lock Subtable ID: 1024  
Row Hash1: 31158,40496 Row Hash2: 31158,40502
```

## PARTITION

The PARTITION command displays granted and blocked partition-aware locks. These types of locks lock a range of row partitions.

### Note:

Currently, the partition range is limited by default to a single row partition, except in the case of the DBC.AccessRights table, where a lock can be applied across a range of row partitions to a single rowhash or all rowhashes.

### Syntax

```
{ PARTITION | PART } [
    DBname.Tablename TypeAndIndex |
    ALL
]
```

### Syntax Elements

#### **DBname**

The name of a database.

#### **Tablename**

The name of a table in *DBname*.

#### **TypeAndIndex**

A subtable identifier.

A table is composed logically of one or more subtables. *TypeAndIndex* specifies one of these subtables. For example:

- 0 is the table header.
- hex 400 (decimal 1024) is a primary subtable.
- hex values 404, 408, and 40C (decimal values 1028, 1032, and 1036), and other +4 incremental values, are secondary index subtables.
- hex values 800, C00, and 1000 (decimal values 2048, 3072, and 4096), and other multiples of hex 400 (decimal 1024) are fallback subtables.

#### **ALL**

That all row-partitioned tables that have a lock applied to a range of partitions are considered.

ALL is the default if you do not specify an object name.



## Usage Notes

The following table shows the components of partition command output.

| Component...              | Specifies...                                                                                                                                                                                                                                                                                                                                                             |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tran                      | currently running transactions with locks being applied.                                                                                                                                                                                                                                                                                                                 |
| Hash Locks                | Hash locks are not relevant to the PARTITION command.                                                                                                                                                                                                                                                                                                                    |
| Range Locks               | the total number of range locks applied by the transaction.                                                                                                                                                                                                                                                                                                              |
| Partition Locks           | the number of partition range locks applied by the transaction.                                                                                                                                                                                                                                                                                                          |
| Host                      | the logical host ID (origin of the transaction).                                                                                                                                                                                                                                                                                                                         |
| Session                   | the session number for the transaction.                                                                                                                                                                                                                                                                                                                                  |
| Mode                      | the type of lock mode: <ul style="list-style-type: none"> <li>• Access</li> <li>• Read</li> <li>• Write</li> <li>• Exclusive</li> </ul>                                                                                                                                                                                                                                  |
| User                      | the logon-ID for whom the lock is being requested.                                                                                                                                                                                                                                                                                                                       |
| Database                  | the name of the database with granted or blocked locks.                                                                                                                                                                                                                                                                                                                  |
| Table                     | the name of the table with granted or blocked locks.                                                                                                                                                                                                                                                                                                                     |
| PLKind                    | the type of partition-level lock that is applied. This only appears for locks on row-partitioned tables.                                                                                                                                                                                                                                                                 |
| Subtable ID               | the identifier of the subtable to which the partition range lock is applied.                                                                                                                                                                                                                                                                                             |
| Partition1 and Partition2 | the internal partition numbers that define the beginning and ending partitions in the range of locked partitions. <p><b>Note:</b></p> <p>The partition range is currently limited to a single partition.</p> <p>The internal partition number is part of the rowid of the physical row. It is displayed in hexadecimal as [[ hhhh. ] hhhh. ] [ hhhh. hhhh ] or zero.</p> |
| RowHash1 and RowHash2     | the first and second portions of the row hash, representing the locked rowhash in the range of partitions. <p>Both decimal and hexadecimal numbers are accepted. A decimal number must be followed by a period.</p>                                                                                                                                                      |

In the following examples RPT1 and RPT2 are row-partitioned tables.

### Examples: PARTITION command examples

This example shows:

- a write lock on all rows in partition 0004 in a row-partitioned table.

- a write lock on the row or rows having rowhash 1,17 in partition 0007 in a non-partitioned table.

```

-> Please enter your selection from the list:
partition
- The following amps are available:

    0    1    2    3

-> Which amp(s) do you want to request on (S=Sampling/A=all/C=cancel/Q=quit):
1

----- AMP 1 REPORTS 2 LOCK ENTRIES -----

          GRANTED LOCK REQUEST(S):

Tran: 30719 00015DCA
Hash Locks      :          1
Range Locks     :          6
Partition Locks :          2
Host: 1025 Session: 0, 1003 Mode: WR   User: EXAMPLE
Database: EXAMPLE Table: RPT1
PLLKind: IPN Partition Range          Subtable ID: 1024

0,    0          : Partition1 :          0004 Row Hash1:
0,    0          : Partition2 :          0004 Row Hash2:

Host: 1025 Session: 0, 1003 Mode: WR   User: EXAMPLE
Database: EXAMPLE Table: RPT2
PLLKind: RowHash + IPN Part'n Range  Subtable ID: 1024

1,    17          : Partition1 :          0007 Row Hash1:
1,    17          : Partition2 :          0007 Row Hash2:

-> Please enter your selection from the list:
partition example.rpt1 400
- The following amps are available:

    0    1    2    3

-> Which amp(s) do you want to request on (S=Sampling/A=all/C=cancel/Q=quit):
1

----- AMP 1 REPORTS 1 LOCK ENTRIES -----

          GRANTED LOCK REQUEST(S):

Tran: 30719 00015DCA
Hash Locks      :          1
Range Locks     :          6
Partition Locks :          1
Host: 1025 Session: 0, 1003 Mode: WR   User: EXAMPLE
Database: EXAMPLE Table: RPT1
PLLKind: IPN Partition Range          Subtable ID: 1024

```

```

0,    0                : Partition1 :          0004 Row Hash1:
0,    0                : Partition2 :          0004 Row Hash2:

-> Please enter your selection from the list:
partition example.rpt2 400
- The following amps are available:
    0    1    2    3

-> Which amp(s) do you want to request on (S=Sampling/A=all/C=cancel/Q=quit):
1

----- AMP 1 REPORTS 1 LOCK ENTRIES -----
          GRANTED LOCK REQUEST(S):

Tran: 30719 00015DCA
Hash Locks      :          1
Range Locks     :          6
Partition Locks :          1

Host: 1025 Session:    0, 1003 Mode: WR   User: EXAMPLE
Database: EXAMPLE Table: RPT2
PLKind: RowHash + IPN Part'n Range Subtable ID: 1024

1,    17                : Partition1 :          0007 Row Hash1:
1,    17                : Partition2 :          0007 Row Hash2:

```

## BLOCKERS

The BLOCKERS command displays currently blocked transactions and the corresponding blocking transactions with granted lock requests.

### Syntax

```
{ BLOCKERS | B } TRAN
  [ ProclD Uniq1 Uniq2 | ALL ]
  [ LIMIT [ NUMBER | NONE ] ]
```

### Syntax Elements

#### *ProclD*

The virtual processor number of the parsing engine processor handling the transaction.

Since virtual processor numbers are designated as integer numbers, the corresponding value for this option normally is specified in decimal notation.

This number is the first component of a transaction ID.

#### *Uniq1*

A value that is normally specified as four hexadecimal digits.

This value is the second component of a transaction ID.

#### *Uniq2*

A value that is normally specified as four hexadecimal digits.

This value is the third component of a transaction ID.

#### **ALL**

That all blocked transactions and their corresponding blocker transactions will be considered.

ALL is the default if you do not specify a transaction ID.

#### **LIMIT**

The number of blocker transactions to consider for a blocked transaction.

#### **NUMBER**

The desired limiting value.

#### **NONE**

That all blocker transactions for a blocked transaction are considered.

Specifying NONE corresponds to specifying zero for Number.

## Usage Notes

A transaction is an internal database concept. A transaction can have more than one blocking transaction. For example, a transaction can have five lock requests, and five transactions can block those same lock requests. In other words, if you have five tables, then conceivably, five other transactions can have the locks on those same five tables.

The following table shows the components of BLOCKERS command output.

| Component....                     | Includes the...                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Number of Blocked Trans displayed | total number of both blocked and blocker transactions.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Blocked Trans                     | number of the blocked transaction and the following information: <ul style="list-style-type: none"> <li>• <b>Number of blockers displays</b><br/>Specifies blocker entry count.</li> <li>• <b>Number of blockers exists</b><br/>Specifies actual blocker count.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Blocker Trans                     | number of blocking transactions and the following information: <ul style="list-style-type: none"> <li>• <b>Lock mode</b> specifies a type of lock mode:               <ul style="list-style-type: none"> <li>◦ Access</li> <li>◦ Read</li> <li>◦ Write</li> <li>◦ Exclusive</li> </ul> </li> <li>• <b>Lock status</b> specifies a type of status of the lock request:               <ul style="list-style-type: none"> <li>◦ Granted</li> <li>◦ Waiting</li> </ul> </li> <li>• <b>Lock objectType</b> specifies a type of object that is locked:               <ul style="list-style-type: none"> <li>◦ Database</li> <li>◦ Table</li> <li>◦ Rowrange</li> <li>◦ Row</li> </ul> </li> <li>• <b>Lock PLLkind</b> specifies the type of partition-level lock applied to row-partitioned tables:               <ul style="list-style-type: none"> <li>◦ RowHash + All Partitions</li> <li>◦ RowKey</li> <li>◦ IPN Partition Range (IPN is internal partition number)</li> <li>◦ RowKey Range</li> <li>◦ RowHash + IPN Part'n Range</li> </ul> </li> <li>• <b>Lock objectID</b> specifies an ID of the locked object:               <ul style="list-style-type: none"> <li>◦ Database ID</li> <li>◦ Database Name</li> <li>◦ Table ID</li> <li>◦ Table Name</li> </ul> </li> </ul> |

| Component.... | Includes the...                                                                                                              |
|---------------|------------------------------------------------------------------------------------------------------------------------------|
|               | <ul style="list-style-type: none"> <li>◦ Partition1</li> <li>◦ RowHash1</li> <li>◦ Partition2</li> <li>◦ RowHash2</li> </ul> |

In the following examples RPT1 is a row-partitioned table, and NOTRPT2 is a non-partitioned table.

### Examples: BLOCKERS command

```
-> Please enter your selection from the list:
blockers
- The following amps are available:

    0      1      2      3

-> Which amp(s) do you want to request on (S=Sampling/A=all/C=cancel/Q=quit):
1

----- AMP 1 REPORTS 2 LOCK ENTRIES -----
Number of Blocked Trans displayed :      2
=====
Blocked Trans : 30719 00015E1C
  Number of blockers displays :      1
  Number of blockers exists   :      1
  Blocker Trans : 30719 00015DCA
    lock mode      : Write
    lock status    : Granted
    lock objectType: Row
    lock objectID  : DBID      : 00000402
                  : DBNAME    : EXAMPLE
                  : TableID    : 00000AA0,0400
                  : TableName  : NOTRPT2
                  : RowHash1   : 79B69E37
                  : RowHash2   : 00000000
Blocked Trans : 30719 00015E1B
  Number of blockers displays :      1
  Number of blockers exists   :      1
  Blocker Trans : 30719 00015DCA
    lock mode      : Write
    lock status    : Granted
    lock objectType: RowRange
    lock PLLKind   : RowKey
    lock objectID  : DBID      : 00000402
                  : DBNAME    : EXAMPLE
                  : TableID    : 00000A9D,0400
                  : TableName  : RPT1
                  : Partition1:      0003
                  : RowHash1   :      79B69E37
                  : Partition2:      0003
                  : RowHash2   :      79B69E37

-> Please enter your selection from the list:
blockers tran 30719. 0001 5e1c
- The following amps are available:

    0      1      2      3

-> Which amp(s) do you want to request on (S=Sampling/A=all/C=cancel/Q=quit):
1
```

```

----- AMP 1 REPORTS 1 LOCK ENTRIES -----
Number of Blocked Trans displayed :      1
=====
Blocked Trans : 30719 00015E1C
  Number of blockers displays :          1
  Number of blockers exists   :          1
  Blocker Trans : 30719 00015DCA
    lock mode      : Write
    lock status    : Granted
    lock objectType : Row
    lock objectID  : DBID       : 00000402
                  : DBNAME     : EXAMPLE
                  : TableID    : 00000AA0,0400
                  : TableName  : NOTRPT2
                  : RowHash1   : 79B69E37
                  : RowHash2   : 00000000

```

-> Please enter your selection from the list:

blockers tran 30719. 0001 5e1b

- The following amps are available:

0      1      2      3

-> Which amp(s) do you want to request on (S=Sampling/A=all/C=cancel/Q=quit):

1

```

----- AMP 1 REPORTS 1 LOCK ENTRIES -----
Number of Blocked Trans displayed :      1
=====
Blocked Trans : 30719 00015E1B
  Number of blockers displays :          1
  Number of blockers exists   :          1
  Blocker Trans : 30719 00015DCA
    lock mode      : Write
    lock status    : Granted
    lock objectType : RowRange
    lock PLLKind   : RowKey
    lock objectID  : DBID       : 00000402
                  : DBNAME     : EXAMPLE
                  : TableID    : 00000A9D,0400
                  : TableName  : RPT1
                  : Partition1 :          0003
                  : RowHash1   :          79B69E37
                  : Partition2 :          0003
                  : RowHash2   :          79B69E37

```

## DB

The DB command displays granted and blocked databases-level locks.

### Syntax

```
{ DB | D } [ DBname | ALL ]
```

### Syntax Elements

#### *DBname*

The name of a database.

#### ALL

That all databases that have a database-level lock request will be considered.

ALL is the default if you do not specify a database name.

### Usage Notes

The following table shows the components of DB command output.

| Component... | Specifies...                                                                                                                    |
|--------------|---------------------------------------------------------------------------------------------------------------------------------|
| Tran         | currently running transactions with locks being applied.                                                                        |
| Host         | the logical host ID (the origin of the transaction).                                                                            |
| Session      | the session number for the transaction.                                                                                         |
| Mode         | the lock mode: <ul style="list-style-type: none"> <li>• Access</li> <li>• Read</li> <li>• Write</li> <li>• Exclusive</li> </ul> |
| User         | the logon-ID for whom the lock is being requested.                                                                              |
| Database     | the name of the database with granted or blocked locks.                                                                         |

### Examples: DB command examples

The following example shows locks on a sampling on all AMPs for the RECBQDTAC database:

```
>lokdisp
Amp Utility
  LOCK DISPLAY UTILITY
LOCK DISPLAY UTILITY Command String Syntax:

  Help or ?
```



```

TRan      [ProcId Uniq1 Uniq2] | [ALL]
Db        [DBname] | [ALL]
Table     [DBname.Tablename] | [ALL]
ROWRange  [DBname.Tablename TypeAndIndex] | [ALL]
ROWKey    [DBname.Tablename TypeAndIndex, Partition RowHash1
           RowHash2] | [ALL]
Partition [DBname.Tablename TypeAndIndex] | [ALL]
Blockers  [TRAN [ProcId Uniq1 Uniq2] | [ALL]] | [LIMIT [Number]
           | [NONE]]
Quit

-> Please enter your selection from the list: db
- The following amps are available:

    0      1      2      3

-> Which amp(s) do you want to request on (S=Sampling/A=all/C=cancel/Q=quit):
a>
a

----- AMP 0 REPORTS 1 LOCK ENTRIES -----

          GRANTED LOCK REQUEST(S):

Tran: 16383 00000114
Host:  2049 Session:      0, 1000 Mode: WR*  User: DBC
Database: RECBDDQTAC

----- AMP 2 REPORTS 1 LOCK ENTRIES -----

          GRANTED LOCK REQUEST(S):

Tran: 16383 00000114
Host:  2049 Session:      0, 1000 Mode: WR*  User: DBC
Database: RECBDDQTAC

```

The following example tries to display database-level locks while trying to create database USER1. The first lock is an Intentional Write Lock on database DBC (user DBC), and the second lock is an Intentional Exclusive Lock on database USER1 (user DBC).

```

LOKDISP >>
DB ALL

- The following amps are available:

    0

-> Which amp(s) do you want to request on (S=Sampling/A=all/C=cancel/Q=quit):
A

----- AMP 0 REPORTS 2 LOCK ENTRIES -----

          GRANTED LOCK REQUEST(S):

Tran: 16383 000008B8
Host:  7169 Session:      0, 1010 Mode: WR*  User: ??????????????????????????????
Database: ??????????????????????????????

Host:  7169 Session:      0, 1010 Mode: EX*  User: ??????????????????????????????
Database: ??????????????????????????????

```

## TABLE

The TABLE command displays granted and blocked table-level locks.

### Syntax

```
{ TABLE | TA } [ DBname.Tablename | ALL ]
```

### Syntax Elements

#### *DBname*

The name of a database.

#### *Tablename*

The name of a table in *DBname*.

#### ALL

That all tables (from all databases) that have a table-level lock request are considered.

ALL is the default if you do not specify an object name.

### Usage Notes

The following table shows the components of TABLE command output.

| Component... | Specifies...                                                                                                                            |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| Tran         | the currently running transactions with locks being applied.                                                                            |
| Host         | the logical host ID (origin of the transaction).                                                                                        |
| Session      | the session number for the transaction.                                                                                                 |
| Mode         | the type of lock mode: <ul style="list-style-type: none"> <li>• Access</li> <li>• Read</li> <li>• Write</li> <li>• Exclusive</li> </ul> |
| User         | the logon-ID for whom the lock is being requested.                                                                                      |
| Database     | the name of the database with granted or blocked locks.                                                                                 |
| Table        | the name of the table with granted or blocked locks.                                                                                    |

### Examples: TABLE command examples

The following example shows the locks on AMP 0 on the database RECBDQTAC and table T1:

```
table
```

```
- The following amps are available:
```

```
0      1      2      3
```

```
-> Which amp(s) do you want to request on (S=Sampling/A=all/C=cancel/Q=quit):
```

```
> a
```

```
a
```

```
----- AMP 0 REPORTS 1 LOCK ENTRIES -----
```

```
GRANTED LOCK REQUEST(S):
```

```
Tran: 16383 00000114
```

```
Host: 2049 Session: 0, 1000 Mode: WR* User: DBC
```

```
Database: RECBDQTAC Table: T1
```

```
----- AMP 2 REPORTS 1 LOCK ENTRIES -----
```

```
GRANTED LOCK REQUEST(S):
```

```
Tran: 16383 00000114
```

```
Host: 2049 Session: 0, 1000 Mode: WR* User: DBC
```

```
Database: RECBDQTAC Table: T1
```

The following is an example of trying to display table-level locks on a table while trying to create the table.

```
table all
```

```
- The following amps are available:
```

```
0
```

```
-> Which amp(s) do you want to request on (S=Sampling/A=all/C=cancel/Q=quit):
```

```
A
```

```
Host: 7169 Session: 0, 1441 Mode: EX User: DBC
```

```
Database: STAFF Table: ?????????????????????????????????
```

```
Table name is not printed.
```



## HELP

The HELP command provides general help for Lock Display.

### Syntax

```
{ HELP | H | ? }
```

### Example: Lock Display Help

The following example shows the first portion of the basic Lock Display Help:

```
-> Please enter your selection from the list:
HELP

LOCK DISPLAY UTILITY

An optional command string may be used to limit the display to specific
requests; otherwise, all requests are displayed except for BLOCKERS.

Commands are case-insensitive and may be abbreviated. The default radix
for numeric entries is hex, but may be forced to decimal by terminating
with a period. For example, ten decimal must be entered as "10.", since
an unmodified "10" is interpreted in hex as 0x10 or sixteen decimal.

For the TRAN command, a set of lines displayed represents one lock
request.

Only object names relevant to a given lock request are displayed: for
example, only database name is displayed for a database lock, whereas
both database name and table name are displayed for a table lock. A "#"
after an object name means the lock has not yet been granted for that
level. For example, "Databasename Tablename# Rowhash1#" means the lock
has been granted at database level, but not yet at table or rowhash level.

--more-- Please press <enter> to continue.
```

## QUIT

The QUIT command exits Lock Display.

### Syntax

```
{ QUIT | Q }
```

# Query Configuration (qryconfig)

---

**Note:**

The utility described is not available for as-a-service customers.

---

The Query Configuration utility, qryconfig, reports the current Advanced SQL Engine configuration from a Advanced SQL Engine system console running the Database Window or from a host terminal. This configuration is defined by Teradata, and may be changed by Teradata Services.

The Query Configuration utility is also referred to as Configuration Display.

## Runs From

- Database Window or comparable interface to the Teradata console subsystem, such as cnstern
- Teradata Viewpoint Remote Console portlet
- Host Utility Console
- Linux command line

For general information on starting the utilities from different interfaces, see [Starting the Utilities](#). For information on Viewpoint, see *Teradata® Viewpoint User Guide*, B035-2206.

## About Query Configuration

### Vprocs and Physical Processors

The Query Configuration utility displays configuration information about the nodes in a Teradata system. This utility supplies information about the PEs associated with the node and the AMPs.

While physical processor identifiers are presented where appropriate in the configuration displays, the focus of the Query Configuration utility is on the vprocs managed by the node.

These vprocs exist within a previously defined physical configuration. Use the Parallel Upgrade Tool (PUT) to configure parts of the physical configuration, such as creating Logical Units (LUNs) on disk arrays. See *Parallel Upgrade Tool (PUT) Reference*.

### Display Options

The Query Configuration utility can display a variety of status and configuration information for the entire Teradata system or portions of the system including the following:

- Complete configuration
- Node, online or offline
- AMPs, online or offline

- PEs, online or offline

## Query Configuration Options

You can display all or parts of the current Teradata system configuration by entering various Query Configuration options (also called key words). Enter help or a question mark (?) at the Query Configuration command prompt to see a list of the available options:

```
Enter display option,  QUIT or END to terminate.
> help
```

The following key words return the indicated display.

|                     |                           |
|---------------------|---------------------------|
| All;                | - complete configuration. |
| Processors;         | - processor status.       |
| Online Processors;  | - online processors.      |
| Offline Processors; | - offline processors.     |
| AMPs;               | - AMP status.             |
| Online AMPs;        | - online AMPs.            |
| Offline AMPs;       | - offline AMPs.           |
| PEs;                | - PE status.              |
| Online PEs;         | - online PEs.             |
| Offline PEs;        | - offline PEs.            |

The terminating semicolons are optional.

In a large Teradata system configuration, use discretion when selecting the All, AMPs, and Online AMPs options, because of the potentially large numbers of output lines that may be produced.

Instead of the All, AMPs, and Online AMPs options, consider Processor level options or the Offline AMPs and Offline PEs options when the configuration is large, to avoid having to deal with large numbers of Query Configuration output lines consuming valuable space and time resources.

### ALL

To produce a display of all the components in the Teradata system and their current status, use the ALL option. The following is an example of an ALL display.

```
Enter display option,  QUIT or END to terminate.
> all
all
```

DBS Configuration Status Report: 16/12/23 20:21:08

| Vproc<br>Number | Node<br>ID | Config<br>Type | Config<br>Status | Global<br>Map<br>Cluster/<br>Host No. | Vproc<br>Number | Node<br>ID | Config<br>Type | Config<br>Status | Global<br>Map<br>Cluster/<br>Host No. |
|-----------------|------------|----------------|------------------|---------------------------------------|-----------------|------------|----------------|------------------|---------------------------------------|
| 0               | 1-01       | AMP            | Online           | 0                                     | 1               | 1-01       | AMP            | Online           | 1                                     |



```

      2      1-01 AMP Online      0      3      1-01 AMP Online      1
30718      1-01 PE Online      1      30719      1-01 PE Online      1

```

#### MAP Configuration

```

Slot 0: TD_GlobalMap
MAP State: Online

```

```

Slot 1: TD_DataDictionaryMap
MAP State: Online

```

```

Slot 2: TD_Map1
MAP State: Online

```

| AMP<br>Number | Slot 0<br>Cluster<br>No | Slot 1<br>Cluster<br>No | Slot 2<br>Cluster<br>No |
|---------------|-------------------------|-------------------------|-------------------------|
| 0             | 0                       | 0                       | 0                       |
| 1             | 1                       | 1                       | 1                       |
| 2             | 0                       | 0                       | 0                       |
| 3             | 1                       | 1                       | 1                       |

The information in the ALL display is ordered by vproc number. The display shows the following for each vproc:

- The identifier (Node ID) of the Node associated with each vproc (virtual processor).
- The type (AMP or PE) of each vproc.
- The status (online or offline) of each vproc. A status of Online means that the associated Node is available and online to the operating system.
- The cluster (for an AMP) or host number (for a PE) associated with each vproc.

In addition, the output shows configuration information for the maps used for data distribution and indicates the fallback clustering defined by those maps.

## Processors

To obtain configuration information for all processors in the Teradata system, use the Processors option. This option produces a display identical to that produced by the [ALL](#) option.

### Online Processors or Offline Processors

Configuration information for all online or all offline processors that is similar to that of all processors is also available.

To obtain configuration information for online processors, use the Online Processors option. This option includes the same information as the all processors display option with the exception of the Status column. The following is an example of an Online Processors display.

```

Enter display option, QUIT or END to terminate.
online processors
DBS Configuration Status Report: 00/06/13 18:35:56

```

| Global Map   |         |             |                   | Global Map   |         |             |                   |
|--------------|---------|-------------|-------------------|--------------|---------|-------------|-------------------|
| Vproc Number | Node ID | Config Type | Cluster/ Host No. | Vproc Number | Node ID | Config Type | Cluster/ Host No. |
| 0            | 1-01    | AMP         | 0                 | 1            | 1-01    | AMP         | 0                 |
| 2            | 1-01    | AMP         | 1                 | 3            | 1-01    | AMP         | 1                 |
| 4            | 1-01    | AMP         | 2                 | 5            | 1-01    | AMP         | 2                 |
| 6            | 1-01    | AMP         | 3                 | 7            | 1-01    | AMP         | 3                 |
| 8            | 1-01    | AMP         | 4                 | 9            | 1-01    | AMP         | 4                 |
| 10           | 1-01    | AMP         | 5                 | 11           | 1-01    | AMP         | 5                 |
| 12           | 1-01    | AMP         | 6                 | 13           | 1-01    | AMP         | 6                 |
| 14           | 1-01    | AMP         | 7                 | 15           | 1-01    | AMP         | 7                 |
| 16           | 1-01    | AMP         | 8                 | 17           | 1-01    | AMP         | 8                 |
| 18           | 1-01    | AMP         | 9                 | 19           | 1-01    | AMP         | 9                 |
| 16380        | 1-01    | PE          | 1                 | 16381        | 1-01    | PE          | 1                 |
| 16382        | 1-01    | PE          | 1                 | 16383        | 1-01    | PE          | 1                 |

To obtain configuration information for offline processors, use the Offline Processors option. The information displayed is similar to that for all processors, but the Status column does not appear, since only offline processors are displayed.

## AMPs

To obtain configuration information for all AMPs in the Teradata system, use the AMPs option. It produces a display containing AMP numbers, node identifiers, status (online or offline), and cluster numbers. The following is an example of an AMPs configuration display.

Enter display option, QUIT or END to terminate.  
amps

DBS Configuration Status Report: 00/06/13 18:38:20

| Global Map   |         |             |               |                   | Global Map   |         |             |               |                   |
|--------------|---------|-------------|---------------|-------------------|--------------|---------|-------------|---------------|-------------------|
| Vproc Number | Node ID | Config Type | Config Status | Cluster/ Host No. | Vproc Number | Node ID | Config Type | Config Status | Cluster/ Host No. |
| 0            | 1-01    | AMP         | Online        | 0                 | 1            | 1-01    | AMP         | Online        | 0                 |
| 2            | 1-01    | AMP         | Online        | 1                 | 3            | 1-01    | AMP         | Online        | 1                 |
| 4            | 1-01    | AMP         | Online        | 2                 | 5            | 1-01    | AMP         | Online        | 2                 |
| 6            | 1-01    | AMP         | Online        | 3                 | 7            | 1-01    | AMP         | Online        | 3                 |
| 8            | 1-01    | AMP         | Online        | 4                 | 9            | 1-01    | AMP         | Online        | 4                 |
| 10           | 1-01    | AMP         | Online        | 5                 | 11           | 1-01    | AMP         | Online        | 5                 |

|    |      |     |        |   |    |      |     |        |   |
|----|------|-----|--------|---|----|------|-----|--------|---|
| 12 | 1-01 | AMP | Online | 6 | 13 | 1-01 | AMP | Online | 6 |
| 14 | 1-01 | AMP | Online | 7 | 15 | 1-01 | AMP | Online | 7 |
| 16 | 1-01 | AMP | Online | 8 | 17 | 1-01 | AMP | Online | 8 |
| 18 | 1-01 | AMP | Online | 9 | 19 | 1-01 | AMP | Online | 9 |

### Online AMPS or Offline AMPS

The same type of configuration information previously described for all AMPs is also available for all online or all offline AMPs.

To obtain configuration information for online AMPs, use the Online AMPs option. The information displayed is similar to that for all AMPs, but the Status column does not appear, since only online AMPs are displayed.

To obtain configuration information for offline AMPs, use the Offline AMPs option. The information displayed is similar to that for all AMPs, but the Status column does not appear, since only offline AMPs are displayed.

### PEs

To obtain configuration information for all PEs in the Teradata system, use the PEs option. It produces a display containing PE numbers, node identifiers, status (online or offline), and host numbers. The following is an example of a PEs configuration display.

Enter display option, QUIT or END to terminate.

pes

DBS Configuration Status Report: 00/06/13 18:44:20

| Global Map   |         |             |               |                   | Global Map   |         |             |               |                   |
|--------------|---------|-------------|---------------|-------------------|--------------|---------|-------------|---------------|-------------------|
| Vproc Number | Node ID | Config Type | Config Status | Cluster/ Host No. | Vproc Number | Node ID | Config Type | Config Status | Cluster/ Host No. |
| 16380        | 1-01    | PE          | Online        | 1                 | 16381        | 1-01    | PE          | Online        | 1                 |
| 16382        | 1-01    | PE          | Online        | 1                 | 16383        | 1-01    | PE          | Online        | 1                 |

### Online PEs and Offline PEs

The same type of configuration information available as described above for all PEs is also available for all online or all offline PEs.

To obtain configuration information for online PEs, use the Online PEs option. The information displayed is similar to that for all PEs, but the status is Online for each PE listed.

To obtain configuration information for offline PEs, use the Offline PEs option. The information displayed is similar to that for all PEs, but the status is Offline for each PE listed.

## Query Session (qrysessn)

---

### Note:

The utility described is not available for as-a-service customers.

---

The Query Session utility, qrysessn, provides information about active Advanced SQL Engine sessions. It allows you to monitor the state of all or selected database sessions on all or selected logical host IDs attached to the Advanced SQL Engine. Query Session is also known as Session States.

This discussion of the information Query Session reports includes several sample Query Session displays.

The Query Session displays for sessions involved in FastLoad, MultiLoad (MLOAD), and FastExport operations are different from other session displays. These displays are described at the end of the discussion of Query Session.

### Runs From

- Database Window or comparable interface to the Teradata console subsystem, such as cnstern
- Teradata Viewpoint Remote Console portlet
- Host Utility Console
- Linux command line

For general information on starting the utilities from different interfaces, see [Starting the Utilities](#). For information on Viewpoint, see *Teradata® Viewpoint User Guide*, B035-2206.

### Query Session States

Query Session provides information on the following possible states of a session.

| The Session State... | Indicates that...                                                                                                                    |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| ABORTING             | the session is aborting its latest request.                                                                                          |
| ACTIVE               | the session has sent steps to the dispatcher and possibly to one or more AMP vprocs.                                                 |
| BLOCKED              | an active session is waiting for a database lock to be released.                                                                     |
| DELAYED              | the session is being delayed because query limit, imposed by a Teradata Active System Management (TASM) throttle rule, has been met. |
| IDLE                 | the Teradata system recognizes a session, but no processing is taking place.                                                         |
| INDOUBT              | a two-phase commit session is in doubt.                                                                                              |
| INDOUBT PARSING      | a two-phase commit session is in doubt and is parsing a vote or commit request.                                                      |

| The Session State...       | Indicates that...                                                                                                                                                     |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PARSING                    | the session is active in the Teradata SQL parser phase, before the steps are dispatched to the AMP vprocs.                                                            |
| QTDELAYED                  | the session is waiting for rows to be inserted into a queue table.                                                                                                    |
| QUIESCED ABORT             | the session is blocked from executing further requests because transactions associated with this session are being aborted.                                           |
| QUIESCED ABORT WITH LOGOFF | the session is quiesced because the transactions or requests associated with this session are being aborted, after which this session will be logged off the machine. |
| QUIESCED INDOUBT           | the session is blocked from exercising further requests because the outstanding transaction/request will be terminated.                                               |
| RESPONSE                   | a response to a session request is in process.                                                                                                                        |
| SESDELAYED                 | the session is being delayed because a utility limit, imposed by a TASM throttle rule, has been met.                                                                  |

## Parent and Child Sessions

When a connected session is part of either a FastLoad, MultiLoad, or FastExport operation, that session establishes subordinate sessions to accomplish the task more quickly. The originating session is then called a *Parent* session; the subordinate sessions are called *Child* sessions that belong to that Parent.

A Child session is not recognized as such, and its activity is not reported until it has transmitted at least one request to the AMPs in the current transaction.

Information about the activity of Child sessions is available to Query Session when the Parent session is in the Active state or when one of several inactive states described later.

In FastLoad operations, Child sessions exist and are reported only while the Parent session is in the loading phase. In MultiLoad operations, Child sessions exist and are reported only while the Parent session is in the acquisition phase. Query Session reports the status of Child sessions when you request detail information.

## Query Session Displays

The following sections describe the Query Session displays that provide information on what you can input. General Help displays are described first, followed by displays for sessions involved in FastLoad, FastExport, and MultiLoad operations.

After you have started Query Session, it prompts for the following:

```
Please Enter A Logical Host ID (? For Help)
```

You can type any of the following responses when you are prompted for the logical host ID.

| IF you type...            | THEN...                                                                                                 |
|---------------------------|---------------------------------------------------------------------------------------------------------|
| an integer                | Query Session provides information only for the sessions associated with the specified logical host ID. |
| an asterisk (*)           | Query Session provides information on sessions for every logical host.                                  |
| a question mark (?)       | you will get Help for this entry.                                                                       |
| a carriage return (Enter) | you will exit from the Query Session utility.                                                           |

If you type an integer or an asterisk, the following prompt displays:

```
Please Enter Session Ids (? For Help):
```

You can input any of the following when you are prompted for the Session ID.

| IF you type...                                                                                                                                                                                      | THEN you see...                                                                                                                                                                                                                                                                                                                   |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| an integer to indicate a specific session ID<br><br><b>Note:</b><br>You can input up to five session IDs in integer format. You can type them on one line separated by spaces or on multiple lines. | the following prompt:<br><pre>Is detail information needed if the session is involved in HUT/ FastLoad/MLoad/Export? y=yes, n=no</pre> <ul style="list-style-type: none"> <li>To view the Child session, type y.</li> <li>If the session is not part of a FastLoad, MultiLoad, or FastExport operation, type n.</li> </ul>        |
| an asterisk (*)                                                                                                                                                                                     | the following prompt:<br><pre>Is detail information needed if the session is involved in HUT/ FastLoad/MLoad/Export? y=yes, n=no</pre> <ul style="list-style-type: none"> <li>To view the Child session states, type y.</li> <li>If the session is not part of a FastLoad, MultiLoad, or FastExport operation, type n.</li> </ul> |
| a question mark                                                                                                                                                                                     | help concerning input information.                                                                                                                                                                                                                                                                                                |

**Note:**

If an unsuccessful attempt was made to obtain temporary table information for the given session, the following query session message can appear:

WARNING: Session may contain incomplete temporary table information due to deadlock!

If temporary table information is important to you, then you should retry querying the session at a later time.

# Session State Display

The session state display consists of session identifier information and session state details. Displays for sessions involved in FastLoad, MultiLoad, or FastExport operations provide additional information described later.

If a session is idle, only the session identifier information is displayed, as shown below.

```
Host Session PE DBC User ID
-----
110 1006    11 DBC
```

The columns for the Session Identifier display the following information.

| The column named... | Contains the...                                       |
|---------------------|-------------------------------------------------------|
| Host                | logical host identifier.                              |
| Session             | session identifier.                                   |
| PE                  | PE number of the PE to which the session is assigned. |
| DBC User ID         | user assigned to the session.                         |

If a session is active, and not part of a FastLoad, MultiLoad, or FastExport, the following session state detail information displays.

```
Session State Query Results : 00/06/15 14:14:13

Host Session PE DBC User ID
-----
110 1006    11 DBC

State Details : ACTIVE

Statements Dispatched Time      CPU Usage Accesses
```

|       |            |   |    |
|-------|------------|---|----|
| ----- |            |   |    |
| 1     | 1 14:22:09 | 2 | 22 |

**Session State Details During Stored Procedure Execution**

If a SQL or external stored procedure is being executed, the following session state information is displayed.

```
Session State Query Results : 00/06/15 13:14:13

Host Session PE DBC User ID
-----
110 1006      11 DBC

State Details : ACTIVE (Stored Procedure is executing)
```

The State Details correspond to the latest SQL request that the stored procedure executed.



## State Information Displays

The following topics provide session information for each possible state.

## ABORTING

The ABORTING state indicates that the session is aborting its latest request.

The ABORTING state display is shown below.

State Details : ABORTING

| Statements | Code  | Time     | CPU Usage | Accesses |
|------------|-------|----------|-----------|----------|
| -----      | ----- | -----    | -----     | -----    |
| 1          | 1043  | 14:22:09 | 2         | 22       |

The columns on the ABORTING state display provide the following information.

| The column named... | Contains the...                                                                                 |
|---------------------|-------------------------------------------------------------------------------------------------|
| Statements          | statements, up to the number displayed, that are aborting.                                      |
| Code                | error that caused the abort.                                                                    |
| Time                | time that the abort step was sent to the AMP.                                                   |
| CPU Usage           | accumulated time in thousandths of a second that all AMPs spent processing the current request. |
| Accesses            | total number of segment access calls executed on all AMPs for the session request.              |

## ACTIVE

The ACTIVE state indicates that the session has sent steps to the dispatcher and possibly to one or more AMP vprocs.

The ACTIVE state display for a session not part of a FastLoad, MultiLoad, or FastExport operation is shown below.

State Details : ACTIVE

| Statements | Dispatched | Time     | CPU Usage | Accesses |
|------------|------------|----------|-----------|----------|
| -----      | -----      | -----    | -----     | -----    |
| 1          | 1          | 14:22:09 | 2         | 22       |

The columns on the ACTIVE state display provide the following information.

| The column named... | Contains the...                                            |
|---------------------|------------------------------------------------------------|
| Statements          | total number of statements in the current session request. |
| Dispatched          | highest statement number dispatched to the AMPs.           |

| The column named... | Contains the...                                                                                 |
|---------------------|-------------------------------------------------------------------------------------------------|
| Time                | time that the last step in the highest statement number was sent to the AMPs.                   |
| CPU Usage           | accumulated time in thousandths of a second that all AMPs spent processing the current request. |
| Accesses            | total number of segment access calls executed on all AMPs for the session request.              |

## BLOCKED

The BLOCKED state indicates that an active session is waiting for a database lock to be released.

The BLOCKED state display for a session is shown below.

State Details : BLOCKED

Resource

-----

X.T

Statement AMPs Mode AMP Vproc HUT

-----

1 1 READ 19 NO

The columns on the BLOCKED state display provide the following information.

| The column named... | Contains the...                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Resource            | database or table for which the lock is requested. <ul style="list-style-type: none"> <li>If the lock is requested for a database, the column displays the information: X.*<br/>where:<br/>X is the database name.</li> <li>If the lock is requested for a table or row, the column displays the information: X.T<br/>where:               <ul style="list-style-type: none"> <li>X is the database name</li> <li>T is the table name.</li> </ul> </li> </ul> |
| Statement           | highest statement number for which steps have been dispatched to the AMPs.                                                                                                                                                                                                                                                                                                                                                                                    |
| AMPs                | number of AMPs for which the session is waiting to acquire a lock.                                                                                                                                                                                                                                                                                                                                                                                            |
| Mode                | type of lock requested by the session.                                                                                                                                                                                                                                                                                                                                                                                                                        |
| AMP Vproc           | number of the AMP with the blocked session.                                                                                                                                                                                                                                                                                                                                                                                                                   |

| The column named... | Contains the...                     |
|---------------------|-------------------------------------|
| HUT                 | host utility lock (if encountered). |

## DELAYED

The DELAYED state indicates that an SQL request is delayed because a query limit has been exceeded. When the limit is no longer exceeded, the delayed requests are processed.

Teradata Viewpoint Workload Designer portlet allows you to define and manage throttle rules, which restrict the number of requests that can be simultaneously executed against Vantage. These rules delay or reject database requests based on specified conditions, such as the number of concurrent queries or the duration of queries. A delayed request is allowed to proceed when the rule's limit is no longer exceeded.

The DELAYED state display for a session is shown below.

State Details : DELAYED

## IDLE

The IDLE state indicates that the Teradata system recognizes a session, but no processing is taking place.

The IDLE state display for a session is shown below.

State Details : IDLE

## INDOUBT

The INDOUBT state indicates that a two-phase commit session is in doubt. This state continues until an abort or commit is received from the host application.

The INDOUBT state display for a session is shown below.

State Details : INDOUBT

## INDOUBT PARSING

The INDOUBT PARSING state indicates that a two-phase commit session is in doubt and is parsing a vote or commit request.

The INDOUBT PARSING state display for a session is shown below.

State Details : INDOUBT PARSING

## PARSING

The PARSING state indicates that the session is active in the Teradata SQL parser phase, before the steps are dispatched to the AMP vprocs. The PARSING state displays no column information and only the word PARSING.

The PARSING state for a session display is shown below.

```
State Details : PARSING
```

## QTDELAYED

The QTDELAYED state indicates that a session is waiting for rows to be inserted into a queue table. The QTDELAYED state is limited to 20% of the total possible sessions, which is 120 x the number of PEs. If the 20% limit is exceeded, then error message 3128 appears:

```
3128 Transaction aborted due to exceeding the maximum consume statement limit.
```

You must decrease the number of sessions consuming queue table rows. For detailed information on error message 3128, refer to *Teradata Vantage™ - Database Messages*, B035-1096.

The QTDELAYED state display for a session is shown below.

```
State Details : QTDELAYED
```

## QUIESCED ABORT

The QUIESCED ABORT state indicates that the session is blocked from executing further requests because transactions associated with this session are being aborted. The outstanding transaction or request will be aborted.

The QUIESCED ABORT state for a session display is shown below.

```
State Details : QUIESCED ABORT
```

## QUIESCED ABORT WITH LOGOFF

The QUIESCED ABORT WITH LOGOFF state indicates that the session is quiesced because the transactions or requests associated with this session are being aborted. The session is logged off.

The QUIESCED ABORT WITH LOGOFF state for a session display is shown below.

```
State Details : QUIESCED ABORT WITH LOGOFF
```

## QUIESCED INDOUBT

The QUIESCED INDOUBT state indicates that the session is blocked from exercising further requests because the outstanding transaction/request will be terminated by the resolver base module.

The QUIESCED INDOUBT state for a session display is shown below.

State Details : QUIESCED INDOUBT

## RESPONSE

The RESPONSE state indicates a response to a session request is in process.

The RESPONSE state for a session display is shown below.

State Details : RESPONSE

Statements

-----

1

This column for the RESPONSE display contains the following information.

| The column named... | Contains the...                                                                                                                   |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| Statements          | highest statement number for which a response has been returned to the host or the SQL or external stored procedure in execution. |

## SESDelayed

The SESDELAYED state indicates that a session is delayed because a utility limit has been exceeded. The utilities that can be limited are FastLoad, MultiLoad, and FastExport. When the limit is no longer exceeded, the delayed requests are processed.

Teradata Viewpoint Workload Designer portlet allows you to define and manage throttle rules, which restrict the type and number of utilities that are allowed to run concurrently.

The SESDELAYED state display for a session is shown below.

State Details : SESDELAYED

## FastLoad Sessions State Displays

The following sections describe displays that are part of FastLoad operations:

- Active Parent Session in a Loading Phase
- Active Parent Session in a Nonloading Phase
- Inactive Parent Session in a Loading Phase
- Inactive Parent Session in a Nonloading Phase
- Child Sessions

If you request detail information, Query Session reports child sessions as well. These displays are identified after the phrase State Details by two lines describing the current phase.

Information for a FastLoad session in the loading phase is different from information for a non-loading phase of FastLoad.

**Active Parent Session in a Loading Phase Display**

The following example shows a display of an active parent session for a FastLoad operation in the loading phase.

|                                                                    |            |          |           |          |           |
|--------------------------------------------------------------------|------------|----------|-----------|----------|-----------|
| State Details : Active Parent Session involved in FASTLOAD utility |            |          |           |          |           |
| FastLoad Phase : Loading                                           |            |          |           |          |           |
| Statements                                                         | Dispatched | Time     | CPU Usage | Accesses | Row Count |
| -----                                                              | -----      | -----    | -----     | -----    | -----     |
| 1                                                                  | 1          | 14:22:09 | 15,110    | 6,462    | 2,060     |

The display for an active parent session in the loading phase of a FastLoad operation contains the following information.

| The column named... | Contains the...                                                                                 |
|---------------------|-------------------------------------------------------------------------------------------------|
| Statements          | total number of statements in the current session request.                                      |
| Dispatched          | highest statement number dispatched to the AMPs.                                                |
| Time                | time that the last step for the highest statement number was sent to the AMPs.                  |
| CPU Usage           | accumulated time in thousandths of a second that all AMPs spent processing the current request. |
| Accesses            | total number of segment access calls executed on all AMPs for the session request.              |
| Row Count           | total number of rows loaded by the FastLoad utility.                                            |

**Active Parent Session in a Nonloading Phase Display**

The following shows an example display of an active parent session of a FastLoad operation in any phase other than the loading phase.

State Details : Active Parent Session involved in FASTLOAD utility  
 FastLoad Phase : LoadPending

| Statements | Dispatched | Time     | CPU Usage | Accesses |
|------------|------------|----------|-----------|----------|
| -----      | -----      | -----    | -----     | -----    |
| 1          | 1          | 14:22:09 | 2         | 22       |

The display for an active parent session in any phase other than the loading phase of a FastLoad operation contains the following information.

| The column named... | Contains the...                                                                                 |
|---------------------|-------------------------------------------------------------------------------------------------|
| Statements          | total number of statements in the current session request.                                      |
| Dispatched          | highest statement number dispatched to the AMPs.                                                |
| Time                | time that the last step for the highest statement number was sent to the AMPs.                  |
| CPU Usage           | accumulated time in thousandths of a second that all AMPs spent processing the current request. |
| Accesses            | total number of segment access calls executed on all AMPs for the session request.              |

### Inactive Parent Session in a Loading Phase Display

The following shows an example display of an inactive parent session of a FastLoad operation in the loading phase.

State Details : Inactive Parent Session involved in FASTLOAD utility  
 FastLoad Phase : Loading

| CPU Usage | Accesses | Row Count |
|-----------|----------|-----------|
| -----     | -----    | -----     |
| 234       | 567      | 456,321   |

The display for an inactive parent session in the loading phase of a FastLoad operation contains the following information.

| The column named... | Contains the...                                                                                 |
|---------------------|-------------------------------------------------------------------------------------------------|
| CPU Usage           | accumulated time in thousandths of a second that all AMPs spent processing the current request. |
| Accesses            | total number of segment access calls executed on all AMPs for the session request.              |
| Row Count           | total number of rows loaded by the FastLoad utility.                                            |



## Inactive Parent Session in a Nonloading Phase Display

The following shows an example display of an inactive parent session involved in a FastLoad operation in any phase other than the loading phase.

```
State Details : Inactive Parent Session involved in FASTLOAD utility.
FastLoad Phase : Load Pending
```

```
CPU Usage Accesses
```

```
-----
      65      870
```

The inactive parent session display for sessions involved in any phase other than the loading phase contains the following information.

| The column named... | Contains the...                                                                                 |
|---------------------|-------------------------------------------------------------------------------------------------|
| CPU Usage           | accumulated time in thousandths of a second that all AMPs spent processing the current request. |
| Accesses            | total number of segment access calls executed on all AMPs for the session request.              |

## Child Sessions Display

A FastLoad operation only involves child sessions when in the loading phase.

If the parent of the queried child session is not in the loading phase, the child session information is not available. The following shows an example display for child sessions involved in a FastLoad operation when the long form of the display is requested in response to the Detail Information Needed prompt.

```
Session # Request # State   TimeStamp Byte Count
-----
      1055      1632 Inactive 15:57:10      5,286
      1056      1635 Active   15:57:23        372
```

The display for Child sessions involved in FastLoad operations contain the following information.

| The column named... | Contains the...                                                                                                                              |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| Sessions#           | session identifier.                                                                                                                          |
| Request #           | number of the request.                                                                                                                       |
| State               | state of the session, indicating whether the session is active or inactive.                                                                  |
| TimeStamp           | time that is updated whenever a request is received from the host, a request is reinitiated to another AMP, or a response is sent to a host. |

| The column named... | Contains the...                             |
|---------------------|---------------------------------------------|
| Row Count           | total number of rows loaded by the session. |

## MultiLoad Sessions State Displays

Several displays are defined for the sessions that are part of MultiLoad operations.

The following sections describe those displays:

- Preliminary Phase Sessions
- Application Phase Session for Apply Task
- Application Phase Session for Delete Task
- Active Parent Session in an Acquisition Phase
- Inactive Parent Session in an Acquisition Phase
- Child Session in an Acquisition Phase

In addition to the phase State Detail line, a line describing the current phase is displayed.

If the session is in the Preliminary or Application phase, the current task type (Delete or Apply) is displayed. The information displayed is different for each phase: Preliminary, Application, and Acquisition. In the Application phase, each of the two task types is displayed.

When the action involves more than one AMP, row count summaries are meaningless and are not reported.

### Preliminary Phase Session Display

The following shows an example display session, that is part of a MultiLoad operation, in the Preliminary phase.

|                                                     |            |           |          |          |  |
|-----------------------------------------------------|------------|-----------|----------|----------|--|
| State Details : Session involved in MLOAD utility   |            |           |          |          |  |
| MLoad Phase : Preliminary - Received all DML Steps. |            |           |          |          |  |
| Task Running : Apply Task                           |            |           |          |          |  |
|                                                     |            |           |          |          |  |
| Statements Dispatched                               | Time       | CPU Usage | Accesses | DMLCount |  |
| -----                                               | -----      | -----     | -----    | -----    |  |
| 10                                                  | 2 12:09:09 | 7         | 15       | 5        |  |

The possible tasks include Apply Task and Delete Task. These are the Subphases:

- No MLOAD step has been received.
- Receiving MLOAD step.
- Received all MLOAD steps.
- Received all DML (Data Manipulation Language) steps.

The state display for the preliminary phase of a MultiLoad operation contains the following information.

| The column named... | Contains the...                                                                                 |
|---------------------|-------------------------------------------------------------------------------------------------|
| Statements          | total number of statements in the current session request.                                      |
| Dispatched          | highest statement number dispatched to the AMPs.                                                |
| Time                | time that the last step for the highest statement number was sent to the AMPs.                  |
| CPU Usage           | accumulated time in thousandths of a second that all AMPs spent processing the current request. |
| Accesses            | total number of segment access calls executed on all AMPs for the session request.              |
| DML Count           | number of DML steps received if the current phase is Received all DML Steps.                    |

### Application Phase Session for Apply Task Display

The Query Session display for each table during an Apply Task includes the name of the database and table, the current action, the number of workrows applied, and the total number of workrows. The following shows an example display session involved in a MultiLoad operation during an Apply Task.

State Details : Session involved in MLOAD utility

MLoad Phase : Application.

Task Running : Apply Task

| Statements | Dispatched | Time     | CPU Usage | Accesses |
|------------|------------|----------|-----------|----------|
| -----      | -----      | -----    | -----     | -----    |
| 1          | 1          | 11:09:37 | 811751    | 349,637  |

|                                  |       |            |
|----------------------------------|-------|------------|
| DataBase.Table                   | =     |            |
| SPOOL_RES.WT_TDEM_PAINT_MED      | ===== | Action     |
| Process Data and Secondary index |       |            |
| # of WorkRows applied            | =     | 1,210,838  |
| Total # of WorkRows              | =     | 51,639,908 |
| # of NUSI change rows applied    | =     | 0          |
| Total # of NUSI change rows      | =     | 0          |

The display for active parent sessions involved in the Apply Task of a MultiLoad operation provides the following information.

| The column named... | Contains the...                                                                |
|---------------------|--------------------------------------------------------------------------------|
| Statements          | total number of statements in the current session request.                     |
| Dispatched          | highest statement number dispatched to the AMPs.                               |
| Time                | time that the last step for the highest statement number was sent to the AMPs. |

| The column named... | Contains the...                                                                                 |
|---------------------|-------------------------------------------------------------------------------------------------|
| CPU Usage           | accumulated time in thousandths of a second that all AMPs spent processing the current request. |
| Accesses            | total number of segment access calls executed on all AMPs for the session request.              |

In addition, the display provides the following information.

| Field                         | Description                                                            |
|-------------------------------|------------------------------------------------------------------------|
| DataBase.Table                | Identifies the table on which the MultiLoad operation is running.      |
| Action                        | Describes the action being performed in the MultiLoad session.         |
| # of WorkRows applied         | Displays the number of work rows processed in the apply phase.         |
| Total # of WorkRows           | Displays the total number of rows processed.                           |
| # of NUSI change rows applied | Number of nonunique secondary index rows processed in the apply phase. |
| Total # of NUSI change rows   | Total number of nonunique secondary index rows processed.              |

**Application Phase Session for Delete Task Display**

The following shows an example of the Query Session display for each table during a Delete Task, which includes the name of the database and table, the current action, the number of rows scanned, and the number of rows deleted.

```
State Details : Session involved in MLOAD utility
MLoad Phase : Application.
Task Running : Delete Task

Statements Dispatched Time      CPU Usage Accesses
-----
              3          3 11:24:32      844679  310,987

DataBase.Table      = SPOOL_RES.WT_TDEM_PAIMT_MED
Action              = Process Data
# of rows scanned   =  22,357
# of rows deleted   = 245,349
```

**Note:**

The values shown include both the primary and the fallback count.

The Delete Task display for a MultiLoad operation contains the same information as described for the Apply Task display except that it reports the number of rows processed and deleted rather than the rows processed by the apply phase.

### Active Parent Session in an Acquisition Phase Display

The following shows an example display of an active parent session which is part of a MultiLoad operation.

```
State Details : Active Parent Session involved in MLOAD utility
MLoad Phase : Acquisition - Data Loading is in progress.
```

| Statements | Dispatched | Time     | CPU Usage | Accesses | Row Count |
|------------|------------|----------|-----------|----------|-----------|
| -----      | -----      | -----    | -----     | -----    | -----     |
| 1          | 1          | 09:23:45 | 8         | 166      | 9,854     |

If you request detail information, Query Session reports child sessions as well. These are the descriptions of the phase:

- Data Loading is in progress.
- Data Loading is complete.

The display for active parent sessions involved in MultiLoad operations provides the following information.

| The column named... | Contains the...                                                                                 |
|---------------------|-------------------------------------------------------------------------------------------------|
| Statements          | total number of statements in the current session request.                                      |
| Dispatched          | highest statement number dispatched to the AMPs.                                                |
| Time                | time that the last step for the highest statement number was sent to the AMPs.                  |
| CPU Usage           | accumulated time in thousandths of a second that all AMPs spent processing the current request. |
| Accesses            | total number of segment access calls executed on all AMPs for the session request.              |
| Row Count           | total number of rows processed by the MultiLoad task.                                           |

### Inactive Parent Session in an Acquisition Phase Display

The following shows an example display of an inactive parent session that is part of a MultiLoad operation.

```
State Details: InActive Parent Session involved in MLOAD Utility
MLoad Phase : Acquisition - Data Loading is complete.
```

| CPU Usage | Accesses | Row Count |
|-----------|----------|-----------|
| -----     | -----    | -----     |
| 130       | 654      | 45,673    |

These are the possible descriptions of the phase:

- Data Loading is in progress.
- Data Loading is complete.

The display for inactive parent sessions involved in a MultiLoad operation contains the following information.

| The column named... | Contains the...                                                                                 |
|---------------------|-------------------------------------------------------------------------------------------------|
| CPU Usage           | accumulated time in thousandths of a second that all AMPs spent processing the current request. |
| Accesses            | total number of segment access calls executed on all AMPs for the session request.              |
| Row Count           | total number of rows processed by the MultiLoad task.                                           |

### Child Session in an Acquisition Phase Display

The following shows an example display for child sessions involved in a MultiLoad operation when the long form of the display is requested in response to the Detail Information Needed prompt.

State Details: CHILD session involved in MLOAD Acquisition Phase

| Session # | Request # | State    | TimeStamp | Row Count |
|-----------|-----------|----------|-----------|-----------|
| 1055      | 1632      | Inactive | 15:57:10  | 5,286     |
| 1056      | 1635      | Active   | 15:57:23  | 372       |

Child sessions involved in MultiLoad operations display these columns.

| The column named... | Contains the...                                                                                                                                |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| Session #           | session identifier.                                                                                                                            |
| Request #           | number of the request.                                                                                                                         |
| State               | state of the session, indicating whether the session is active or inactive.                                                                    |
| TimeStamp           | time that is updated whenever a request is received from the host, a request is reinitiated to another AMP, or a response is sent to the host. |
| Row Count           | total number of rows loaded by the session.                                                                                                    |

## FastExport Sessions State Displays

FastExport of data includes sessions under which Teradata SQL statements are executed, as well as sessions used to transfer response data to the host.

The Teradata SQL session assumes one of five sequential states:

- Executing the select
- Releasing resource locks
- Vertical data redistribution
- Horizontal data redistribution
- Inactive

Vertical data redistribution and horizontal data redistribution are processes executed to prepare the response data from a SELECT request for transfer back to the host. Vertical redistribution occurs only if the SELECT contains an ORDER BY clause. The idle state occurs before the select request is transmitted to the Teradata system, or while the FastExport sessions are transmitting data to the host.

FastExport sessions are children of the Teradata SQL session that executes the select request. A child session exists in one of two states. A child session in the inactive state means that either the select has not completed, or that the host utility has not yet issued a request for this child session in returning response data. A child session in the active state is currently transmitting response data.

Like FastLoad and MultiLoad, the FastExport utility executes bulk activities. The Teradata system limits the maximum number of load operations (that is, FastExport, FastLoad, or MultiLoad) to the number specified in DBS Control field MaxLoadTasks.

Examples of Teradata SQL sessions involved in FastExport activity and examples of the output for a FastExport session are shown in the following sections.

### Teradata SQL Session - Releasing Locks

The following shows an example display of a single FastExport Teradata SQL session in release locks state.

Session State Query Results : 00/06/14 18:35:54

| Host | Session | PE  | DBC | User | ID |
|------|---------|-----|-----|------|----|
| 114  | 1090    | 1-4 | DBC |      |    |

State Details : Active PARENT session involved in FastExport

FastExport Phase : Vertical redistribution.

| Statements | Dispatched | Time     | CPU Usage | Accesses |
|------------|------------|----------|-----------|----------|
| 1          | 1          | 18:35:54 | 6174      | 2,922    |

Detail information for CHILDREN sessions in FastExport Util.

| Session # | Request # | State    |
|-----------|-----------|----------|
| 1091      | 0         | Inactive |

|      |            |
|------|------------|
| 1092 | 0 Inactive |
| 1093 | 0 Inactive |
| 1094 | 0 Inactive |

**Teradata SQL Session - Vertical Redistribution**

The following shows an example display of the Teradata SQL session in the vertical redistribution state. The select request consists of a single SELECT statement.

```
Session State Query Results : 00/06/13 18:38:59

Host Session PE  DBC User ID
-----
114 1090      1-4 DBC

State Details : Active PARENT session involved in FastExport

FastExport Phase : Vertical redistribution.

Statements Dispatched Time      CPU Usage Accesses
-----
1          1 18:37:54      6603    3,314

Detail information for CHILDREN sessions in FastExport Util.

Session # Request # State
-----
1091      0 Inactive
1092      0 Inactive
1093      0 Inactive
1094      0 Inactive
```

The following example shows that the Teradata system continues to process the vertical redistribution phase.

```
Session State Query Results : 00/06/12 18:38:59

Host Session PE  DBC User ID
-----
114 1090      1-4 DBC

State Details : Active PARENT session involved in FastExport

FastExport Phase : Horizontal redistribution.
```



| Statements Dispatched | Time       | CPU Usage | Accesses |
|-----------------------|------------|-----------|----------|
| 1                     | 1 18:42:35 | 6603      | 3,314    |

Detail information for CHILDREN sessions in FastExport Util.

| Session # | Request # | State    |
|-----------|-----------|----------|
| 1091      | 0         | Inactive |
| 1092      | 0         | Inactive |
| 1093      | 0         | Inactive |
| 1094      | 0         | Inactive |

The CPU Usage and the Accesses counts have increased, indicating the Teradata system is in operation.

### Teradata SQL Session - Horizontal Redistribution

With the Teradata SQL session in the horizontal redistribution state, the display might take the form shown in the following example.

Session State Query Results : 00/06/09 18:43:41

| Host | Session | PE  | DBC | User | ID |
|------|---------|-----|-----|------|----|
| 114  | 1090    | 1-4 | DBC |      |    |

State Details : Active PARENT session involved in FastExport

FastExport Phase : Horizontal redistribution.

| Statements Dispatched | Time       | CPU Usage | Accesses |
|-----------------------|------------|-----------|----------|
| 1                     | 1 18:42:35 | 7893      | 5,305    |

Detail information for CHILDREN sessions in FastExport Util.

| Session # | Request # | State    |
|-----------|-----------|----------|
| 1091      | 0         | Inactive |
| 1092      | 0         | Inactive |
| 1093      | 0         | Inactive |
| 1094      | 0         | Inactive |

**FastExport Session - Inactive**

The following example shows that the FastExport session is inactive, waiting for the select to complete.

```
Session State Query Results : 00/06/08 18:39:28

Host Session PE  DBC User ID
----
114 1091      N/A DBC

State Details : Child Session involved in FastExport Utility

FastExport Phase : Returning data.

Request # State      Parent Session
-----
          0 Inactive          1069
```

**FastExport Session - Data Transmission**

The following example shows FastExport returning data. This child session has returned one data block to the host for the first SELECT statement, where the response data contains 69 blocks.

```
Session State Query Results : 00/06/07 18:49:09

Host Session PE  DBC User ID
----
114 1091      N/A DBC

State Details : Child Session involved in FastExport Utility

FastExport Phase : Returning data.

Request # State      Statement Blocks Returned Total Block
-----
    1000 Active           1              4          69
```

FastExport transmission continues. As shown in the following example, the session has returned four data blocks to the host. In this way, the user is able to monitor the progress of the data transmission phase.

```
Session State Query Results : 00/06/06 18:50:52

Host Session PE  DBC User ID
----
```

114 1091 1-5 DBC

State Details : Child Session involved in FastExport Utility

FastExport Phase : Returning data.

| Request # | State  | Statement | Blocks Returned | Total Block |
|-----------|--------|-----------|-----------------|-------------|
| -----     | -----  | -----     | -----           | -----       |
| 1000      | Active | 1         | 4               | 69          |

# Recovery Manager (rcvmanager)

---

**Note:**

The utility described is not available for as-a-service customers.

---

The Recovery Manager utility, rcvmanager, monitors and provides information about the progress of Advanced SQL Engine in recovery, including transaction rollbacks, because of an Advanced SQL Engine system crash or user abort.

The recovery process might include one or more of the following:

- Online Transaction recovery
- Down AMP recovery of changed data rows
- Down AMP recovery of Advanced SQL Engine system level changes

Also, rcvmanager allows you to do the following:

- Cancel the rollback of one or more specified host and session IDs
- Set the priority level of rollbacks in progress for a specified host and session ID
- Set priority levels to optimize Advanced SQL Engine system recovery

## Runs From

- Database Window or comparable interface to the Teradata console subsystem, such as cnstern.
- Teradata Viewpoint Remote Console portlet
- Host Utility Console
- Linux command line

For general information on starting the utilities from different interfaces, see [Starting the Utilities](#). For information on Viewpoint, see *Teradata® Viewpoint User Guide*, B035-2206.

## Prerequisites

You can run rcvmanager only while the database is in one of the following states:

- Logon
- Logon/quiet
- Logoff
- Logoff/quiet
- Startup (if the Teradata system has completed voting for transaction recovery)

If you attempt to start rcvmanager while the Teradata system is not in one of the allowable states, the utility displays the following error information and terminates:

RCVMANAGER can only be run when the system is in the “LOGON” or “LOGOFF” state or if the system is in the “STARTUP” state and has completed voting for transaction recovery. The current system is not in one of these allowable states thus preventing RCVMANAGER from operating.  
RCVMANAGER terminated.

## Assigning Priority Levels

You can assign priority levels to Teradata system recovery or table rebuild by using the PRIORITY command feature of rcvmanager. Control over job priority is assigned by selecting one of three priority levels:

- HIGH
- MEDIUM
- LOW

For information about these priority levels, see [REBUILD/RECOVERY PRIORITY](#).

Teradata system recovery and disk rebuild are primarily I/O intensive tasks. The major portion of the time taken by a given task is involved in setting up for or waiting for completions of a disk or a BYNET message traffic operation. Very little manipulation or computation is required on the data once it is available.

If the competing work load of the Teradata system can be reasonably characterized, then you can gauge the impact of the priority changes to recovery and rebuild operations. In general, use the following guidelines when assigning priorities:

| Guideline                        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| No work load competition         | If there is no competition for resources as in a COLDWAIT restart for recovery, the priority setting of the recovery and rebuild jobs will have no practical effect.                                                                                                                                                                                                                                                                                                                                                                                   |
| Compute-intensive work load      | If the online Teradata system work load is heavily compute intensive, raising the priority of the I/O intensive recovery operations can dramatically improve the recovery (including rebuild).<br>The high recovery will have a relatively minor impact on the online Teradata system operations. However, it will provide a better resource utilization and result in better Teradata system throughput. Similarly, a low priority of recovery in this work load will dramatically slow down recovery with only a moderate gain in online throughput. |
| Moderate or heavy disk work load | If a moderate or heavy amount of disk and/or BYNET usage occurs by the online system, then recovery will show moderate throughput changes by controlling the priority setting but with a larger impact against the Teradata system throughput. Memory contention becomes a major component of operation in these cases.                                                                                                                                                                                                                                |
| I/O saturation                   | As the I/O utilization approaches saturation, there are fewer opportunities to improve throughput or execution time of either the online Teradata system or the recovery job. In this case, we are competing for the same resource and that resource is not amenable to manipulation by control over the scheduling priority.                                                                                                                                                                                                                          |

## Online Transaction Recovery

Online transaction recovery is automatically invoked by a Teradata system restart and includes the following processes:

- Rolling back transactions that were not completed at the time of the crash or restart
- Completing transactions that were committed

Changes made to underlying data by transactions are recorded in the Transient Journal (TJ). AMPs keep track of transactions in progress using the TJ which is stored on each AMP's disk.

| IF...                                                                                                      | THEN...                                                                                                              |
|------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|
| all the modification operations successfully complete on all the AMPs working on behalf of the transaction | that transaction is successful and there is no further concern for recovery of that transaction.                     |
| a transaction does not complete due to a Teradata system crash or forced restart                           | those transactions are backed out as part of the Teradata system recovery process when the Teradata system restarts. |

## Transaction Recovery Sequence

The general sequence of a Teradata system recovery is as follows:

1. The status of each transaction on each online AMP is determined by the Teradata system.  
All of the online AMPs at the time of the restart work together to determine which transactions were complete, and which ones were not completed. Completed transactions are ignored and incomplete ones are backed out. The process of rolling back incomplete transactions might take some time.  
Write and exclusive locks are set against all data modified by incomplete transaction.
2. All locks needed for the recovery are set, and the Teradata system begins rollback or completion of the transactions.
3. The Teradata system accepts work and is operational.

---

### Note:

If new transactions reference the inconsistent data of to-be-rolled-back transactions, they are blocked until the recovery process restores the data and releases the lock.

---

4. Down AMP recovery begins.

## Multiple Recovery Sessions

A recovery session is the set of actions to be taken as a result of the Teradata system restart for all transactions that were in progress at the time the Teradata system restarted.

All of the online AMPs at the time of the restart work together to determine which transactions were complete, and which ones were not completed. Completed transactions are ignored and incomplete ones are backed out.

If new work or transactions are allowed in during recovery, and another restart occurs, an additional recovery session is created. Then there will be two recovery sessions:

- The first one that was created for the previous restart
- The new one that was created for the current work

Since there is a sequential relationship between these two recovery sets and they are inherently mutually exclusive, they are kept as separate operations.

Therefore, if a system crash or user abort occurs and the amount of work to be done in each recovery session is large, then three, four or more recovery sessions may be created. Each session exists until all the incomplete transactions of the session are rolled back.

The issue of multiple recovery sessions may be avoided by having all the restarts be COLDWAIT, since the WAIT means to wait for recovery to complete before allowing the Teradata system to accept new work from the hosts. Although the recovery proceeds faster, since it is not competing with any other new work for computing resources, the Teradata system remains totally unavailable to the database users.

## Deferred Transaction Recovery

If the Teradata system crashes, a COLD restart is activated and an online transaction recovery or deferred recovery is started. The only time a deferred recovery is not done is when the operator enters a COLDWAIT restart.

Deferred transaction recovery allows new transactions to come in from the connected hosts. The process of bringing the Teradata system up after a crash causes the locks to be set. If new transactions should attempt to reference the inconsistent data of the to-be-rolled-back transactions, they are blocked until the recovery process restores the data and releases the lock.

## Down AMP Recovery

Down AMP Recovery is a process that handles all changes to entire tables or rows, either fallback and primary, while the AMP is down or offline. Down AMP Recovery updates a recovering AMP with data that the Teradata system processed while the AMP was down. The down AMP is considered to be in offline catchup mode. Catchup mode indicates that the AMP is logically offline and in the process of updating its tables so that they are synchronized with the online AMPs in the cluster.

In a crash or restart condition, it is possible to lose an AMP from, for example, a CPU board failure without losing its underlying disk data. The remainder of the Teradata system can then restart and recover the transactions without the failed AMP.

After the down AMP is ready to rejoin the Teradata system, the down AMP recovers the lost data by performing the following steps:

1. Restoring the data to a consistent state, relative to the transactions the down AMP was working on at the time of the failure.

The down AMP applies the information in its Transient Journal against its underlying data. Moreover, the down AMP must concur with the choice of the rest of the Teradata system whether to roll back or commit each transaction.

2. Updating the restored data to match all changes made to the online Teradata system while the AMP was down (Down AMP Recovery).

### Down AMP Recovery Operations Display

The following display indicates the fields that are referenced in the down AMP recovery operations explanation on the next page. This display is printed out by rcvmanager. For a description of the fields in the following display, see [Down/Catchup AMP Recovery Status](#).

DOWN AMP RECOVERY STATUS at HH:MM:SS MM/DD/YY

| AMP to be<br>caught up                           | Pass | Current Pass |         | Next Pass |        |
|--------------------------------------------------|------|--------------|---------|-----------|--------|
|                                                  |      | OJ           | CJ      | OJ        | CJ     |
| -----                                            | ---- | ---          | -----   | ---       | -----  |
| 0001                                             | 0    | 0            | 0       | 1         | 1,081  |
| - AMP Status: Online Catchup                     |      |              |         |           |        |
| - Not currently executing recovery               |      |              |         |           |        |
| 0002                                             | 1    | 5            | 145,822 | 1         | 25,081 |
| - AMP Status: Offline Catchup                    |      |              |         |           |        |
| - Transaction Recovery: 25,488 TJ Rows           |      |              |         |           |        |
| 0003                                             | 1    | 0            | 2,142   | 0         | 0      |
| - AMP Status: Online Catchup                     |      |              |         |           |        |
| - Change Row Recovery: 26% complete in this pass |      |              |         |           |        |
| 0004                                             | 2    | 2            | 201,558 | 0         | 4,228  |
| - AMP Status: Offline Catchup                    |      |              |         |           |        |
| - Rebuilding Database1.Table1: 45% complete      |      |              |         |           |        |
| 0005*                                            | 4    | 0            | 0       | 0         | 2,888  |
| - AMP Status: Offline Catchup                    |      |              |         |           |        |
| - Between passes                                 |      |              |         |           |        |

\* - would probably be placed in online catch up if a restart occurred



## Recovering Down AMPs

### Note:

Perform steps one through four only when offline catchup AMPs exist. If an AMP is still physically down, do not perform these steps.

To recover a down AMP, do the following:

1. Previously down AMPs that are now available begin their local transaction recovery (referred to as offline catchup AMPs). This step must be completed before going to step 2.
2. The extracted Ordered System Change Journal (OJ) is processed, rebuilding various subtables.
3. The extracted Changed Row Journal (CJ) rows are then processed by sending the changed rows (or notification that the rows were deleted) to the catchup AMP.
4. The OJ entries are applied. Ones which take a significant amount of time are the build operations.
5. Current CJ entries are applied. Each CJ entry represents one row to be updated (insert, delete, update).
6. Online AMPs in the down AMP cluster extract all the current build records from the OJ and CJ, and sort them to eliminate duplicates. Any CJ entries referring to rows in a table which will be rebuilt are deleted. Other OJ operations are sent to the catchup AMP for execution.
7. The Next Pass OJ and CJ entries become the Current Pass entries.
8. Repeat steps 2 through 7 indefinitely.
9. After the AMP is sufficiently caught up, it becomes eligible to become an online catchup AMP on the next restart. This is denoted by displaying an asterisk (\*) under the column entitled, AMP to be caught up, in the DOWN AMP RECOVERY STATUS screen of the rcvmanager status display. See the previous display screen.

## Recovery Journal

When an AMP in a cluster goes down, and the fallback option is specified for a table, the Down AMP Recovery Journal (RJ) records changes made to the fallback tables that are applicable to the down AMP. The journal is active only during an AMP failure and is only used for fallback tables.

The Recovery Journal process for a down AMP is as follows:

1. Operational AMPs in the cluster begin logging entries into the Down AMP RJ.
2. The Down AMP RJ records the changes that should have been made to fallback-protected rows of the inoperative AMP.
3. The changes are applied when the down AMP recovers.

The Recovery Journal maintains two sets of records: the Changed Row Journal (CJ) and the Ordered System Change Journal (OJ).

### Changed Row Journal

The CJ logs changed rows in an AMP cluster by logging pointers to the rows that have been changed, but does not log the actual rows. The CJ recovery needs to match its data against all changes made to the online Teradata system while the AMP was down. The most common type of changes includes modifications made

to individual rows by Data Manipulation Language (DML) operations of inserting, deleting, and updating rows in pre-existing tables.

Each modified row is remembered in the Teradata system CJ, which is local to each AMP on which the modification takes place. Since the modification is done only while an AMP is down, the CJ is only populated on AMPs in a cluster while an AMP is down. AMPs are arranged into a group called clusters so that each AMP provides fallback protection to other members within that same group.

Entries stored in the CJ include only the table ID and row ID of the row which was modified. When the down AMP recovers, the actual row is extracted from the fallback or primary subtable; the row image is not stored redundantly in the CJ.

### Ordered System Change Journal

The OJ logs data changes that are applied to the online Teradata system while an AMP is down, such as building the index, creating a permanent journal, and performing a table rebuild. These types of changes may be termed Teradata system-level changes and involve Data Definition Language (DDL) operations, since they affect all AMPs, not only a single row update.

These changes cover DDL operations such as, DROP TABLE, CREATE TABLE, and CREATE/DROP INDEX. Other operations include a change to every single row within a table, for example, drop a column. For these types of operations, recovery involves copying every single row of the table, for those rows the AMP owns, over to that down AMP (a table rebuild).

Therefore, the majority of the rows found in the OJ are build records that identify tables that need to be built for created tables, or rebuilt when the AMP is recovered. Other OJ records are HUT Lock set and release records, and in-doubt two-phase commit transaction.

### Deferred Down AMP Recovery

The process of Deferred Down AMP Recovery means that while a down AMP remains down and recovering, the rest of the Teradata system continues its operations with the connected hosts.

### Offline Catchup Mode

In offline catchup mode, the new transactions for the down AMP are performed by other AMPs in the cluster. Therefore, new change row and Teradata system change entries are made while the down AMP is processing the old ones. While the AMP is in catchup mode, the AMP is considered logically offline.

To set a down AMP to offline catchup mode, do the following:

1. Start the Vproc Manager utility.
2. To list the Teradata system logical configuration, type:

```
status;
```

A screen similar to the following appears:

```
DBS LOGICAL CONFIGURATION
Vproc  Rel.   Node          Crash  Vproc  Config  Config  Cluster/  RvcJrnl/
Number Vproc# ID   Movable Count  State  Status  Type     Host No.  Host Type
0*      1      0-0   No      0      Online Online  AMP      0          On
```

|       |   |     |    |   |         |        |     |    |     |
|-------|---|-----|----|---|---------|--------|-----|----|-----|
| 1     | 2 | 0-0 | No | 0 | Offline | Down   | AMP | 0  | On  |
| 16383 | 3 | 0-0 | No | 0 | Online  | Online | PE  | 52 | COP |

---

\* DBS Control AMP

DBS State: Logons are enabled - Users are logged on

DBS RestartKind: COLD

The disable list is empty

3. To bring up the downed AMP, type:

```
set 1 = online;
```

If vproc 1 goes into catchup mode, the following message appears:

```
Vproc 1 will begin recovery in the background via the Recovery Control Task
```

4. To verify that the AMP is in Utility Catchup mode, type:

```
status:
```

While in the offline mode, the AMP has OJ build records to process and/or a large number of CJ rows. This reduces the amount of data that is locked, but any new transactions on the online Teradata system creates additional OJ and CJ rows. Since you cannot place an offline AMP into the online Teradata system, these passes could continue forever or until the Teradata system restarts.

A screen similar to the following appears:

#### DBS LOGICAL CONFIGURATION

| Vproc Number | Rel. Vproc# | Node ID | Movable | Crash Count | Vproc State | Config Status | Config Type | Cluster/ Host No. | RvcJrnl/ Host Type |
|--------------|-------------|---------|---------|-------------|-------------|---------------|-------------|-------------------|--------------------|
| 0*           | 1           | 0-0     | No      | 0           | Online      | Online        | AMP         | 0                 | On                 |
| 1            | 2           | 0-0     | No      | 0           | Utility     | Catchup       | AMP         | 0                 | On                 |
| 16383        | 3           | 0-0     | No      | 0           | Online      | Online        | PE          | 52                | COP                |

---

\* DBS Control AMP

DBS State: Logons are enabled - The system is quiescent

DBS RestartKind: COLD

The disable list is empty

To verify if an offline AMP, for example, is almost in catchup, do the following:

1. Start rcvmanager.
2. To check the offline AMP status, type:

```
list status;
```

One of the following messages appears:

| Message...                                                                                                                                                                             | Means that the...                                                                          |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>- AMP Status: Not in recovery</li> <li>- Down</li> </ul>                                                                                        | AMP is still down.                                                                         |
| <ul style="list-style-type: none"> <li>- AMP Status: Offline Catchup</li> <li>- Executing miscellaneous Recovery actions</li> </ul>                                                    | AMP is in catchup mode.                                                                    |
| <ul style="list-style-type: none"> <li>- AMP Status: Offline Catchup</li> <li>- Between Passes</li> <li>* - would probably be placed in online catchup if a restart occurs.</li> </ul> | AMP is in catchup mode, and if a restart occurs, the AMP is placed in online catchup mode. |

If a COLDWAIT restart is performed, the operations are similar to those of Transaction Recovery. In COLDWAIT, the Teradata system remains offline with no incoming transactions until all recovering AMPs have fully recovered. After recovery, the AMPs are all set to online status, and the Teradata system completes start-up.

In offline catchup, catchup tries to run faster than the new update transactions coming in online. The other AMPs handle the fallback responsibility for the down AMP, as well as the additional work involved in writing CJ or OJ records.

If an AMP is designated to be in offline catchup mode, then you must initiate a COLD restart, but only after the AMP has sufficiently recovered, to bring the offline AMP back online. "Sufficiently recovered" refers to the third message in the table above.

### Online Catchup Mode

In online catchup mode, the previously down AMP will also accept transactions, as will the rest of the Teradata system. In this mode, all the data that needs updating is locked, so that new data does not operate on the obsolete data.

When the down AMP has a small amount to recover, it can be placed into online catchup.

Since the AMP is online and participating in the new transaction coming into the Teradata system, no new CJ or OJ entries are being created for it. After all of the residual CJ processing is completed, the Teradata system becomes online. Start the Vproc Manager utility and check the status of the database. The Vproc State and Config Status should appear as Online. At this point, the AMP rejoins the other online AMPs automatically without the need of a Teradata system restart.

## Startup/Restart Messages

At startup, messages are displayed to report the startup progress and the Teradata system status. This report includes some recovery process messages as shown below:

```
H005 Control AMP 001-2
```

```
98/02/06 11:35:31 Running DBS Version: 14.00.00.00
```

```
98/02/06 11:35:31 Running PDE Version: 14.00.00.00
```

```

98/02/06 11:35:32 Current DBS config maps are synchronized (Version: 11)
98/02/06 11:35:32 New DBS config maps are synchronized (Version: 11)
98/02/06 11:35:32 AMP 0000 has been selected as the Control AMP
98/02/06 11:35:33 Initializing DBS Vprocs
98/02/06 11:35:36 Configuration is operational
98/02/06 11:35:36 Starting AMP partitions
98/02/06 11:35:40 Connection accepted
98/02/06 11:36:02 Voting for transaction recovery
98/02/06 11:36:05 Recovery session 1 contains 77 rows on AMP 0000
98/02/06 11:36:16 Starting transaction recovery
98/02/06 11:36:18 Completed transaction recovery
98/02/06 11:36:18 Starting PE partitions
98/02/06 11:37:03 System is operational
98/02/06 11:37:05 Users are logged on
98/02/06 11:37:05 Logons are enabled

```

The following table lists and explains these and other messages.

| Message #...                                                        | Means...                                                                                                                                                                                                                                                                                                                                   |
|---------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 - AMP <i>mmmm</i> in cluster <i>nnn</i> is back online            | the specified AMP which was offline is now back online.                                                                                                                                                                                                                                                                                    |
| 2 - AMP <i>mmmm</i> is in offline catchup                           | the AMP is in offline catchup mode.                                                                                                                                                                                                                                                                                                        |
| 3 - AMP <i>mmmm</i> is in online catchup                            | the AMP is in online catchup mode.                                                                                                                                                                                                                                                                                                         |
| 4 - AMP <i>mmmm</i> not brought online for the following reason(s): | one of the following: <ul style="list-style-type: none"> <li>• Changed row journal count of ZZ,ZZZ,ZZ9 exceeds 3000 rows.</li> <li>• At least one “long” running ordered journal record remains.</li> <li>• At least one HUT lock might exist in the cluster.</li> <li>• At least one two-phase-commit transaction is in doubt.</li> </ul> |
| 5 - Completed transaction recovery                                  | Transaction Recovery for online AMPs is complete.                                                                                                                                                                                                                                                                                          |
| 6 - Recovering down AMPs                                            | the recovery process is updating tables on AMPs that are in offline catchup.<br><b>Note:</b><br>If there was no down AMP in catchup recovery mode, this message is not displayed.                                                                                                                                                          |

| Message #...                                         | Means...                                                                                                                                                                                                                                                                                                                                                                         |
|------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7 - Recovery session 1 contains 103 rows on AMP 0001 | that AMP 0001 has the largest number of rows in the transient journal for the given recovery session. Other AMPs probably have a similar or less rows for the same recovery session. The count reflects how many rows were in the transient journal at the time of the Teradata system restart. It does not necessarily mean a rollback action has to be performed for each row. |
| 8 - Starting transaction recovery                    | transaction recovery has started on online AMPs.                                                                                                                                                                                                                                                                                                                                 |
| 9 - Startup will wait for recovery to complete       | the user initiated a COLDWAIT restart.                                                                                                                                                                                                                                                                                                                                           |
| 10 - Voting for transaction recovery                 | the application software is determining if there were any incomplete transactions left from the previous operation. Voting is a process by which each AMP examines the transactions that were in process. Any uncompleted transaction will have to be rolled back.                                                                                                               |

## Restarts

Two types of restarts exist:

- Automatic
- User-initiated

### Automatic Restarts

When a software or hardware failure occurs, the Teradata system automatically attempts to bring itself back up into an operational state. This type of restart results in an automatic execution of a Teradata system recovery operation. As part of the restart processing, the Teradata system saves the error codes that were generated, reloads Teradata system software, and enables logons.

### User-Initiated Restarts

Manual restarts are activated by the user. Users can initiate a restart from any of the hardware switches or by one of the following methods:

- Issue the RESTART command. For information on the RESTART command, see [Vproc Manager \(vprocmanager\)](#).
- At the system command line, type the following command:  

```
tpareset comment-string
```

where *comment-string* briefly describes the reason for the restart.

## Canceling Rollback on Tables

rcvmanager provides a mechanism to cancel or skip the rollback of specified tables during a database restart or an aborted, online transaction. Canceling the rollback of long-running transactions and unwanted tables improves the availability of Teradata system resources and reduces the database startup time after a crash.

When the CANCEL ROLLBACK ON TABLE command is executed for a table, the database marks the related table header invalid. Only the rollback pertaining to the specified table in the transaction is canceled. The rollback processing for the rest of the transaction is not impacted.

---

### Note:

You cannot cancel rollback on a table that is not in the rollback list. Before issuing the CANCEL ROLLBACK ON TABLE command, you should use the LIST ROLLBACK TABLES command to see which tables are undergoing rollback. You can cancel rollback only on tables that appear in this list.

---

Use the CANCEL ROLLBACK ON TABLE command when one of the following occurs:

- The rollback of a table is likely to take longer than its restoration.
- The table, such as a temporary table, is unimportant.

### NOTICE

Teradata recommends that you use the CANCEL ROLLBACK ON TABLE command with caution because the target table becomes invalid and unusable after executing this command. Teradata highly recommends that you perform a DELETE ALL operation on the table after canceling rollback on it.

The typical process of canceling rollback on a table is as follows:

1. The rollback is taking too long.
2. You identify a large table(s) that can be restored faster than the rollback will take.
3. You perform a LIST ROLLBACK TABLES to see which tables are undergoing rollback.
4. You perform a CANCEL ROLLBACK ON TABLE.
5. You perform a DELETE ALL and restore the table(s).

The only time that you would not restore the table(s) immediately is if you do not need the table(s). You still should perform a DELETE ALL immediately. You have to know what you are going to do about the invalid table prior to performing the CANCEL ROLLBACK ON TABLE. And you will have to do it immediately to get the database back online.

You can reuse the table only when one of the following occurs:

- You drop the table and create it again.
- You restore the table from an archived backup.
- You perform a DELETE ALL operation on that table if you do not want to lose the DDL associated with the table. When you issue a DELETE ALL, the partially rolled back rows can be removed, and the table is made usable.

Before canceling rollback on tables, see [CANCEL ROLLBACK ON TABLE](#).

To cancel the rollback on tables, do the following:

1. Start rcvmanager.
2. At the command prompt, type:

```
LIST ROLLBACK TABLES;
```

rcvmanager displays the names and details of the tables undergoing rollback.

3. At the command prompt, type:

```
CANCEL ROLLBACK ON TABLE nnnn:mmmm, nnnn:mmmm, ...;
```

The table IDs specified must appear on the rollback list from step 2.

rcvmanager displays the following:

```
Type the password for user DBC or press the Enter key to return:
```

---

**Note:**

You need to specify the DBC password only for the first use of the CANCEL ROLLBACK ON TABLE command in a rcvmanager session. For subsequent use of the command, rcvmanager does not ask for the DBC password.

---

4. Type the password for user DBC or press the **Enter** key.

If logon fails due to an incorrect password or for some other reason, the following message appears:

```
*** Logon failed ***
```

Vantage returns to the RcvManager command prompt. Repeat step 2 if this occurs.

If you successfully log on to Vantage, the following message appears:

```
Rollback will be cancelled for:
  mmmm:nnnn "DBname"."TableName"
Confirm y/n ?
```

5. Type **Y** to confirm.

For more information on rollbacks, see the following commands:

- [CANCEL ROLLBACK ON TABLE](#)
- [LIST CANCEL ROLLBACK TABLES](#)
- [LIST ROLLBACK TABLES](#)



## Retrieving Tables

You can perform a single table retrieve operation on tables whose rollback was canceled. To do so, use the LOCKING request modifier with the READ OVERRIDE option. You cannot perform update operations (UPDATE, INSERT, DELETE, and MERGE requests) on such tables. See also the information about the LOCKING modifier in *Teradata Vantage™ - SQL Data Manipulation Language*, B035-1146.

## Recovery Manager Commands

Recovery Manager presents a command-line environment that allows the entry of Recovery Manager commands. Commands must be terminated with a semicolon.

The following table summarizes the commands.

| Command                                     | Description                                                                                                                                                                                             |
|---------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">CANCEL ROLLBACK ON TABLE</a>    | Cancels rollback processing on tables currently undergoing rollback as part of a Teradata system recovery or an online, user-transaction abort.                                                         |
| <a href="#">DEFAULT PRIORITY</a>            | Sets the priorities to the default values.                                                                                                                                                              |
| <a href="#">HELP</a>                        | Displays the syntax for the commands supported by rcvmanager.                                                                                                                                           |
| <a href="#">LIST CANCEL ROLLBACK TABLES</a> | Displays a report containing the table-id, database name, and table name of the tables whose rollback processing during an online, user-requested abort or during Teradata system recovery is canceled. |
| <a href="#">LIST LOCKS</a>                  | Displays all locks currently held by transaction recovery.                                                                                                                                              |
| <a href="#">LIST ROLLBACK TABLES</a>        | Displays a report with the names and details of the tables undergoing rollback in the Teradata system.                                                                                                  |
| <a href="#">LIST STATUS</a>                 | Displays transaction recovery information and AMP recovery information for unavailable AMPs or detailed information about the recovery process of a specific AMP.                                       |
| <a href="#">QUIT</a>                        | Exits rcvmanager.                                                                                                                                                                                       |
| <a href="#">REBUILD/ RECOVERY PRIORITY</a>  | Sets or displays a priority level for use by the Table Rebuild utility and a Teradata system recovery operation.                                                                                        |
| <a href="#">ROLLBACK SESSION WORKLOAD</a>   | Sets or displays the workload used by rollbacks for a specified host and session.                                                                                                                       |

The following sections describe each of the Recovery Manager commands in more detail.

## CANCEL ROLLBACK ON TABLE

The CANCEL ROLLBACK ON TABLE command allows you to cancel rollback processing on tables currently undergoing rollback as part of a Teradata system recovery or an online, user-transaction abort.

### Note:

Before using CANCEL ROLLBACK ON TABLE, you should use the LIST ROLLBACK TABLES command to obtain the table IDs of tables undergoing rollback. CANCEL ROLLBACK ON TABLE will fail for tables that do not appear on this list.

### Syntax

```
CANCEL ROLLBACK ON TABLE table_id [, ...]
```

### Syntax Elements

#### *table\_id*

The identifier for the table whose rollback is to be canceled. This is specified in hexadecimal format.

You can specify any number of *table\_ids* by separating them with commas.

The *table\_id* must be in the rollback tables list, meaning the table specified must be undergoing rollback.

### Usage Notes

When the CANCEL ROLLBACK ON TABLE command is executed for a table, the database marks the related table header invalid. Only the rollback pertaining to the specified table in the transaction is canceled. The rollback processing for the rest of the transaction is not impacted.

Use the CANCEL ROLLBACK ON TABLE command when one of the following occurs:

- The rollback of a table is likely to take longer than its restoration.
- The table, such as a temporary table, is unimportant.

### NOTICE

Teradata recommends that you use the CANCEL ROLLBACK ON TABLE command with caution because the target table becomes invalid and unusable after executing this command.

Teradata highly recommends that you perform a DELETE ALL operation on the table after canceling rollback on it.

The typical process of canceling rollback on a table is as follows:

1. The rollback is taking too long.

2. You identify a large table(s) that can be restored faster than the rollback will take.
3. You perform a LIST ROLLBACK TABLES to see which tables are undergoing rollback.
4. You perform a CANCEL ROLLBACK ON TABLE.

**Note:**

You can cancel rollbacks only on tables whose IDs appear on the rollback list from step 3.

5. You perform a DELETE ALL and restore the table(s).

The only time that you would not restore the table(s) immediately is if you do not need the table(s). You still should perform a DELETE ALL immediately. You have to know what you are going to do about the invalid table prior to performing the CANCEL ROLLBACK ON TABLE. And you will have to do it immediately to get the Teradata system back online.

You can reuse the table only when one of the following occurs:

- You drop the table and create it again.
- You restore the table from an archived backup.
- You perform a DELETE ALL operation on that table if you do not want to lose the DDL associated with the table. When you issue a DELETE ALL, the partially rolled back rows can be removed, and the table is made usable.

Upon specifying the CANCEL ROLLBACK ON TABLE command for the first time in a rcvmanager session, you are prompted to type the password for user DBC. Although any user can execute this command, the DBC password is required. This is a built-in safeguard to restrict the use of the feature.

This command is not instantaneous. It takes effect after reading the Transient Journal (TJ) rows for all the tables specified with the command.

You can cancel the rollback on a table even after a part of the transaction rollback is complete on that table.

The priority level in effect during a rollback also applies to the aborting of rollback.

The following usage rules apply to the CANCEL ROLLBACK ON TABLE command:

- The *table\_id* specified with the CANCEL ROLLBACK ON TABLE command must exist in the rollback tables list, meaning the table must have been marked for rollback.

You can specify any number of tables for canceling rollback. rcvmanager verifies each *table\_id*. If some of these tables do not exist in the rollback list, rcvmanager reports error messages for the non-existing tables and asks for a single confirmation for canceling rollback on all the valid tables. See the Examples below.

**Example 1: Valid Command**

Assume that the table with table-id 6712 exists.

```
> CANCEL ROLLBACK ON TABLE 0:6712;
Type the password for user DBC or press the Enter key to return:
> dbc
```

```
Rollback will be cancelled for:
0000:6712 "EmployeeDB"."LogTable"
Confirm y/n ?
> y
```

rcvmanager cancels rollback on LogTable.

## Example 2: Invalid Command

### Table does not exist

Assume that the table with *table\_id* 6708 does not exist in the rollback list.

```
> CANCEL ROLLBACK ON TABLE 0:6708;
Table 0:6708 does not exist in rollback list.
Enter command, "QUIT;" or "HELP;" :
```

rcvmanager has ignored the command because the table does not exist.

You cannot cancel rollback on tables that have any referential integrity constraints.

---

### Note:

An exception to this rule is the self-referencing table.

---

The *table\_ids* of all referenced and referencing tables are marked with an asterisk (\*) in the rollback list generated by the LIST ROLLBACK TABLES command.

### Specifying multiple tables

When you specify multiple tables in the CANCEL ROLLBACK ON TABLE command, rcvmanager verifies each table for referential integrity constraints.

Assume three tables T1 (*table\_id* 6710), T2 (*table\_id* 6711) and T3 (*table\_id* 6712) exist in a rollback list, and T2 has a foreign key referencing T1. Assume that T3 has only a self-reference and no constraints that reference other tables.

```
> CANCEL ROLLBACK ON TABLE 0:6710;
Entry ignored as table 0:6710 has referential integrity constraint.
Enter command, "QUIT;" or "HELP;" :
> CANCEL ROLLBACK ON TABLE 0:6711;
Entry ignored as table 0:6711 has referential integrity constraint.
Enter command, "QUIT;" or "HELP;" :
> CANCEL ROLLBACK ON TABLE 0:6712;
Rollback will be cancelled for:
0000:6712 "SG"."T3"
Confirm y/n ?
> y
```

rcvmanager cancels rollback only on table T3.

### Specifying tables associated with join or hash indexes

If the tables specified for canceling rollback are associated with join or hash indexes, then rcvmanager lists all such tables and asks for a single confirmation.

Upon confirmation (y), the tables, as well as the associated join and hash indexes, are marked as invalid.

### Specifying tables associated with join or hash indexes, or have errors

When you specify multiple tables with the CANCEL ROLLBACK ON TABLE command, and some tables have join or hash indexes and some have errors, then rcvmanager asks for a single confirmation for all the valid tables.

Assume that Temp\_Table with *table\_id* 6707 is associated with a join index.

```
> CANCEL ROLLBACK ON TABLE 0:6707;
Rollback will be cancelled for:
0000:6707 "EmployeeDB"."Temp_Table"
The following join and/or hash indexes will be invalidated:
"EmployeeDB"."Sal_Join"
Confirm y/n ?
> y
```

If the *table\_id* specified with the CANCEL ROLLBACK ON TABLE command was specified previously or more than once in the same command, then rcvmanager reports a message indicating a duplicate entry.

Assume that a rollback was canceled previously on table-id 6712.

```
> CANCEL ROLLBACK ON TABLE 0:6712;
```

The following message appears:

```
Table 0:6712 already on the cancel rollback list, input ignored.
Enter command, "QUIT;" or "HELP;"
```

When you cancel rollback on a table, it becomes invalid and unavailable for any subsequent transactions.

If any user attempts an update on such tables, then the following message appears:

```
"Invalid operation on table table-name."
```

### Rollback on table already canceled

In the following example, the INSERT into LogTable fails because rollback on the table was canceled earlier, and the table is invalid.

```

BTEQ -- Enter your DBC/SQL request or BTEQ command:
      INSERT INTO EmployeeDB.LogTable;
*** Failure 5792 Invalid operation on table 'LogTable'.
      Statement# 1, Info =0
*** Total elapsed time was 1 second.

```

### Valid Operations on Rollback-Canceled Tables

Canceling rollback on a table makes data in the table invalid and unusable. Only the following operations are valid on the tables on which CANCEL ROLLBACK ON TABLE is performed.

| To...                                                    | Use the...                                              |
|----------------------------------------------------------|---------------------------------------------------------|
| delete all the rows in the table and make it valid again | DELETE statement with the ALL option.                   |
| drop the table so that you can create it again           | DROP TABLE statement.                                   |
| retrieve a single table for SELECT operations            | LOCKING request modifier with the READ OVERRIDE option. |
| restore the table from an archive using DSA utility      | RESTORE command.                                        |
| take a dump of the table using DSA utility               | DUMP command.                                           |
| rebuild the table headers                                | <a href="#">Table Rebuild (rebuild)</a> utility.        |

#### Note:

CheckTable utility skips checking on tables whose rollback you cancel.

## DEFAULT PRIORITY

The DEFAULT PRIORITY command sets the priorities to the default values (that is, REBUILD will be set to MEDIUM and RECOVERY to LOW).

### Syntax

```
DEFAULT PRIORITY ;
```

### Usage Notes

If the PRIORITY command has never been executed, or a system initialization (sysinit) has been performed on the system, the priorities are set to the default values. The priority remains the same until an initial or subsequent PRIORITY command changes it again. When the priority command is entered during a rebuild or recovery operation, several minutes may elapse before the new priorities take effect.

When you type the DEFAULT PRIORITY command, the event is logged in the /var/adm/streams/error.mm-dd file in the Teradata system you are working on, and the following messages are displayed.

```
YY/MM/DD HH:MM:SS RECOVERY priority changed to LOW; it was  
<old priority>
```

```
YY/MM/DD HH:MM:SS REBUILD priority changed to MEDIUM; it  
was <old priority>
```

When the default priority command is entered, both messages are generated.

## HELP

The HELP command displays the syntax for the commands supported by rcvmanager.

### Syntax

```
HELP ;
```

### Example: Recovery Manager help

The HELP command displays information for all the rcvmanager commands:

RCVMANAGER provides a means for the user to interact with the recovery sub-system.

The syntax of an RCVMANAGER command is:

```
HELP;
```

```
QUIT;
```

```

      { STATUS [<proc-id>]      }
      { LOCKS                  }
LIST { ROLLBACK TABLES        }
      { CANCEL ROLLBACK TABLES };

```

```

{ REBUILD  }           [ Low    ]
{          } PRIORITY  [ Medium ] ;
{ RECOVERY }           [ High   ]

```

```
DEFAULT PRIORITY;
```

```
CANCEL ROLLBACK ON TABLE <table-id> [{,<table-id>} ...] ;
```

The HELP command displays this help message text.

The QUIT command will terminate the RcvManager Utility.

```

.
.
.

```



## LIST CANCEL ROLLBACK TABLES

The LIST CANCEL ROLLBACK TABLES command displays the table-id, database name, and table name of the tables whose rollback processing during an online, user-requested abort or during Teradata system recovery is canceled.

### Syntax

```
LIST CANCEL ROLLBACK TABLES ;
```

### Usage Notes

This report lists all the tables for which rollback is being canceled, as the process of cancellation of rollback is not instantaneous. The report is in alphabetical order, by the database name and table name.

The list excludes Permanent Journal (PJ) tables and all DBC tables except DBQL and ResUsage tables.

#### Note:

The LIST ROLLBACK TABLES command report does not include invalid tables, that is, tables on which rollback is being canceled. Therefore, if all the tables in a session have been specified for rollback cancellation, they appear only in the output of the LIST CANCEL ROLLBACK TABLES command.

If no tables on which rollback is canceled exist, then only the column headings are displayed.

### Example: Displaying a list of tables with canceled rollback processing

The following is sample output of the LIST CANCEL ROLLBACK TABLES command.

```
PENDING CANCEL ROLLBACK TABLES AT 03:02:25 01/11/15
```

| Table ID  | Name                        |
|-----------|-----------------------------|
| 0000:6707 | "Department"."Dept_Details" |
| 0000:6712 | "Department"."Temp_Table"   |

## LIST LOCKS

The LIST LOCKS command displays all locks currently held by transaction recovery.

### Syntax

```
LIST LOCKS ;
```

### Usage Notes

The report displays the mode of the lock held (write or exclusive), the object type locked (database, table, row range, or row hash) and the name of the object. The report is sorted alphabetically by object name.

For row range and row hash locks, the row information does not display. Only the table within which the row resides is displayed.

If rcvmanager is unable to determine the database name associated with an object, rcvmanager displays the database ID in decimal and hex. The same is true if the table name cannot be determined.

LIST LOCKS displays only those locks currently held by transaction recovery. You cannot display locks held by online catchup or offline catchup.

### Example: Displaying a list of locks held by transaction recovery

Only a single report is generated by the LIST LOCKS command. An example of this report is shown below:

```
LOCKS HELD BY ONLINE TRANSACTION RECOVERY at
02:29:16 04/06/16
```

| Lock<br>Mode | Lock<br>Object | Object<br>Name            |
|--------------|----------------|---------------------------|
| ----         | -----          | ----                      |
| Write        | Database       | "AssetsDB"                |
| Write        | Row hash       | "Clients"."TurnOver"      |
| Write        | Row hash       | "EmployeeInfo"."NewHires" |
| Exclusive    | Table          | "SampleDB"."SampleTable"  |

## LIST ROLLBACK TABLES

The LIST ROLLBACK TABLES command displays the names and details of the tables undergoing rollback in the Teradata system.

### Syntax

```
LIST ROLLBACK TABLES ;
```

### Usage Notes

The tables undergoing rollback must satisfy the following criteria:

- The table in a transaction must have more than 10000 rows yet to rollback on at least one AMP.
- The transaction must be in abort status.

The report displays only the headings if no rollback is in progress in the Teradata system, or if no table satisfies these criteria.

The list is in descending order of the Host and Session, and then in alphabetical order by the database name and table name.

The list excludes Permanent Journal (PJ) tables and all DBC tables except DBQL and ResUsage tables.

An asterisk (\*) appears after the table name if the table has any Referential Integrity constraints. You cannot cancel rollback on such tables.

Table names having non-printable characters are displayed in hexadecimal.

The list is in two parts, as shown in the following table.

#### Note:

Global Temporary tables and Volatile tables are not listed.

| This part...                    | Displays a list of tables in the sessions that are undergoing rollback as part of... |
|---------------------------------|--------------------------------------------------------------------------------------|
| online user rollback tables     | user-requested abort.                                                                |
| system recovery rollback tables | Teradata system recovery.                                                            |

The following table explains what information the report columns provide:

| Column...           | Specifies the...                                           |
|---------------------|------------------------------------------------------------|
| Host                | ID of the host machine.                                    |
| Session             | session number in which rollback is in progress.           |
| User ID             | ID of the user who has initiated the session.              |
| Workload Definition | Workload at which all transactions in the session perform. |

| Column...                    | Specifies the...                                                                                          |
|------------------------------|-----------------------------------------------------------------------------------------------------------|
| AMP W/Count                  | AMP on which the maximum number of Transient Journal rows are left to be processed.                       |
| TJ Rows Left<br>TJ Row Count | maximum number of rows left to be processed in the Transient Journal for the specific session on any AMP. |
| TJ Rows Done                 | number of rows processed in the Transient Journal for the specific session.                               |
| Time Est.                    | estimated time (in hours and minutes) left for the rollback to complete for that session.                 |

### Example: Displaying a list of the tables undergoing rollback processing

The following is a sample output of the LIST ROLLBACK TABLES command.

```
TABLES BEING ROLLED BACK AT 14:25:34 02/11/15
```

```
ONLINE USER ROLLBACK TABLE LIST
```

| Host | Session | User ID   | Workload Definition | AMP W/Count |
|------|---------|-----------|---------------------|-------------|
| 110  | 23857   | 0000:0481 | NoWD-PG1054         | 1           |

| TJ Rows Left | TJ Rows Done | Time Est. |
|--------------|--------------|-----------|
| 30,478       | 1,098        | 01:37     |

| Table ID  | Name                    |
|-----------|-------------------------|
| 0000:3883 | "rcv_user1"."cr_in001"  |
| 0000:388B | "rcv_user1"."cr_in002"  |
| 0000:3884 | "rcv_user1"."cr_ri001"* |
| 0000:3889 | "rcv_user1"."cr_ri002"* |

```
SYSTEM RECOVERY ROLLBACK TABLE LIST
```

| Host | Session | TJ Row Count |
|------|---------|--------------|
| 0    | 0       | 30,032       |

| Table ID | Name |
|----------|------|
|----------|------|

```
-----  
0005:045A  "EmployeeDB"."Parent_Table"*  
0005:045B  "EmployeeDB"."Temp_Table"  
  
* - Referential Integrity table, cannot be used in CANCEL ROLLBACK ON  
TABLE command.
```

## LIST STATUS

The LIST STATUS command generates two reports:

- Online Transaction Recovery Journal Counts
- Down/Catchup AMP Recovery Status

The first reports transaction recovery information, and the second reports AMP recovery information for unavailable AMPs.

### Syntax

```
LIST STATUS ;
```

### Syntax Elements

#### STATUS

Transaction recovery information and AMP recovery information for unavailable AMPs.

- The displayed row count of a large sized table will be rounded to the nearest thousand.
- The recovery build records specify rebuild operations separately for each index of a table. This causes a rebuild of both the primary data and fallback data for that index, so the displayed sector count will differ from the total size for the table as derived using other methods.
- The sector count that is displayed is the sum of the estimated number of sectors for both primary and fallback rows to be copied over from the other AMPs in the cluster or recreated Non Unique Secondary Index (NUSI) indexes for all index subtables of the table which have an OJ build record. This includes the OJ build records of both the current pass and the next pass.

### Online Transaction Recovery Journal Counts

The Online Transaction Recovery Journal Counts allows the user to monitor transaction recovery processing. It displays information on the transient journal and transactions that were in progress during transaction recovery processing.

### Report Information

This report displays a list of all active recovery sessions and the maximum number of transaction journal rows remaining to be processed for the AMP that has this maximum count.

Since all AMPs must complete processing of a given recovery session before the processing of the next session begins, this information is sufficient to calculate the worst-case count of transaction journal entries to be scanned. You can use this count as a rough guide to recovery processing time.

However, because this is only a rough guide, you must also take into consideration the following variables when estimating the time involved in the recovery process:

- The amount of work required by recovery to process a given transaction journal row can vary by orders of magnitude; that is, some rows can be processed in a fraction of a second, whereas others can take hours to process.
- The transaction journal may contain many rows which require no recovery processing.

The online transaction recovery journal counts are updated by each AMP every time a checkpoint is taken.

Thus, every time an AMP performs checkpoint, its online transaction recovery journal count is decreased by 1000, and a later LIST STATUS command may display different results.

If no recovery sessions are active, the report title is printed without any contents.

### Example: Online transaction recovery journal counts report

ONLINE TRANSACTION RECOVERY JOURNAL COUNTS at 13:49:28 98/02/06

| Recovery<br>Session | Count  | AMP<br>W/Count |
|---------------------|--------|----------------|
| -----               | -----  | -----          |
| 1                   | 765    | 12             |
| 2                   | 38,432 | 5              |
| 3                   | 1,388  | 20             |

#### Note:

A new recovery session is created for each restart that is not a COLDWAIT restart.

### Down/Catchup AMP Recovery Status

The example report contains three sets of data lines (together with a column header) for each AMP participating in the down AMP recovery process. The first data line is explained by the column headers. The second and third data lines indicate the status.

The following table describes the Header and first-line data.

| Data                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AMP to be caught up | <p>Indicates the AMP number to be caught up.</p> <p>In addition to the processor number, this column might contain one of the following notations:</p> <ul style="list-style-type: none"> <li>• * (asterisk)</li> </ul> <p>Indicates that this AMP may be placed in online catchup if the Teradata system restarts. If the displayed information is no longer valid by the time the Teradata system restarts, the AMP might remain in offline catchup.</p> <ul style="list-style-type: none"> <li>• [1]</li> </ul> <p>Indicates that one or more host utility locks are held in the cluster of the recovering AMP and that the data on the AMP cannot be recovered because there is a conflict between the host utility lock and the recovery locking requirements.</p> <ul style="list-style-type: none"> <li>• [2]</li> </ul> |

| Data         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|              | Indicates that one or more 2 Phase Commit (2PC) in-doubt sessions exist within the cluster of the recovering AMP.                                                                                                                                                                                                                                                                                                                                                                                |
| Pass         | Indicates the number of the recovery pass. At the beginning of a recovery pass all rows in the OJ and CJ are extracted. This is the total number of rows to be processed during that pass. If additional changes are sent to the AMP during the processing of the current pass, they are queued up for the next pass. When the processing of rows of the current pass is complete, the pass number is incremented and the rows for the next pass are extracted from the OJ and CJ and processed. |
| Current Pass | The OJ column contains the number of rows in the ordered Teradata system change journal to be processed during the current recovery pass. The CJ column contains the number of rows in the changed row journal to be processed during the current recovery pass.                                                                                                                                                                                                                                 |
| Next Pass    | The OJ column contains the number of rows in the ordered Teradata system change journal to be processed during the next recovery pass. The CJ column contains the number of rows in the changed row journal to be processed during the next recovery pass.<br>If there are transactions that are updating user tables, these counts will go up as a result of those updated transactions. Subsequent displays could show an increase in the count.                                               |

The following table describes the second data line (AMP Status).

| Data            | Description                                                                                                                                                                                                                                          |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Online Catchup  | This AMP is online during recovery processing and accepts new work. Locks are applied against all objects that need to be updated before new work is accepted. In this status, the OJ count is usually zero and no new OJ or CJ entries are created. |
| Offline Catchup | This AMP is logically (not physically) offline during recovery processing. No new work is accepted by this AMP. Read locks are applied on the online AMPs in the cluster only against the specific data of an object that has to be updated.         |
| Not In Recovery | This AMP is not running down AMP recovery; it is physically offline.                                                                                                                                                                                 |

The following table describes the third data line (AMP Recovery Status Line).

| Data                                                      | Description                                                                                                                                                                                                                                                                                                                                                     |
|-----------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Transaction Recovery: ZZZ,ZZZ, ZZ9 TJ Rows                | Specifies the number of rows in the transaction recovery journal to be processed in this recovery pass. The transaction recovery journal contains the before images of all change objects affected by every transaction. Transaction recovery is the first step of the first recovery pass only. Each Z represents a digit and 9 represents any non-zero value. |
| Rebuilding Table [DBase.Table]: Z9% complete in this pass | Specifies the name of the table being rebuilt and the percentage of completion. Tables are rebuilt as a result of DDL changes (that is, drop or add a column) to a table, or a MultiLoad operation has affected a table while an AMP was down.                                                                                                                  |
| Change Row Recovery: Z9% complete in this pass            | This is caused by DML changes to the tables. This step of recovery is entered when DML (for example insert update, or delete) changes have been made to tables in a down AMP. While an AMP is down, the other AMPs in the cluster modify                                                                                                                        |



| Data                             | Description                                                                                                                                                                                                                                                                                    |
|----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                  | their fallback rows for the down AMP, and also modify their own rows for which the down AMP has fallback responsibility.                                                                                                                                                                       |
| Between Passes                   | The state of between passes arises since there is a five minute pause between passes. Primarily, this is intended to allow current operations, including any new work creating CJ and OJ entries, to finish and release their locks, so that recovery will not compete with online operations. |
| Miscellaneous OJ Processing      | Indicates that time is spent processing the various short running OJ entries, for example, releasing host utility locks.                                                                                                                                                                       |
| Down                             | Indicates that this AMP is not involved in the recovery process. However, statistics are maintained to indicate how much accumulated work remains to be recovered before the AMP can become operational.                                                                                       |
| Not currently executing recovery | Indicates that the AMP has not yet reached the point of starting Transaction Recovery or extracting OJ logs. The AMP is neither at the very early stage in recovery or has not yet started it. (The AMP might be down.)                                                                        |

### Example: Down AMP recovery status report

The Down AMP Recovery Status report generates the information shown in the following screen example.

```

DOWN AMP RECOVERY STATUS AT 12:27:25 99/10/13
AMP to be          Current Pass          Next Pass
caught up    Pass    OJ          CJ          OJ          CJ
-----
00001          0          0          0          107          16,531
- AMP Status: Not in recovery
- Down

```

## LIST STATUS *proc-id*

The LIST STATUS command with the *proc-id* option specified as mmmm provides additional detailed information about the recovery process of a specific AMP.

### Syntax

```
LIST STATUS [ proc-id ] ;
```

### Syntax Elements

#### STATUS

Transaction recovery information and AMP recovery information for unavailable AMPs.

- The displayed row count of a large sized table will be rounded to the nearest thousand.
- The recovery build records specify rebuild operations separately for each index of a table. This causes a rebuild of both the primary data and fallback data for that index, so the displayed sector count will differ from the total size for the table as derived using other methods.
- The sector count that is displayed is the sum of the estimated number of sectors for both primary and fallback rows to be copied over from the other AMPs in the cluster or recreated Non Unique Secondary Index (NUSI) indexes for all index subtables of the table which have an OJ build record. This includes the OJ build records of both the current pass and the next pass.

#### *proc-id*

Additional detailed information about the recovery process of a specific AMP including the list of tables needing to be rebuilt.

The LIST STATUS *proc-id* option is only allowed for processors that are in down AMP recovery (offline catchup) or the AMPs listed in the LIST STATUS display. No additional detail information is provided for AMPs that are in online catchup, since no build records exist for these AMPs.

### Usage Notes

The following additional information is reported:

- Status
- Pass number
- Current pass count
- Next pass count
- A list of tables that need to be rebuilt, which also includes the following information:
  - Total rows

- Total bytes
- Rebuild speed in bytes per second
- Estimated rebuild time

**Note:**

All tables that need to be rebuilt will display in the list.

- The tables for which an OJ build record exists
- An estimated sector count of that table

The LIST STATUS *proc-id* command is only allowed for processors that are in down AMP recovery (offline catchup), or the AMPs listed in the LIST STATUS display. No additional detail information is provided for AMPs that are in online catchup since no build records exist for these AMPs.

The following table shows the information contained in the columns under the TABLES TO BE REBUILT report.

| Data      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Row Count | The number of rows in the table to be rebuilt.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Status    | <p>Displays up to six possible states where the recovery process may be when LIST STATUS <i>proc-id</i> is executed.</p> <ul style="list-style-type: none"> <li>• <b>Blank</b><br/>Indicates that the table might be or will be rebuilt.</li> <li>• <b>MultiLoad Target Table in Apply</b><br/>Indicates that the table is a target table of a MultiLoad job. In this case the OJ build record is discarded since the likelihood is that the AMP will be online before the MultiLoad job finishes. If this is the case, MultiLoad will force a rebuild of the table while the AMP is online. If the MultiLoad job completes before the AMP is online, a new OJ build record is generated, meaning the rebuild will occur in a later pass.</li> <li>• <b>Locked</b><br/>Indicates that the table is currently locked by an EXCLUSIVE lock and that a valid sector count cannot be taken. Unless the lock is part of a drop table operation, the table will eventually be rebuilt in the next pass.</li> <li>• <b>Non-Fallbacked</b><br/>Indicates that a non-fallback table has been marked for a rebuild operation. Since all non-fallback tables are deleted during rebuild operation, the sector count associated with this table is meaningless.</li> <li>• <b>Table Rebuild</b><br/>Indicates that this OJ build record is for a table that is currently being rebuilt. This OJ build record will be discarded.</li> <li>• <b>Restore</b><br/>Indicates that a table may have an OJ build record for it, but it is currently the target of a (not necessarily active) restore operation. In this case, the OJ build record is discarded.</li> </ul> |
| Name      | The name of the table to be rebuilt.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

**Note:**

If the LIST STATUS *proc-id* command is executed while creating an index on a table that is to be rebuilt, then the time to rebuild the index is not added to the estimated rebuild time for that table.

**Example: Down AMP recovery status report for specified AMP**

This example shows a typical down AMP recovery status report, which displays the total rows, total bytes, and the rebuild speed in bytes per second.

```
DOWN AMP RECOVERY STATUS AT 12:27:25 99/10/13

AMP to be      Current Pass      Next Pass
caught up      Pass      OJ      CJ      OJ      CJ
-----
00001          0          0          0          107         16,531
- AMP Status: Not in recovery
- Down

TABLES TO BE REBUILT

Row Count      Status      Name
-----
      897,598
      897,598      "RESCRIBE"."Rescribe11"
      897,598      "RESCRIBE"."Rescribe12"
      897,598      "RESCRIBE"."Rescribe13"
      897,598      "RESCRIBE"."Rescribe14"
-----
3,590,392 total rows.
2,582,240KB total bytes.
      7,680KB is the rebuild speed in bytes per second.
The estimated rebuild time is 5.60 minutes.
```

## QUIT

The QUIT command exits rcvmanager.

### Syntax

```
QUIT ;
```

### Usage Notes

You can only type this command when rcvmanager is prompting for a command.

You cannot abort a rcvmanager command in progress.

## REBUILD/RECOVERY PRIORITY

The REBUILD/RECOVERY PRIORITY command sets or displays a priority level for use by the Table Rebuild utility and a Teradata system recovery operation.

### Syntax

```
{ REBUILD | RECOVERY } PRIORITY [ HIGH | MEDIUM | LOW ] ;
```

### Usage Notes

Setting a priority applies to future and currently running operations. The PRIORITY command takes the following forms.

Both priorities are independent of each other and can hold different values at any period of time. That is, recovery initiated rebuilds will use recovery priority and not rebuild priority. If you type the command without specifying high, medium, or low, the current priority setting is displayed.

The REBUILD PRIORITY command applies to any Table Rebuild started from the console, automatic table rebuild due to disk error recovery, and MLOAD rebuild of target tables for non-participant online AMPs.

The REBUILD PRIORITY command sets a priority for the rebuild utility. You can select HIGH, MEDIUM, or LOW level priority. If you do not explicitly set a priority, default rates exist and will be used. If you do not type a new priority, the current priority setting is displayed.

The RECOVERY PRIORITY command sets a priority for the Teradata system recovery operation. You can select HIGH, MEDIUM, OR LOW level priority. If you do not explicitly set a priority, the current priority setting is displayed. The priority settings are saved in the Recovery Status system table.

## ROLLBACK SESSION WORKLOAD

The ROLLBACK SESSION WORKLOAD command sets or displays the workload used by rollbacks for a specified host and session.

### Syntax

```
ROLLBACK SESSION host_id, session_id WORKLOAD [ workload_name ] ;
```

### Syntax Elements

#### *host\_id*, *session\_id*

Identifiers for the host and session for which you want Recovery Manager to display or set the workload for rollbacks.

The *session\_id* maximum is 4294967296.

The *session\_id* must be in the rollback list, that is, at least one of the tables in the output of LIST ROLLBACK TABLES must be from the session specified.

#### *workload\_name*

Name of the workload to apply to rollback operations in the specified session.

Vantage subsequently executes session rollbacks in the specified workload.

If you do not specify a *workload\_name*, Recovery manager displays the workload name for the current session.

### Usage Notes

By default, rollback processing runs in the workload of the current session.

The ROLLBACK SESSION WORKLOAD command allows you to change the priority of the rollback operation. By reducing the priority of less important rollbacks, you can increase the system resource availability for other processes.

This command affects the priority of the rollback process alone. It does not affect the workload or priority of the user session or other session work.

If the DBS Control General field RollbackPriority is set to TRUE, the rollback process runs in the workload of the aborted transaction that is being rolled back. The ROLLBACK SESSION WORKLOAD command can override the DBS Control Rollback Priority field setting, and change the priority of the rollback at any time.

The following usage rules apply to the ROLLBACK SESSION WORKLOAD command:

- The *host\_id* and the *session\_id* must be in the rollback tables list generated by the LIST ROLLBACK TABLES command from rcvmanager. This indicates that rollback is in progress in that session. You either can display or change the workload.

If you specify a *host\_id* or a *session\_id* that does not exist in the rollback list, rcvmanager displays an error message, and the command is ignored.

- If you do not specify a *workload\_name*, the Recovery Manager displays the workload name for the specified *host\_id* and *session\_id*.
- When you execute the ROLLBACK SESSION WORKLOAD command, an event is logged, even if the specified workload is the same as the current session workload. If you specify an invalid *workload\_name*, no event is logged.

The following example displays the event log entry after successful execution of the rcvmanager command.

```
> ROLLBACK SESSION 52, 1664 WORKLOAD WD-Rollback-Workload;  
ROLLBACK SESSION WORKLOAD command completed successfully.  
02/01/31 15:34:28 Rollback WORKLOAD for 52, 1664 changed to WD-Rollback-  
Workload;  
it was WD-Medium.
```



# Show Locks (showlocks)

---

**Note:**

The utility described is not available for as-a-service customers.

---

The Show Locks utility, `showlocks`, provides information about Host Utility (HUT) locks placed on databases and tables by the Archive/Recovery (ARC) utility during database operations.

These locks might interfere with application processing and should be released after utility processing is complete. Show Locks also displays information about locks placed during an Advanced SQL Engine system failure.

Lock information can also be monitored and displayed using the Lock Viewer portlet in Teradata Viewpoint. For more information, see *Teradata® Viewpoint User Guide*, B035-2206.

## Runs From

- Database Window or comparable interface to the Teradata console subsystem, such as `cnstern`
- Teradata Viewpoint Remote Console portlet
- Host Utility Console
- Linux command line

For general information on starting the utilities from different interfaces, see [Starting the Utilities](#). For information on Viewpoint, see *Teradata® Viewpoint User Guide*, B035-2206.

## Host Utility Locks

You can release HUT locks either by submitting a separate `RELEASE LOCK` SQL command, or by using the `RELEASE LOCK` option of the appropriate command. For example, `ARCHIVE`, `ROLLBACK`, `RESTORE`, `BUILD`, and `ROLLFORWARD`.

HUT locks placed by the Archive/Recovery utility differ from locks placed by other operations or transactions. HUT locks have the following characteristics:

- Archive/Recovery (ARC utility) HUT locks are associated with the currently logged-on user who entered the command, rather than with a batch job or transaction. Data Stream Architecture (DSA) HUT locks are placed at the job level.
- An ARC operation can place HUT locks on all AMPs, a single AMP, or a group of AMPs. DSA operations always place all-AMP HUT locks.
- HUT locks for ARC that are placed at one level of an object never conflict with a utility lock at another level that was placed on the same object for the same user. HUT locks for DSA are always placed at the job level. The locking modes and levels are applied as follows:

- A Read lock is placed on an object being dumped.
- Locks are placed at the cluster level during a CLUSTER dump.
- If a table being dumped is defined for an after-image permanent journal (and the appropriate option was selected on the DUMP command), a group Read lock is placed on the table rows.
- A Write lock is placed on all tables involved in ROLLFORWARD and ROLLBACKWARD recovery operations.
- A Write lock is placed on a journal table that is being deleted.
- A Write lock is placed on a permanent journal table that is being restored.
- An Exclusive lock is placed on any object being restored that is not a journal table.
- DSA does not preserve permanent journal settings for a table. After the table is restored using DSA, there will be no permanent journaling for the table.
- Archive/Recovery locks remain active until you release them.

**Note:**

If Archive/Recovery locks are not specifically released, they are automatically reinstated following a database or client system restart.

## Interpreting the Show Locks Display

The Show Locks utility display provides this information:

- Summary of the showlocks function
- Name of the databases, tables, and hash IDs on which locks are placed
- Username that placed each lock
- Lock mode: read, write, exclusive, or access
- Identification numbers of the AMPs on which the locked database or table resides
- Job identifier (only for DSA archive/recovery operations)
- Proxy table locks, which enable sequential locking on all-AMP table locks

Show Locks reports All AMPs rather than individual AMP numbers when the locked database or table resides on all AMPs. Information is provided for only the most restrictive lock a user has placed on an object.

When no locks are found, Showlocks reports this message:

```
There are currently no host utility locks in the DBS.
```

### Example: Show Locks Output with DSA Database-Level HUT Lock

```
EMPLOYEE_SW_ZONE
USER DBC    MODE Read      AMP All Amps JobId: 1300542
```

**Example: Show Locks Output with DSA Object-Level HUT Lock**

```

EMPLOYEE_SW_ZONE.Address
  USER DBC  MODE Read      AMP All Amps JobId: 3400241
EMPLOYEE_SW_ZONE.Phone
  USER DBC  MODE Read      AMP 1          JobId: 3400241  PROXY LOCK

```

## Conflicts

If a host utility lock that conflicts with Show Locks is in place when showlocks is executed, the Teradata system displays this message:

```
'Unable to proceed due to xxxx lock on yyyy'
```

Where xxxx refers to either a Write or Exclusive lock, and yyyy refers to DBC, DBC.TVM, or DBC.DBase. After reporting a conflict, Show Locks terminates.

# Table Rebuild (rebuild)

---

**Note:**

The utility described is not available for as-a-service customers.

---

The Table Rebuild utility, rebuild, recovers data tables (primary, fallback, or both) on specified AMPs.

Normally, Advanced SQL Engine utilizes fallback data, if available, to recover offline AMPs automatically during startup. However, occasionally, primary or fallback data can be corrupted by software errors or other abnormal conditions. Table Rebuild allows on-demand recovery of data that is suspected of being corrupt.

## Runs From

Table Rebuild runs from Database Window or comparable interface to the Teradata console subsystem, such as cnstern.

For general information on starting the utilities from different interfaces, see [Starting the Utilities](#).

## Rebuilding Tables

Table Rebuild can recover primary data, fallback data, or both primary and fallback data that is stored on an AMP. You can rebuild a single table, all tables in a specified database, or all tables on the AMP. As an option, the rebuild process can be limited to only those tables that have fallback protection.

---

**Note:**

On Teradata platforms optimized for fallback, all user tables have fallback. On platforms not optimized for fallback, you have the option to create tables with or without fallback.

---

The rebuild process uses fallback data to recover primary data, and primary data to recover fallback data on a specified AMP. The current, presumably corrupted contents of the table being rebuilt are deleted and replaced with the fallback or primary data, as appropriate. For tables with no fallback protection, Table Rebuild restores an empty table that retains only the table header information. Using the FALLBACK TABLES option ensures that only tables with fallback protection are rebuilt. See [REBUILD AMP FALLBACK TABLES](#).

---

**Note:**

Tables that are marked down at the time they are rebuilt will remain down after the rebuild. To clear the down status, use the ALTER TABLE ... RESET DOWN statement after the table is rebuilt.

---

Global temporary tables, volatile tables, join indexes, and hash indexes cannot be rebuilt. For indexes that may be corrupted, drop the index and recreate it as needed.

Certain non-fallback system tables, such as DBC.DataBaseSpace and DBC.Acctg, store information that is unique to each AMP. These tables maintain information such as the current database space utilization, CPU utilization, and I/O statistics for the AMP. Instead these tables are updated by the system automatically after the rebuilt AMP is placed back on line.

The data for foreign tables is external to Vantage, so for these tables, only the table headers are rebuilt. For more information about foreign tables, see CREATE FOREIGN TABLE in *Teradata Vantage™ - SQL Data Definition Language Syntax and Examples*, B035-1144.

If a database restart interrupts an ALL TABLES rebuild process, it can be restarted again, and will continue from where the rebuild process was interrupted. For more information, see [RESTART REBUILD](#).

## Locking During Rebuilds

Table Rebuild can apply one of three types of read locks while it is rebuilding tables:

- Database-level read locks are placed on the database on the AMP used as the source of data for rebuilding the corrupted tables. This is the default type of lock that Table Rebuild applies.
- Table-level read locks are placed on the individual tables that are the sources of data for rebuilding the corrupted tables.
- Rowrange-level read locks consist of selected groups of row-only locks. This type of lock allows concurrent updates of the tables being used as the sources of data for rebuilding the corrupted tables.

In the case of rebuilding primary data, these locks are placed on the AMPs containing the corresponding fallback data. In the case of rebuilding fallback data, the locks are placed on the AMPs containing the corresponding primary data.

Whether the database, table, or rowrange option is specified, Table Rebuild requests a read lock on all online AMPs which belong to the same cluster for the database or table being rebuilt. These AMPs contain the valid data that will be copied back to the AMPs being rebuilt. Rowrange-level locks are applied locally to the AMP that is being rebuilt, so that the valid data can be copied back to the rebuilt AMP.

For database and table rebuilds, processing begins after the lock is acquired. For rowrange rebuilds, the table-level lock is changed to a rowrange lock before processing begins.

If an AMP is online while all data or primary data is being rebuilt, an exclusive lock is placed on the database or table. If an AMP is online while fallback data is being rebuilt, a write lock is placed on the database or table. If the AMP is offline, a read lock is used in all cases.

If all tables on an AMP are being rebuilt, Table Rebuild attempts to set database-level read locks. If a database lock fails because of a conflict with another lock, that database is bypassed. Table Rebuild attempts to rebuild the database again after all accessible databases and tables are processed. Table Rebuild successively runs through the list of bypassed databases to attempt to get a lock. If the bypass list contains a single database, or Table Rebuild has tried the list of bypassed databases ten times, rebuild processing stops and waits until a lock can be acquired.

If Table Rebuild is running in the background, a message is displayed on the system console identifying the database and stating that the utility is waiting for a lock.

## Table Rebuild Utility Commands

Table Rebuild presents a command-line environment that allows the entry of the following commands.

The commands are discussed in detail in the sections that follow.

## REBUILD AMP

Rebuilds data on a specified AMP. The rebuild operation can include:

- All tables on the AMP
- For a specified database: Primary, fallback, or both types of data in a specified database
- For a specified table: Primary, fallback, or both types of data, down regions, table header, or a specific range of rows

### Syntax

```
REBUILD AMP nnnn {
  { ALL TABLES ALL DATA |

    { dbase | dbase.tbl } { ALL DATA | PRIMARY DATA | FALLBACK DATA }
  }
  [, rebuild_amp_spec [, ...] ] |

  dbase.tbl { DOWN REGION | TABLE HEADER | row_range } [ lock_wait_minutes ]
} ;
```

#### *rebuild\_amp\_spec*

```
{ LOG INTO logdbase.logtbl |
  NO LOCK [ ON NO FALLBACK TABLES ] |
  WITH { DATABASE | TABLE | ROWRANGE } LOCK |
  [ n TABLES ] IN PARALLEL
}
```

#### *row\_range*

```
SUBTABLE subtable_id ROWRANGE start_rowid end_rowid [ AUTOADJUSTBLOCKS ]
```

### Syntax Elements

#### *nnnn*

The number of the AMP that is to be rebuilt.

#### ALL TABLES

Specifies that all table data stored on the AMP, whether primary or fallback data, is to be rebuilt.

Tables with fallback protection are fully recovered. Tables without fallback protection are left empty on the AMP. All the data in the permanent journals will be recovered, except for journal rows for non-fallback tables without the dual journaling option.

The AMP to be rebuilt must be in the UTILITY vproc state and running DBS partitions. All the other AMPs in the cluster must be on-line.

For more information, see [Usage Notes](#).

### ***dbase***

The name of a database. All tables (including stored procedures) in this database will be rebuilt on the specified AMP. Tables with fallback protection are fully recovered. Tables without fallback protection are left empty on the AMP. If the database contains a permanent journal, the journal is left unchanged.

If the DBC database is specified, the specified AMP must be off line. For other databases, the AMP may be on line or off line during the rebuild.

---

#### **Note:**

Join indexes and hash indexes are skipped and not rebuilt. This is true whether or not the join index has fallback.

---

### ***dbase.tbl***

The name of a table, stored procedure, UDF, or UDM that is to be rebuilt. The AMP on which the table or stored procedure will be rebuilt can be either online or offline.

If the specified table is fallback protected, the appropriate data is recovered. If the table is not fallback protected, the table is left empty if ALL or PRIMARY DATA rebuild was selected. You cannot rebuild fallback data for a table that has no fallback protection.

### **ALL DATA**

Specifies that both primary data and fallback data tables stored on the AMP should be rebuilt. recommended, unless you are certain that only primary or only fallback data on the AMP has been corrupted.

### **PRIMARY DATA**

Specifies that only the primary data tables stored on the AMP should be rebuilt.

Use the ALL DATA option unless you are certain that only primary data on the AMP has been corrupted.

### **FALLBACK DATA**

Specifies that only the fallback data tables stored on the AMP should be rebuilt.



Use the ALL DATA option unless you are certain that only fallback data on the AMP has been corrupted.

## DOWN REGION

To rebuild only the down regions of the specified table.

Down regions are ranges of rows for which Vantage has detected file system errors. The rows are marked as down, and their data can be rebuilt from fallback copies.

## TABLE HEADER

To rebuild the table header on the specified AMP. At least one AMP in the system must be online. Information in fields 4, 6, and 10 of the table header row may be lost as a result of the rebuild. For more information on these fields, see *Teradata Vantage™ - Database Design*, B035-1094.

## SUBTABLE *subtable\_id*

To rebuild a specific subtable.

*subtable\_id* is a non-zero 16-bit number in hexadecimal format that identifies the subtable to be rebuilt.

For example, 400 represents the primary data subtable of the specified table, and 800 represents the fallback data subtable for the primary data.

## ROWRANGE *start\_rowid end\_rowid*

Rebuilds the specified range of rows.

The format for a row ID comprises five hexadecimal values:

- partition number: a 64-bit number in hexadecimal format, or 0 for a non-partitioned table
- hash0: 16-bit hexadecimal number
- hash1: 16-bit hexadecimal number
- unique0: a 16-bit hexadecimal number
- unique1: a 16-bit hexadecimal number

Each hexadecimal value can optionally include an H suffix.

Examples of valid row IDs:

0H B334 4BFA 00 01

0 B334H 4BFA 00H 01

0H B334H 4BFAH 0H 1H

0 B334 4BFA 0 1

**AUTOADJUSTBLOCKS**

Specifies that Table Rebuild should automatically adjust the specified row range to include complete blocks if the specified row range starts or ends in the middle of a bad block.

If this option is not specified and a row range starts or ends in the middle of a bad data block, REBUILD returns an error.

***lock\_wait\_minutes***

The amount of time in minutes that Table Rebuild should wait to obtain a table lock. The default is zero, which means REBUILD waits indefinitely.

**LOG INTO *logdbase.logtbl***

Specifies that Table Rebuild is to run in the quiet mode or background mode. All messages will be written to the system console and to a user-defined table. The table is specified by database name and table name. For more information, see [Usage Notes](#).

**NO LOCK [ON NO FALLBACK TABLES]**

Specifies that no lock should be applied to any non-fallback tables being rebuilt. (non-fallback tables are tables that were created without fallback protection.)

By default, tables that do not have fallback protection are flagged in their table headers as being in the process of being rebuilt. (Field 4 of the table header row contains 'rebuild in progress'.) This causes locks to be applied which limit the operations that are allowed on these tables. As long as these tables are flagged, they cannot be dropped or restored, and the rebuild cannot be rerun on them.

The only ways to remove the flag is by one of the following:

- rebuild the table with the NO LOCK option
- drop the table
- restore the table

The NO LOCK option prevents the flagging and locking of these tables,. It should be used when access to the tables is no longer important. Rebuilding non-fallback tables causes their contents to be deleted.

**Note:**

ON NO FALLBACK TABLES has no effect on this option, but optionally may be entered for additional console clarity.

**WITH DATABASE LOCK**

Specifies that a database-level read lock will be placed on the source AMP database data used to rebuild each corrupted table. This is the default lock setting.

**Note:**

This option is valid only with the ALL TABLES option.

**WITH TABLE LOCK**

Specifies that a table-level read lock will be placed on the source AMP table to be used to rebuild the corrupted table.

**Note:**

This option is valid only with the ALL TABLES option.

**WITH ROWRANGE LOCK**

Specifies that a rolling-read lock (selected groups of row-only locks) will be placed on the source AMP table used to rebuild the corrupted table. This lock allows concurrent updates of the tables being used on the source AMP for the rebuild.

For a column-partitioned table, a full table lock is used.

**Note:**

This option is valid only with the ALL TABLES option.

**[*n* TABLES] IN PARALLEL**

Specifies that during all-table rebuilds (rebuilds using ALL TABLES), or during fallback table rebuilds (using FALLBACK TABLES), or during database rebuilds (using the *dbase* options described above) multiple tables can be rebuilt in parallel. This can make the rebuild operations complete more quickly. From 2 to 255 tables can be rebuilt simultaneously.

*n* is an integer from 2 to 255 that specifies how many tables will be rebuilt in parallel. If *n*TABLES is not specified, the default number of tables that will be rebuilt in parallel is six.

If Vantage is reset during an ALL TABLES ALL DATA rebuild, when the rebuild process is restarted (see [RESTART REBUILD](#)), the rebuild preserves the IN PARALLEL setting, and continues rebuilding tables in parallel.

**Note:**

If the IN PARALLEL option is used together with the ALL TABLES, FALLBACK TABLES, or *dbase* options, the database will be locked during the entire duration of the parallel rebuild.

## Usage Notes

- Regardless of whether the AMP to be rebuilt is offline or online during Table Rebuild, all other AMPs in the same cluster must be online. AMPs in other clusters may be offline.
- Tables that use sparse maps (or contiguous maps that include only a subset of all AMPs in the system) are not necessarily distributed to all AMPs of the system. If a table specified for rebuilding does not exist on an AMP specified in the REBUILD command, Table Rebuild issues the following message:

```
Unable to rebuild table tablename due to table not in rebuilt AMP
```

For more information on contiguous and sparse maps, see *Teradata Vantage™ - Database Design*, B035-1094.

- Vantage can isolate some file system errors to a specific data or index subtable, or to a contiguous range of rows ("region") in a data or index subtable. In these cases, Vantage marks only the affected subtable or region down. This improves system performance and availability by allowing transactions that do not require access to the down subtable or rows to proceed, without causing a database crash that would require a system restart.

The normal rebuild process removes down-region information from the table header.

## Before Rebuilding All Tables

Before rebuilding all tables, do the following:

1. Use the Vproc Manager utility to set the VprocState of the AMP that will be rebuilt to FATAL.
2. Restart the database.

---

### Note:

This step is not necessary if the rebuilding AMP state was FATAL before the last database restart.

---

3. Use the Vproc Manager utility to boot the AMP that will be rebuilt.

Messages will be displayed on the system console to indicate the status of the boot. If the boot is successful, this AMP is ready for an ALL TABLES rebuild.

---

### Note:

The BOOT command will re-initialize the disk of the AMP in anticipation of all-tables table rebuild and start the DBS partitions on the specified AMP. This applies only to vprocs with a VprocState of FATAL and a ConfigStatus of Down. A confirmation input is necessary to process the initialization.

---

Valid Vproclds are decimal numbers in the range of 0 through either 30719 or 16383, depending on the system.

Hex numbers can also be specified by appending a trailing “x” (for example, 0x, 3FFx).

4. Start Table Rebuild and run an ALL TABLES rebuild on this AMP.
5. When the rebuild is done, use the Vproc Manager utility to set the VprocState of this AMP to ONLINE.

---

**Note:**

Table Rebuild automatically sets the VprocState of this AMP from UTILITY to OFFLINE when complete.

---

6. Restart the database.

## Rebuilding One or All Tables

When all data on an AMP has been lost and needs to be rebuilt or the database DBC on that AMP needs to be rebuilt, Table Rebuild can recover this information. While database DBC is being rebuilt, the AMP whose data needs to be rebuilt must be offline. By contrast, when any other database on the AMP or a single table is being rebuilt, the AMP can be either online or offline.

After all databases are rebuilt, Vantage must be restarted to update the rebuilt tables and to return the AMP whose data has been rebuilt to online operation.

## Running Table Rebuild in the Background

When you specify the LOG INTO logdbase.logtbl option, Table Rebuild runs as a background task. You can run multiple Table Rebuild operations both in the background and foreground (interactive mode) at the same time. Completion messages for background rebuilds are sent to the system console and to the user-defined table specified in the LOG INTO option.

The table specified in the LOG INTO option must have been created previously as follows:

```
CREATE SET TABLE logDB.LogTbl, FALLBACK
( MsgDate CHAR(8),      /* format: 'yy/mm/dd' */
  MsgTime CHAR(8),      /* format: 'hh:mm:ss' */
  MsgAMP  CHAR(6),      /* format: 'nnnn' */
  MsgCode CHAR(1),      /* see below */
  MsgText VARCHAR(600) CHARACTER SET UNICODE) /* message text */
PRIMARY INDEX (MsgDate, MsgTime);
```

| Column                 | Contents                                                                                                                               |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| MsgDate<br>and MsgTime | The system date and time when the message was generated. Together these columns comprise a non-unique primary index for the log table. |
| MsgAMP                 | The four-digit vproc number of the rebuilding AMP.                                                                                     |
| MsgCode                | A single character code indicating the type of rebuild message. See below.                                                             |

| Column  | Contents                         |
|---------|----------------------------------|
| MsgText | The text of the rebuild message. |

MsgCode is one of the following values:

| Value | Meaning                                              |
|-------|------------------------------------------------------|
| ' '   | A normal message                                     |
| 'D'   | Rebuilding database message                          |
| 'E'   | Error message                                        |
| 'J'   | Rebuilding table message for a journal               |
| 'N'   | Rebuilding table message for a no-fallback table     |
| 'R'   | Rebuilding table message for tables used by recovery |
| 'S'   | Start/Restart rebuild operation                      |

---

**Note:**

The table-level messages do not include the database names. The reports should include all the 'D' class messages and be ordered by date and time for proper identification.

---

## REBUILD AMP FALLBACK TABLES

Rebuilds only tables with fallback protection, including stored procedure tables. Because Table Rebuild deletes all tables selected for rebuild before the start of the rebuild process, this command prevents tables without fallback protection from being deleted.

This command performs most of the work which is done by normal down AMP recovery, making the actual restart to bring a down AMP back on line as fast as possible, while guaranteeing data integrity. The AMP to be rebuilt must be off line and all of the other AMPs in the cluster must be on line.

### Syntax

```
REBUILD AMP nnnn FALLBACK TABLES [ , rebuild_amp_spec [ , ... ] ] ;
```

#### *rebuild\_amp\_spec*

```
{ LOG INTO logdbase.logtbl |  
  WITH { DATABASE | TABLE | ROWRANGE } LOCK |  
  [ n TABLES ] IN PARALLEL  
}
```

### Syntax Elements

#### *nnnn*

The number of the AMP vproc to be rebuilt.

#### FALLBACK TABLES

Specifies that only tables with fallback protection will be rebuilt. This prevents tables without fallback protection from being deleted. The first step that Table Rebuild usually performs is to delete the contents of the table that Table Rebuild is rebuilding.

#### LOG INTO *logdbase.logtbl*

Specifies that Table Rebuild is to run in the quiet mode or background mode. All messages will be written to the system console and to a user-defined table. The table is specified by database name and table name. For more information about running Table Rebuild in the background, see [REBUILD AMP](#).

#### WITH DATABASE LOCK

Specifies that a database-level read lock will be placed on the source AMP database data used to rebuild the corrupted table. This is the default lock setting.

**WITH TABLE LOCK**

Specifies that a table-level read lock will be placed on the source AMP table to be used to rebuild the corrupted table.

**WITH ROWRANGE LOCK**

Specifies that a rolling-read lock (selected groups of row-only locks) will be placed on the source AMP table used to rebuild the corrupted table. This lock allows concurrent updates of the tables being used on the source AMP for the rebuild.

**Note:**

For a column-partitioned table, a full table lock is used.

**[*n* TABLES] IN PARALLEL**

Specifies that multiple tables can be rebuilt in parallel. This can make the rebuild operations complete more quickly. From 2 to 255 tables can be rebuilt simultaneously.

*n* is an integer from 2 to 255 that specifies how many tables will be rebuilt in parallel. If *n*TABLES is not specified, the default number of tables that will be rebuilt in parallel is six.

**Usage Notes**

- Regardless of whether the AMP to be rebuilt is offline or online during Table Rebuild, all other AMPs in the same cluster must be online. AMPs in other clusters may be offline.
- Tables that use sparse maps (or contiguous maps that include only a subset of all AMPs in the system) are not necessarily distributed to all AMPs of the system. If a table specified for rebuilding does not exist on an AMP specified in the REBUILD command, Table Rebuild issues the following message:

```
Unable to rebuild table tablename due to table not in rebuilt AMP
```

For more information on contiguous and sparse maps, see *Teradata Vantage™ - Database Design*, B035-1094.

- Vantage can isolate some file system errors to a specific data or index subtable, or to a contiguous range of rows ("region") in a data or index subtable. In these cases, Vantage marks only the affected subtable or region down. This improves system performance and availability by allowing transactions that do not require access to the down subtable or rows to proceed, without causing a database crash that would require a system restart.

The normal rebuild process removes down-region information from the table header.



## Running Table Rebuild in the Background

When you specify the LOG INTO logdbase.logtbl option, Table Rebuild runs as a background task. You can run multiple Table Rebuild operations both in the background and foreground (interactive mode) at the same time. Completion messages for background rebuilds are sent to the system console and to the user-defined table specified in the LOG INTO option.

The table specified in the LOG INTO option must have been created previously as follows:

```
CREATE SET TABLE logDB.LogTbl, FALLBACK
( MsgDate CHAR(8),      /* format: 'yy/mm/dd' */
  MsgTime CHAR(8),      /* format: 'hh:mm:ss' */
  MsgAMP CHAR(6),       /* format: 'nnnn' */
  MsgCode CHAR(1),      /* see below */
  MsgText VARCHAR(600) CHARACTER SET UNICODE) /* message text */
PRIMARY INDEX (MsgDate, MsgTime);
```

| Column              | Contents                                                                                                                               |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| MsgDate and MsgTime | The system date and time when the message was generated. Together these columns comprise a non-unique primary index for the log table. |
| MsgAMP              | The four-digit vproc number of the rebuilding AMP.                                                                                     |
| MsgCode             | A single character code indicating the type of rebuild message. See below.                                                             |
| MsgText             | The text of the rebuild message.                                                                                                       |

MsgCode is one of the following values:

| Value | Meaning                                              |
|-------|------------------------------------------------------|
| ' '   | A normal message                                     |
| 'D'   | Rebuilding database message                          |
| 'E'   | Error message                                        |
| 'J'   | Rebuilding table message for a journal               |
| 'N'   | Rebuilding table message for a no-fallback table     |
| 'R'   | Rebuilding table message for tables used by recovery |
| 'S'   | Start/Restart rebuild operation                      |

### Note:

The table-level messages do not include the database names. The reports should include all the 'D' class messages and be ordered by date and time for proper identification.



## RESTART REBUILD

Restarts an all-table rebuild operation (REBUILD command with the ALL TABLES options specified).

If a database failure occurs while a Table Rebuild is running for all tables, this command restarts the rebuild process, bypassing databases and tables already rebuilt, and continuing from the table that was being rebuilt when the system failed.

### Syntax

```
RESTART REBUILD ON AMP nnnn ;
```

### Syntax Elements

*nnnn*

The vproc number of the AMP which was previously running the ALL TABLES rebuild.

### Usage Notes

The RESTART REBUILD command restarts an all-table rebuild that was terminated by a system restart or failure. After RESTART REBUILD is executed, no intervention is required.

Table Rebuild automatically determines where the rebuild terminated and restarts at the appropriate place. Table Rebuild determines where to restart by using the spool files that contain the database names and table names to be rebuilt (in order by database ID and by table ID within the database). Before each subtable (primary or fallback) is rebuilt, Table Rebuild writes a restart record to the current DBC table SysRcvStatJournal.

When you start a RESTART REBUILD, the program reads the restart record and positions itself in the spool file accordingly. After completion of the Table Rebuild, the restart row is deleted from the journal table.

## Table Rebuild Usage Notes

### General

Normally, Table Rebuild is run interactively, as a foreground process started from a Database Window application window. Output from the program is directed to the window. Optionally, Table Rebuild can be run as a background process. Output from the program is logged to a special table, rather than being displayed on screen. All background rebuild operations continue to run to completion, even after Table Rebuild is quit. For more information, see the LOG INTO option described for [REBUILD AMP](#) and [REBUILD AMP FALLBACK TABLES](#).

Several rebuild operations, both foreground and background rebuilds, can be run simultaneously. Up to four interactive foreground sessions can utilize the four application windows available in Database Window. Each session must rebuild tables on AMPs in different clusters.

Any number of background process rebuilds can run simultaneously, and can be launched from the same application window. Each rebuild process must rebuild tables on an AMP in different clusters.

## Error Handling

Table Rebuild issues messages interactively.

The following message might be returned by Table Rebuild after processing is complete:

```
Bad table header on AMP ccc-p for table tablename
```

This message can be caused by one of the following conditions:

- The table header does not exist. This might be because the table was dropped after Table Rebuild was started.
- A header was found, but the table was rebuilt unsuccessfully on the identified AMP.

To recover the table, restore it from the latest archive. If restoring from an archive is not an option, contact the Teradata Support Center.

When a table is marked as being in an invalid state and not redistributable during a system reconfiguration, there is no guarantee that the rows in the table are where they should be. Table Rebuild cannot rebuild the table, and displays “Unable to rebuild due to pending Reconfig Abort of table *tablename*.” In these cases, before rebuilding the table you should first recover these tables by restoring them from the latest archive, or drop these tables.

Tables that use sparse maps (or contiguous maps that include only a subset of all AMPs in the system) are not necessarily distributed to all AMPs of the system. If a table specified for rebuilding does not exist on an AMP specified in the REBUILD command, Table Rebuild issues the following message:

```
Unable to rebuild table tablename due to table not in rebuilt AMP
```

For more information on contiguous and sparse maps, see *Teradata Vantage™ - Database Design*, B035-1094.

For more information on error messages involving AMP operations, see *Teradata Vantage™ - Database Messages*, B035-1096.

## Getting Help

To get help information about the Table Rebuild utility, press the F7 key while in the Database Window application window. A set of second-level function keys will display as shown below:

### Help

The Table Rebuild utility allows you to recover the data tables on a specified AMP's disk(s) by copying the fallback copy maintained by the other AMPs in the cluster. You may rebuild all or part of the tables on an AMP via the options listed below.>

```
<F2> - Rebuild all tables on an AMP  
<F3> - Rebuild all data tables in a database  
<F4> - Rebuild a single table  
<F5> - Rebuild fallback protected tables only  
<F6> - Rebuild a previous rebuild after a system restart  
<F7> - General syntax
```

For information on the subjects listed on the screen, press the corresponding function key. To return to the next-higher menu level, press the **F8** key. To exit the help system and return to the Table Rebuild command prompt, press **F8** from the top-level menu.

# Teradata Locale Definition Utility (tdlocaledef)

---

## Note:

The utility described is not available for as-a-service customers.

---

The Teradata Locale Definition utility, `tdlocaledef`, is a command-line utility that allows you to define or change how Vantage formats numeric, date, time, and currency output.

`tdlocaledef` converts a specification for data formatting (SDF) text file into a Teradata globally distributed object (GDO), an internal binary format file that stores configuration information. The GDO is made available simultaneously to all nodes of an MPP system. The utility can also convert the text file to a local, non-distributed, binary file, that can be converted back to text in order to ensure the formatting syntax is valid.

---

## Note:

Format changes take effect only after the database is restarted, and do not affect columns that were created prior to the restart.

---

## Runs From

`Tdlocaledef` runs from the Linux command line.

For general information on starting the utilities from different interfaces, see [Starting the Utilities](#).

## Syntax

```
tdlocaledef {
  [ -input filename ] [ -output { filename | new } ] |
  -reverse { current | filename } [ -source filename ]
}
```

## Syntax Elements

### -input *filename*

Specifies the text file containing the SDF settings.

If *filename* does not include the full path to the SDF file, `tdlocaledef` assumes the named file is in the current directory.

If the `-input` option is not specified, `tdlocaledef` looks for a `tdlocaledef.txt` SDF file located in `/etc/opt/teradata/tdconfig`.

By default there is no `tdlocaledef.txt` file in those locations. Teradata provides a sample file in `/usr/tdbms/etc`.

Teradata recommends that you never modify the sample `tdlocaledef.txt` file directly, but instead copy the file to another location and make changes to the copy.

For more information on customizing Vantage output formatting settings, see [SDF File](#).

### **-output *filename***

#### **-output new**

Specifies what happens to the compiled, binary version of the SDF formatting settings that is produced by `tdlocaledef`.

- *filename* specifies that the compiled settings be stored only in a local file. This file is suitable for use with the `-reverse` option to verify that the syntax used in the SDF text file is correct. If no path is specified, the binary file is placed in the current directory.
- `new` specifies that the settings become effective at the next database restart. The settings are compiled into a GDO, and used by all nodes of the system. A copy of the SDF text file is stored on every node in `/etc/opt/teradata/tdconfig`.

If the `-output` option is not specified, `tdlocaledef` creates a local binary file named `tdlocaledef.loc` in `/etc/opt/teradata/tdconfig`.

### **-reverse current**

#### **-reverse *filename***

Specifies that the compiled output format settings currently in effect or stored in a local binary file be written to a text file suitable for viewing and editing.

This option is primarily used for creating an editable text file of the current output formatting settings that can be used to customize or change the output formatting that Vantage uses for dates, times, numbers, and currency.

`-reverse` can also be used to verify correct syntax in an edited SDF text file.

- `current` specifies that the output formatting settings currently in effect be written to a text file.
- *filename* specifies that the SDF settings in a local compiled binary file be converted to a text file. Use this option to ensure that a previously compiled SDF text file uses proper syntax. If the original text file does not match the file produced using the `-reverse filename` option, there is a syntax error in the original SDF file.

### **-source *filename***

Specifies the SDF text file `tdlocaledef` should create based on the specified `-reverse` options. If `-source` is not specified, `tdlocaledef` creates an SDF file named `tdlocaledef.txt` in `/etc/opt/teradata/tdconfig`.

## SDF File

The SDF file is a text file that defines how Vantage formats numeric, date, time, and currency output. The formatting strings it contains are used in conjunction with special formatting characters used in SQL FORMAT output phrases.

Vantage includes a default SDF file that contains a viewable and editable text version of the default output formatting settings. The default SDF file is located in `/usr/tdbms/etc`.

The SDF file controls how the following kinds of information are formatted in the output of Vantage.

- Day names
- Month names
- AM and PM names
- Numeric and currency separators
- Numeric and currency digit grouping rules
- Currency symbols
- Default display formats for data types

The SDF file also controls the default display formats for the following data types:

- BYTEINT
- SMALLINT
- BIGINT
- INTEGER
- NUMERIC (includes DECIMAL)
- REAL (includes DOUBLE PRECISION and FLOAT)
- DATE
- TIME and TIME WITH TIME ZONE
- TIMESTAMP and TIMESTAMP WITH TIME ZONE
- NUMBER

The SDF file also can be used to specify a time zone string, which identifies time zone the server will use, and can specify rules for when Daylight Saving Time begins and ends for time zones that observe DST.

---

### Note:

You cannot use the SDF file to control output formatting for the INTERVAL data type.

---

The formatting characters permitted in the format strings of the SDF file are the same formatting characters permitted in an SQL FORMAT for the data types listed above. You can override the default display format for a data type in the SDF file by using the FORMAT output phrase in a CREATE TABLE, ALTER TABLE, or SELECT statement in SQL.

SDF quoted strings are delimited by the quotation mark (U+0022) and can specify almost any Unicode character accepted by Vantage. However note the following restrictions:



- Characters within SDF quoted strings are limited to the printable 7-bit ASCII characters (U+0020, U+0021 and U+0023 through U+007E). The quotation mark itself (U+0022) cannot be used within an SDF quoted string.
- The reverse solidus (\) character (U+005C) does not represent itself, but instead can be used to specify Unicode characters above U+009F. The sequence \u XXXX specifies the corresponding Unicode character U+ XXXX, where XXXX is the hexadecimal for the Unicode character.

Formats are restricted to valid formats as specified in *Teradata Vantage™ - Data Types and Literals*, B035-1143.

For example, you can set the value of the currency string in the SDF file to the currency symbol native to your locale. To include that currency symbol when you display numeric monetary information in an SQL SELECT statement, use the L formatting character in the FORMAT output phrase. For more information on display formats and SQL output format phrases, see *Teradata Vantage™ - Data Types and Literals*, B035-1143.

Additionally, the SDF file can be used to specify a time zone string, which allows Vantage to automatically adjust the system time for locales that use Daylight Saving Time.

## SDF Elements

The SDF file uses a simple format where each entry is an SDF element that specifies the output formatting for one kind of data, or specifies a time zone string. The generic format of an SDF element is:

```
KEYWORD {value_list}
```

where *value\_list* is one or more values delimited by double-quote characters and separated by semicolons.

Keywords are not case sensitive, but values are, unless stated otherwise. The SDF format strings consist of printable characters from 7-bit ASCII, unless an element is restricted further by the type of data. For out-of-range characters, use the \u Unicode hex notation. The SDF can contain single-line comments that start with //.

### Note:

All elements must be specified once in the SDF file, so it is convenient to start from an existing SDF file and edit to suit your needs.

The following table described the individual SDF elements.

| SDF Element Keyword | Value Description                                                                                                                                                                                                                                           |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ShortDays           | <p>Seven abbreviated names for the days of the week in the native language. These names can vary in length. There must be seven values entered for this keyword.</p> <p>Example:</p> <pre>ShortDays {"Sun"; "Mon"; "Tue"; "Wed"; "Thu"; "Fri"; "Sat"}</pre> |

| SDF Element Keyword | Value Description                                                                                                                                                                                                                                                                                                                                                                                        |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| LongDays            | <p>Seven full names for the days of the week in the native language. The order of the names must match the order of the names of the ShortDays. These names can vary in length. There must be seven values entered for this keyword.</p> <p>Example:</p> <pre>LongDays {"Sunday";"Monday";"Tuesday"; "Wednesday";"Thursday";"Friday"; "Saturday"}</pre>                                                  |
| ShortMonths         | <p>Twelve abbreviated names for the months of the year in the native language. These names can vary in length. There must be twelve values entered for this keyword.</p> <p>Example:</p> <pre>ShortMonths {"Jan";"Feb";"Mar";"Apr"; "May";"Jun";"Jul";"Aug";"Sep";"Oct";"Nov";"Dec"}</pre>                                                                                                               |
| LongMonths          | <p>Twelve full names for the months of the year in the native language. The order of the names must match the order of the names of the ShortMonths. These names can vary in length. There must be twelve values entered for this keyword.</p> <p>Example:</p> <pre>LongMonths {"January";"February";"March"; "April";"May"; "June";"July";"August"; "September";"October";"November"; "December"}</pre> |
| AMPM                | <p>The abbreviated names for AM (ante meridiem), and PM (post meridiem) in the native language. These names can vary in length. There must be two values entered for this keyword.</p> <p>Example:</p> <pre>AMPM {"AM";"PM"}</pre>                                                                                                                                                                       |
| RadixSeparator      | <p>The character that separates the integer and fractional parts of non-monetary strings. Separators cannot contain the following:</p> <ul style="list-style-type: none"> <li>• plus sign (+)</li> <li>• minus sign (-)</li> <li>• digits (0 through 9)</li> <li>• E</li> <li>• e</li> <li>• V</li> <li>• v</li> </ul>                                                                                   |

| SDF Element Keyword | Value Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                     | <p>The RadixSeparator value must be different from the GroupSeparator value. However, the RadixSeparator value can be the same as the CurrencyRadixSeparator value.</p> <p>Example:</p> <pre>RadixSeparator {"."}</pre>                                                                                                                                                                                                                                                                                                                                                                                                           |
| GroupSeparator      | <p>The character that separates groups of digits in the integer part of non-monetary strings.</p> <p>Separators cannot contain the following:</p> <ul style="list-style-type: none"> <li>• plus sign (+)</li> <li>• minus sign (-)</li> <li>• digits (0 through 9)</li> <li>• E</li> <li>• e</li> </ul> <p>The GroupSeparator value can be the same as the CurrencyGroupSeparator value.</p> <p>Example:</p> <pre>GroupSeparator {","}</pre>                                                                                                                                                                                      |
| GroupingRule        | <p>A group of numbers that defines the size of each group of digits in non-monetary strings.</p> <p>The size of each group can vary. Each number specifies how many digits are in each group with the initial integer defining the size of the group immediately preceding the radix separator, and the following integers defining groups preceding that.</p> <ul style="list-style-type: none"> <li>• If the last integer is -1, no further grouping shall be performed.</li> <li>• If the last integer is not -1, the size of the previous group (if any) shall be repeatedly used for the remainder of the digits.</li> </ul> |
|                     | <p>Examples:</p> <pre>GroupingRule {"3"}</pre> <p>For GroupingRule {"3"}, the numeric value 123456789.00 is formatted as 123,456,789.00.</p> <pre>GroupingRule {"3,2,-1"}</pre> <p>For GroupingRule {"3,2,-1"}, the numeric value 123456789.00 is formatted as 1234,56,789.00.</p>                                                                                                                                                                                                                                                                                                                                                |
| Currency            | <p>The string used to represent the local currency.</p> <p>Currencies cannot contain the following:</p> <ul style="list-style-type: none"> <li>• plus sign (+)</li> <li>• minus sign (-)</li> <li>• digits</li> <li>• CurrencyRadixSeparator</li> <li>• CurrencyGroupSeparator</li> <li>• E (by itself)</li> </ul>                                                                                                                                                                                                                                                                                                                |

| SDF Element Keyword | Value Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                     | <ul style="list-style-type: none"> <li>• e (by itself)</li> <li>• percent sign (%)</li> <li>• comma (,)</li> <li>• dot (.)</li> <li>• slash (/)</li> <li>• colon (:)</li> </ul> <p>Examples of dollar, yen, and euro currency strings:</p> <pre>Currency {"\$"} Currency {"\u00A5"} Currency {"\u20AC"}</pre>                                                                                                                                                                                                                                                                                                       |
| ISOCurrency         | <p>String that represents the local currency as an uppercase three-character code from ISO 4217. (You should verify that the string is specified in ISO 4217.)</p> <p>Currencies cannot contain the following:</p> <ul style="list-style-type: none"> <li>• plus sign (+)</li> <li>• minus sign (-)</li> <li>• digits</li> <li>• CurrencyRadixSeparator</li> <li>• CurrencyGroupSeparator</li> <li>• E (by itself)</li> <li>• e (by itself)</li> <li>• percent sign (%)</li> <li>• comma (,)</li> <li>• dot (.)</li> <li>• slash (/)</li> <li>• colon (:)</li> </ul> <p>Example:</p> <pre>ISOCurrency {"USD"}</pre> |
| CurrencyName        | <p>String that represents the local currency as a completely spelled out currency name.</p> <p>Currencies cannot contain the following:</p> <ul style="list-style-type: none"> <li>• plus sign (+)</li> <li>• minus sign (-)</li> <li>• digits</li> <li>• CurrencyRadixSeparator</li> <li>• CurrencyGroupSeparator</li> <li>• E (by itself)</li> <li>• e (by itself)</li> <li>• percent sign (%)</li> <li>• comma (,)</li> <li>• dot (.)</li> <li>• slash (/)</li> </ul>                                                                                                                                            |

| SDF Element Keyword    | Value Description                                                                                                                                                                                                                                                                                                                                                                                                                 |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                        | <ul style="list-style-type: none"> <li>colon (:)</li> </ul> <p>Example:</p> <pre>CurrencyName {"US Dollars"}</pre>                                                                                                                                                                                                                                                                                                                |
| CurrencyRadixSeparator | <p>The character that separates the integer and fractional part of monetary strings. Separators cannot contain the following:</p> <ul style="list-style-type: none"> <li>plus sign (+)</li> <li>minus sign (-)</li> <li>digits</li> <li>E</li> <li>e</li> <li>V</li> <li>v</li> </ul> <p>The value must be different that the value of CurrencyGroupSeparator.</p> <p>Example:</p> <pre>CurrencyRadixSeparator {"."}</pre>        |
| CurrencyGroupSeparator | <p>The character that separates groups of digits in the integer part of monetary strings. Separators cannot contain the following:</p> <ul style="list-style-type: none"> <li>plus sign (+)</li> <li>minus sign (-)</li> <li>digits</li> <li>E (by itself)</li> <li>e (by itself)</li> </ul> <p>The value must be different that the value of CurrencyRadixSeparator.</p> <p>Example:</p> <pre>CurrencyGroupSeparator {","}</pre> |
| CurrencyGroupingRule   | <p>A group of numbers that defines the size of each group of digits in monetary strings.</p> <p>The size of each group can vary. Each number specifies how many digits are in each group with the initial integer defining the size of the group immediately preceding the radix separator, and the following integers defining groups preceding that.</p> <p>Example:</p> <pre>CurrencyGroupingRule {"3"}</pre>                  |
| DualCurrency           | <p>String that represents the dual currency character.</p> <p>Currencies cannot contain the following:</p> <ul style="list-style-type: none"> <li>plus sign (+)</li> <li>minus sign (-)</li> </ul>                                                                                                                                                                                                                                |

| SDF Element Keyword | Value Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                     | <ul style="list-style-type: none"> <li>• digits</li> <li>• CurrencyRadixSeparator</li> <li>• CurrencyGroupSeparator</li> <li>• E (by itself)</li> <li>• e (by itself)</li> <li>• percent sign (%)</li> <li>• comma (,)</li> <li>• dot (.)</li> <li>• slash (/)</li> <li>• colon (:)</li> </ul> <p>Example:</p> <pre>DualCurrency {"\$"}</pre>                                                                                                                                                                                                 |
| DualISOCurrency     | <p>String that represents the dual currency as a three-character code from ISO 4217.</p> <p>Currencies cannot contain the following:</p> <ul style="list-style-type: none"> <li>• plus sign (+)</li> <li>• minus sign (-)</li> <li>• digits</li> <li>• CurrencyRadixSeparator</li> <li>• CurrencyGroupSeparator</li> <li>• E (by itself)</li> <li>• e (by itself)</li> <li>• percent sign (%)</li> <li>• comma (,)</li> <li>• dot (.)</li> <li>• slash (/)</li> <li>• colon (:)</li> </ul> <p>Example:</p> <pre>DualISOCurrency {"USD"}</pre> |
| DualCurrencyName    | <p>String that represents the dual currency as a completely spelled out currency name.</p> <p>Currencies cannot contain the following:</p> <ul style="list-style-type: none"> <li>• plus sign (+)</li> <li>• minus sign (-)</li> <li>• digits</li> <li>• CurrencyRadixSeparator</li> <li>• CurrencyGroupSeparator</li> <li>• E (by itself)</li> <li>• e (by itself)</li> <li>• percent sign (%)</li> </ul>                                                                                                                                    |

| SDF Element Keyword | Value Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                     | <ul style="list-style-type: none"> <li>• comma (,)</li> <li>• dot (.)</li> <li>• slash (/)</li> <li>• colon (:)</li> </ul> <p>Example:</p> <pre>DualCurrencyName {"US Dollars"}</pre>                                                                                                                                                                                                                                                                                                                                                                                                                            |
| BYTEINT             | <p>String that represents the default format applied to BYTEINT data types.</p> <p>Example:</p> <pre>BYTEINT {"-(3)9"}</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| INTEGER             | <p>String that represents the default format applied to INTEGER data types.</p> <p>Example:</p> <pre>INTEGER {"-(10)9"}</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| SMALLINT            | <p>String that represents the default format applied to SMALLINT data types.</p> <p>Example:</p> <pre>SMALLINT {"-(5)9"}</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| BIGINT              | <p>String that represents the default format applied to BIGINT data types.</p> <p>Example:</p> <pre>BIGINT {"-(19)9"}</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| NUMERIC             | <p>String that represents the default format applied to NUMERIC and DECIMAL data types.</p> <p>Example:</p> <pre>NUMERIC {"--(I).9(F)"}</pre> <ul style="list-style-type: none"> <li>• I represents the number of characters needed to display the integer portion of NUMERIC and INTEGER data. For example: <ul style="list-style-type: none"> <li>◦ With INTEGER type, I is 10.</li> <li>◦ With DECIMAL type (10,2), I is 8</li> </ul> </li> <li>• F represents the number of characters required to display the fractional portion of NUMERIC data. For example, with DECIMAL type (10,2), F is 2.</li> </ul> |
| REAL                | <p>String that represents the default format applied to REAL, DOUBLE PRECISION, and FLOAT data types.</p> <p>Example:</p> <pre>REAL {"-9.999999999999999E-999"}</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                            |

| SDF Element Keyword | Value Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DATE                | <p>String that represents the default format applied to DATE and PERIOD(DATE) data types.</p> <p>Example:</p> <pre>DATE {"YY/MM/DD"}</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| TIME                | <p>String that represents the default format applied to TIME, TIME WITH TIME ZONE, and PERIOD(TIME) data types.</p> <p>Example:</p> <pre>TIME {"HH:MI:SS.S(F)Z"}</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| TIMESTAMP           | <p>String that represents the default format applied to TIMESTAMP, TIMESTAMP WITH TIME ZONE, and PERIOD(TIMESTAMP) data types.</p> <p>Example:</p> <pre>TIMESTAMP {"YYYY-MMDBHH:MI:SS.S(F)Z"}</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| TimeZoneString      | <p>TimeZoneString names a time zone or GMT offset.</p> <p>The value is a sequence of strings that represents the time offset for the time zone.</p> <p>Example:</p> <pre>{"GMT-8"; "-8"; "0"}</pre> <p>For locations that observe Daylight Savings Time (DST), the string can define the set of rules for when in the calendar year the system time zone offset should change. The system will automatically adjust for DST according to this code. For example:</p> <p>Example:</p> <pre>{"America Pacific"; "-8"; "0"; "2"; "4"; "4"; "1"; "0"; "0"; "02:00:00"; "3"; "10"; "0"; "0"; "-1"; "02:00:00"; "1987"; "2006"; "-8"; "0"; "-7"; "0"; "4"; "3"; "8"; "0"; "0"; "02:00:00"; "4"; "11"; "1"; "0"; "0"; "02:00:00"; "2007"; "9999"; "-8"; "0"; "-7"; "0"}</pre> <p><b>Note:</b></p> <p>Rules in sample time zone strings cover years from 1987 to 9999. If you need the rules to cover years prior to 1987, you can modify the samples and add additional rules.</p> <p>For more information on time zone strings and time zone rules, see <i>Teradata Vantage™ - SQL Date and Time Functions and Expressions</i>, B035-1211.</p> <p>Also refer to the sample time zone strings and rules in the file <code>tdlocaledef_tzrules.txt</code>, which is in <code>/usr/tdbms/etc</code>.</p> |
| NUMBER              | <p>String the represents the default format applied to NUMBER data types.</p> <p>Example:</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |



| SDF Element Keyword | Value Description |
|---------------------|-------------------|
|                     | NUMBER {"FN9"}    |

## Example SDF Files

### Example: Vantage default locale format settings

```
// File name      : tdlocaledef.txt
// Purpose       : This file contains the specifications for data formatting(SDF)
//                defined by the user.
// Scope         : Project - subsystems
// Required files : None
// History        : 05-21-2001 - Created
//                10-27-2005 - Added BIGINT
//                05-24-2010 - Added TimeZoneString
// Description    : This data is compatible with pre-V2R5 releases.
// DBS System Formatting Data
// Day and month names
    ShortDays {
        "Sun";
        "Mon";
        "Tue";
        "Wed";
        "Thu";
        "Fri";
        "Sat"
    }
    LongDays {
        "Sunday";
        "Monday";
        "Tuesday";
        "Wednesday";
        "Thursday";
        "Friday";
        "Saturday"
    }
    ShortMonths {
        "Jan";
        "Feb";
        "Mar";
        "Apr";
```

```

    "May";
    "Jun";
    "Jul";
    "Aug";
    "Sep";
    "Oct";
    "Nov";
    "Dec"
}
LongMonths {
    "January";
    "February";
    "March";
    "April";
    "May";
    "June";
    "July";
    "August";
    "September";
    "October";
    "November";
    "December"
}
AMPM {
    "AM";
    "PM"
}
// Parsing Elements
RadixSeparator {"."}
GroupSeparator {","}
GroupingRule   {"3"}
Currency       {"$"}
ISOCurrency    {"USD"}
CurrencyName   {"US Dollars"}
CurrencyRadixSeparator {"."}
CurrencyGroupSeparator {","}
CurrencyGroupingRule {"3"}
DualCurrency   {"$"}
DualISOCurrency {"USD"}
DualCurrencyName {"US Dollars"}
// Data type default formats
BYTEINT        {"-(3)9"}
INTEGER        {"-(10)9"}
SMALLINT       {"-(5)9"}

```

```

BIGINT      {"-(19)9"}
NUMERIC     {"--(I).9(F)"}
REAL        {"-9.999999999999999E-999"}
DATE        {"YY/MM/DD"}
TIME        {"HH:MI:SS.S(F)Z"}
TIMESTAMP   {"YYYY-MM-DDBHH:MI:SS.S(F)Z"}
NUMBER      {"FN9"}
// System Time Zone string
TimeZoneString {""}

```

### Example: Customized SDF for Japanese Data Formatting

This example shows how an SDF file might be customized for a Japanese location. The file contains Japanese date and time default formatting and the Unicode hex notation for Kanji characters. Also, Unicode characters in the default format strings are specified with “\u.”

```

// DBS System Formatting Data
// Day and month names
ShortDays {
    "\u65E5";
    "\u6708";
    "\u706B";
    "\u6C34";
    "\u6728";
    "\u91D1";
    "\u571F"
}
LongDays {
    "\u65E5\u66DC\u65E5";
    "\u6708\u66DC\u65E5";
    "\u706B\u66DC\u65E5";
    "\u6C34\u66DC\u65E5";
    "\u6728\u66DC\u65E5";
    "\u91D1\u66DC\u65E5";
    "\u571F\u66DC\u65E5"
}
ShortMonths {
    "1\u6708";
    "2\u6708";
    "3\u6708";
    "4\u6708";
    "5\u6708";
    "6\u6708";
    "7\u6708";

```

```

        "8\u6708";
        "9\u6708";
        "10\u6708";
        "11\u6708";
        "12\u6708"
    }
    LongMonths {
        "1\u6708";
        "2\u6708";
        "3\u6708";
        "4\u6708";
        "5\u6708";
        "6\u6708";
        "7\u6708";
        "8\u6708";
        "9\u6708";
        "10\u6708";
        "11\u6708";
        "12\u6708"
    }
    AMPM {
        "\u5348\u524D";
        "\u5348\u5F8C"
    }
    // Parsing Elements
    RadixSeparator      {"."}
    GroupSeparator      {","}
    GroupingRule        {"3"}
    Currency             {"\u00A5"}
    ISOCurrency         {"JPY"}
    CurrencyName        {"Yen"}
    CurrencyRadixSeparator {"."}
    CurrencyGroupSeparator {","}
    CurrencyGroupingRule {"3"}
    DualCurrency         {"\u00A5"}
    DualISOCurrency     {"JPY"}
    DualCurrencyName    {"Yen"}
    // Data type default formats
    BYTEINT             {"-(3)9"}
    INTEGER              {"G-(I)9"}
    SMALLINT            {"G-(I)9"}
    BIGINT              {"G-(I)9"}
    NUMERIC              {"G--(I)D9(F)"}
    REAL                 {"G-9D9999999999999E-999"}

```

```
DATE          {"YYYY\u5E74MM\u6708DD\u65E5"}
TIME          {"HH\u6642MI\u5206SSDS(F)\u79D2Z"}
TIMESTAMP     {"YYYY\u5E74MM\u6708DD\u65EBHH\u6642MI\u5206SSDS(F)
\u79D2Z"}      NUMBER          {"FN9"}
// System Time Zone string
TimeZoneString {""}
```

## TPA Reset (tpareset)

---

### Note:

The utility described is not available for as-a-service customers.

---

The TPA Reset utility, `tpareset`, resets Advanced SQL Engine. All PDE tasks except for some kernel daemons are killed. All database processes, including those for AMPs and PEs, are stopped and restarted. TPA Reset options control whether dump information is collected, whether the node on which `tpareset` is run is rebooted, and whether the system remains down or restarts.

TPA Reset can be used:

- after Advanced SQL Engine has been reconfigured
- to activate new versions of Advanced SQL Engine software
- to recover from a database hang
- in other situations that warrant a full database shutdown and restart

Use TPA Reset only when absolutely necessary. TPA Reset has a system-wide effect. Shutting down the system and restarting not only causes system down time, but can also have a performance impact when the system starts up and runs standard recovery and reconcile operations.

## Runs From

- Database Window or comparable interface to the Teradata console subsystem, such as `cnstern`
- Linux command line

For general information on starting the utilities from different interfaces, see [Starting the Utilities](#).

## Syntax

```
tpareset {
  [ { -xit | -x | -exit | -e } |
    { -stop | -s } |
    { -force | -f }
  ]
  [ { -panic | -p } [-nodump] | { -dump | -d } ]
  [ -yes | -y ] reason |

  { -help | -h }
}
```

**Note:**

Some strictly internal and rarely-needed options have been omitted from this discussion. Those options are documented in the tpareset command-line help.

**Syntax Elements****-xit****-x****-exit****-e**

Shuts down Vantage on all nodes of the system. The system must be explicitly restarted after the -x option is used.

**-stop****-s**

Brings PDE down on the node from which the tpareset was initiated. The other nodes go through a reset cycle, but the initiating node stays down. On an SMP system, this is equivalent to -exit.

Sets the state of the initiating node to DOWN/TDMAINT. Nodes in this state stay down until tpareset -f or a system wide shutdown and restart brings them back into the Teradata system.

**-force****-f**

Forces a full restart on all nodes of the system. This option has the following effects:

- Brings any nodes that are running but in DOWN/TDMAINT state back into the Teradata system.
- Clears the system-wide crash count

If the crash count was preventing the database from being started, this option forces the database to be started on all nodes. -force should not be used until the condition that caused the crash loop has been corrected.

- Forces reconcile to run

Reconcile normally runs when any node that was down is brought back up, or when a database version switch has been requested, necessitating a system reset. Reconcile does the following:

- validates that each node is running the same version of the various software components that comprise Vantage
- ensures that these versions are compatible with each other
- switches versions of specific components if such a version switch has been requested. (Such version switches are specified from the ctl utility programs.)

The `-f` option forces reconcile to run during a reset, regardless of status of system nodes and regardless of whether a version switch has been requested.

- Forces the database cache to be discarded

The database cache may also be discarded in other reset situations when the `-f` option has not been specified.

Reinitializing the cache can add several minutes to restart time.

### **-panic**

#### **-p**

Initiates a reboot of the node where `tpareset` was executed. A database dump is captured from all nodes, and an operating system dump is captured on the rebooted node. The remaining nodes undergo a normal reset and restart cycle without being rebooted.

When the database restarts, the node from which `tpareset` was run is excluded from the parallel database system. When the problem that required the panic reset is resolved, the node can be returned to service by issuing another `tpareset`.

---

#### **Note:**

This option should be used only if the Teradata Support Center determines that a dump is necessary for problem isolation.

---

### **-nodump**

Prevents the `-panic` option from capturing a database-level dump. However, an operating system dump will still be captured on the initiating node, initiating a node reboot.

### **-dump**

#### **-d**

Causes a database dump to be captured before the system is reset and restarted.

---

#### **Note:**

This option should be used only if the Teradata Support Center determines that a dump is necessary for problem isolation.

---

### **-yes**

#### **-y**

Suppresses the normal `tpareset` confirmation prompt.



**reason**

The tpareset command must end with a string that describes the reason for the reset. The reason need not be delimited by double quotation marks or apostrophes. The string cannot start with a dash (-).

**-help****-h**

Displays tpareset command syntax information.

## Usage Notes

The RESTART TPA command, available from Database Window (DBW), is similar to TPA Reset, in that it resets and restarts Vantage. However, RESTART TPA lacks several of the options of TPA Reset. For example, RESTART TPA cannot be used to shut down the PDE component of Vantage. For more information about DBW, see [Database Window \(xdbw\)](#).

TPA Reset can only be run on nodes where PDE is running.

## Example: Resetting (restarting) the Database

```
> tpareset -d Recover from DBS hang.
```

```
You are about to restart the database
      on the system
      'Test1'
Do you wish to continue (default: n) [y,n]
```

## Update DBC (updatedbc)

**Note:**

The utility described is not available for as-a-service customers.

The Update DBC utility, updatedbc, performs the following:

| In table ...  | Update DBC recalculates ...                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DBASE         | <p>PermSpace, SpoolSpace, and TempSpace values for user DBC.</p> <ul style="list-style-type: none"> <li>• The PermSpace value in DBASE for user DBC is the total available storage space minus the PermSpace for all other databases.</li> <li>• The SpoolSpace and TempSpace values in DBASE for user DBC are the total available storage space.</li> </ul> <p>For databases other than DBC, the PermSpace, SpoolSpace, and TempSpace values in the DBASE table are the maximums declared when the database is defined.</p> <p>The DBASE table includes the following columns:</p> <ul style="list-style-type: none"> <li>• PermSpace</li> <li>• SpoolSpace</li> <li>• TempSpace</li> </ul> |
| DATABASESPACE | <p>MaxPermSpace, MaxSpoolSpace, and MaxTempSpace values for each database in the system based on the PermSpace, SpoolSpace, and TempSpace values in the DBASE table for that database.</p> <p>The DATABASESPACE table includes the following columns:</p> <ul style="list-style-type: none"> <li>• CurrentPermSpace</li> <li>• CurrentSpoolSpace</li> <li>• CurrentTempSpace</li> <li>• MaxPermSpace</li> <li>• MaxSpoolSpace</li> <li>• MaxTempSpace</li> </ul>                                                                                                                                                                                                                             |

Values in the DBASE table are the global values. Values in the DATABASESPACE table are local AMP values. The calculation is the global value divided by the number of AMPs in the system.

The following table lists the difference between the Update DBC and Update Space utilities.

| The utility ... | Recalculates the ...                                              |
|-----------------|-------------------------------------------------------------------|
| Update DBC      | maximum allowed values for permanent, temporary, and spool space. |
| Update Space    | current usage for permanent, temporary, and spool space.          |

Use Update DBC only to correct inconsistency in the DBASE or DATABASESPACE tables, which might occur as the result of rare types of system failures.

For descriptions of the DBASE and DATABASESPACE tables and columns, see *Teradata Vantage™ - Data Dictionary*, B035-1092.

For more information on the Update Space utility, see [Update Space \(updatespace\)](#).

## Runs From

- From Database Window or comparable interface to the Teradata console subsystem, such as cnstern.
- Linux command line

For general information on starting the utilities from different interfaces, see [Starting the Utilities](#).

## Usage Notes

In order to run Update DBC, the system should be quiescent (no users should be logged on). To see the system state, type `pdestate -a` at the Linux command line.

Do not use the CNS/Database Window STOP command to abort Update DBC when it is running.

## Update Space (updatespace)

---

### Note:

The utility described is not available for as-a-service customers.

---

The Update Space utility, `updatespace`, recalculates the permanent, temporary, or spool space used by either of the following:

- A single database and its individual tables
- All databases in a system and their individual tables

Update Space accomplishes this by performing the following:

- Examining storage descriptors and adding up space for each table.
- Setting values in `CurrentPermSpace`, `CurrentTempSpace`, or `CurrentSpoolSpace` in the `DATABASESPACE` table for each table and for the containing database as a whole.

The following table lists the difference between the Update DBC and Update Space utilities.

| The utility ... | Recalculates the ...                                              |
|-----------------|-------------------------------------------------------------------|
| Update DBC      | maximum allowed values for permanent, temporary, and spool space. |
| Update Space    | current usage for permanent, temporary, and spool space.          |

Use Update Space only to correct inconsistency in the `DATABASESPACE` table, which might occur as the result of rare types of system failures.

For descriptions of the `DBASE` and `DATABASESPACE` tables and columns, see *Teradata Vantage™ - Data Dictionary*, B035-1092.

For more information on the Update DBC utility, see [Update DBC \(updatedbc\)](#).

## Runs From

- From Database Window or comparable interface to the Teradata console subsystem, such as `cnstern`.
- Linux command line

For general information on starting the utilities from different interfaces, see [Starting the Utilities](#).

## Syntax

Update Space presents a command-line environment that allows the entry of the `UPDATE` command, which has the following syntax.

```
UPDATE {
  { TEMPORARY | ALL } SPACE FOR { ALL DATABASES | dbname | ALL PROXYUSERS } |

  { SPOOL | PSPOOL } SPACE FOR { dbname | ALL PROXYUSERS }
} ;
```

## Syntax Elements

### TEMPORARY

Specifies to update only the temporary space.

### ALL

Specifies to update all the space.

### SPACE FOR

Specifies the name of the database(s) for which space is to be recalculated.

### ALL DATABASES

Specifies all the databases in a system.

### *dbname*

Specifies the name of a single database.

### ALL PROXYUSERS

Specifies to update the specified space for all Trusted Sessions proxy users.

### SPOOL

Specifies to update only the spool space.

---

#### Note:

Spool space can be calculated for a single database only.

---

### PSPOOL

Specifies to update only the spool space that persists across restarts.

## Usage Notes

To exit the Update Space utility, type q or quit with no semicolon terminator, on the command line.



```
Updating space for ALL  
Space updated for ALL
```

## Example: Updating temporary space for PUBLIC database

```
Enter Command  
  
update temporary space for public;  
  
Updating space for PUBLIC  
Space updated for PUBLIC
```

## Vproc Manager (vprocmanager)

---

**Note:**

The utility described is not available for as-a-service customers.

---

The Vproc Manager utility, `vprocmanager`, allows you to manage the virtual processors (vprocs) in an Advanced SQL Engine system. Vprocs are a set of software processes that run under Teradata Parallel Database Extensions (PDE) within the multitasking environment of the operating system. Vprocs are the heart of Advanced SQL Engine, managing the basic functions of the system.

Advanced SQL Engine systems can include the following vproc types.

| Vproc Type | Description                                                                                                                                                                                                    |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AMP        | Access module processors perform database functions, such as executing database queries. Each AMP owns a portion of the overall database storage.                                                              |
| GTW        | Gateway vprocs provide a socket interface to Vantage.                                                                                                                                                          |
| Node       | The node vproc handles PDE and operating system functions not directly related to AMP and PE work. Node vprocs cannot be externally manipulated, and do not appear in the output of the Vproc Manager utility. |
| PE         | Parsing engines perform session control, query parsing, security validation, query optimization, and query dispatch.                                                                                           |
| RSG        | Relay Services Gateway provides a socket interface for communication with Teradata Data Stream Architecture (DSA).                                                                                             |
| TVS        | Manages Vantage storage. AMPs acquire their portions of database storage through the TVS vproc.                                                                                                                |

Vproc Manager allows you to perform the following:

- Obtain the status of some or all of the vprocs
- Change vproc states
- Initialize and boot one or more specified AMP vprocs
- Initialize the storage associated with one or more specified AMP vprocs
- Force a database restart

Vproc Manager can be used with the Table Rebuild utility to initialize and boot a specific AMP vproc in order to rebuild the portion of all tables associated with the vproc.



## Runs From

- Database Window or comparable interface to the Teradata console subsystem, such as cnstern
- Linux command line
- Teradata Viewpoint Remote Console portlet

For general information on starting the utilities from different interfaces, see [Starting the Utilities](#). For information on Viewpoint, see *Teradata® Viewpoint User Guide*, B035-2206.

## Definitions of Terms Used in Vproc Manager

The following terms apply to Vproc Manager:

- VprocID

A VprocID is a number that identifies the vproc to Vantage. VprocIDs are in the range of 0 through 16383, or 0 through 30719, depending on the system. When specifying VprocIDs, decimal or hexadecimal numbers can be used. Hexadecimal numbers must include a trailing X or x, for example, 3FFx and 77FFx.

---

### Note:

Vproc Number and Rel. Vproc # are two columns in the output of Vproc Manager STATUS command output displays. Where commands allow vproc numbers to be specified, use the absolute Vproc Number rather than the Rel. Vproc # number to identify a vproc.

---

- VprocList

A VprocList is defined as a list of one or more vproc identifiers or a range of vproc identifiers in the following format:

```
{ vprocid | vprocid TO vprocid } [[,]...]
```

Examples:

```
0 1 2 3
0, 1, 16382 to 16383
0 to 10, 16381, 16382
0 to 1, 27FFx
```

- VprocState

VprocState defines the PDE system state of a vproc. VprocStates include the following:

- **FATAL**

This VprocState indicates a serious problem with a vproc or its associated storage. For example, if a vproc crashes repeatedly, or if there are corrupt tables that require a Table Rebuild, a vproc will be set to the FATAL state.

When a TVS vproc is in FATAL state, all AMPs associated with it will be put into FATAL state at the next database restart.

---

**Note:**

The AMP or PE partitions are not started for a vproc in this state.

---

- **FATAL\***

This VprocState indicates a lost I/O write.

- **FATAL\*\***

This VprocState indicates an I/O read issue.

- **NEWPROC**

This VprocState applies only to AMP and PE vprocs.

This VprocState indicates that either a new vproc is being added to the database configuration or an existing vproc is being deleted.

A vproc with status NEWPROC is a member of the PDE message group but is not a member of the operational database message group. For more information, contact the Teradata Support Center.

- **NONODE**

This VprocState indicates that the physical hardware required to run this vproc is not available. This state is not accepted as an argument to the SET command, although this state might appear in the Vproc Status Table produced by the STATUS command.

---

**Note:**

The AMP or PE partitions are not started for a vproc in this state.

---

- **NULL**

This VprocState is undefined. It is not accepted as an argument to the SET command, although this state might appear in the Vproc Status Table produced by the STATUS command.

- **ONLINE**

This VprocState indicates that the vproc is fully operational and actively participating with the database.

A vproc with status ONLINE is a member of both the PDE and database message group. For more information, contact the Teradata Support Center.

- **OFFLINE**

Generally, this VprocState indicates vprocs that are not fully operational and have been forced down by the DBA or system administrator.

For example, if a node fails during system startup, its associated storage clique may be prevented from starting, and the AMPs that use that storage will be set to OFFLINE. This behavior is governed by settings in the Control GDO Editor (ctl) utility.

A vproc with status OFFLINE is a member of the PDE message group but is not a member of the operational database message group. For more information, contact the Teradata Support Center.

When a TVS vproc is in OFFLINE state, all AMPs associated with it will be put into FATAL state at the next database restart.

- **UTILITY**

This vproc state applies only to AMP vprocs.

This VprocState is transitional and is used by database recovery, reconfiguration, and the Table Rebuild utilities to indicate that a previously OFFLINE/FATAL/NEWPROC is interacting with the online database.

This vproc is a member of the PDE message groups but not a member of the operational database message groups.

- **ConfigStatus**

The database component of Vantage maintains its own internal version of the status of AMP and PE vprocs. That status is called the Teradata Logical Configuration Status, and represented in the output of the Vproc Manager STATUS command as a column named ConfigStatus. Vproc Manager cannot modify the ConfigStatus of a vproc.

ConfigStatus can be any of the following:

---

**Note:**

To clearly differentiate between VprocStates and ConfigStatus, VprocStates here are written in all UPPERCASE letters, and ConfigStatus is written in mixed-case letters.

---

- **Online**

The vproc is fully operational.

This usually coincides with the ONLINE VprocState.

- **Down**

The vproc has been forced down.

This ConfigStatus usually coincides the OFFLINE, UTILITY, FATAL, and NONODE VprocStates.

- **Catchup**

This ConfigStatus for this vproc was previously Down, and it is being recovered in the background. If System RestartKind is COLDWAIT, the ConfigStatus of this vproc will become Online after recovery is complete.

This ConfigStatus usually coincides with the UTILITY VprocState.

- **Hold**

The ConfigStatus for this vproc was previously Catchup or Down, and its data is in the process of being recovered. The ConfigStatus of this vproc will become Online after recovery is complete.

This ConfigStatus usually coincides with the ONLINE VprocState.

- **NewReady**

This is either a newly added vproc or one that has been removed from the database logical configuration.

This ConfigStatus usually coincides with the UTILITY or NEWPROC VprocState.

- **NewDown**

This is a newly added vproc and has been forced down.

This ConfigStatus usually coincides with the NEWPROC VprocState.

- **Null**

This vproc is not yet in the Teradata Logical Configuration. Recently, the vproc might have been added to the PDE Physical Configuration or deleted from the Teradata Logical Configuration, and database startup has not run yet. Startup will note that this vproc does not appear in its configuration map and will change its ConfigStatus to NewReady. Null usually coincides with the NEWPROC VprocState.

The following table illustrates each ConfigStatus and the VprocState that usually coincides with each ConfigStatus.

| VprocState                 | ConfigStatus            |
|----------------------------|-------------------------|
| ONLINE                     | Online, Hold            |
| OFFLINE<br>FATAL<br>NONODE | Down                    |
| UTILITY                    | Catchup, Down, NewReady |
| NEWPROC                    | NewReady, NewDown, Null |

For an understanding of how vprocs behave during database startup, see [VprocState/ConfigStatus Transitions During System Startup](#).

- **RestartKind**

RestartKind specifies the type of system restart to perform during the next database restart. Types of valid restarts include:

- **COLD**

A full restart, but transaction recovery will be deferred.

- **COLDWAIT**

A full restart, but database startup will be held up until transaction recovery is complete.

## Error VprocStates

Vproc Manager displays specific error states that may indicate an unrecognized vproc or a conflict between the DBS Logical Configuration Status of a vproc and the PDE system state of the same vproc. Depending on subsequent activities targeted to this vproc, such as attempts to change its state or to move it into or out of the configuration, data corruption or other unintended consequences can occur. When Vproc Manager displays one of the following error states in any column of the output of the Vproc Manager STATUS command, contact the Teradata Support Center for information about returning the vproc to a valid VprocState.

- MOCK
- InID
- NoAP
- NoGT
- INA
- INAP
- ISt
- NV2P
- InV

## VprocState/ConfigStatus Transitions During System Startup

The following is the Key for the table that follows:

- “=” is read as “remains.”
- “->” is read as “becomes.”

For a definition of RcvJrnl, see [STATUS](#).

| CONFIG-STATUS<br>(prior to System startup) | VprocState (prior to System startup)          |                                                                                               |                                                                |                                                                                                                        |
|--------------------------------------------|-----------------------------------------------|-----------------------------------------------------------------------------------------------|----------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|
|                                            | ONLINE                                        | UTILITY (AMP vprocs only)                                                                     | OFFLINE/FATAL / NONODE                                         | NEWPROC                                                                                                                |
| Online                                     | VprocState = ONLINE<br>ConfigStatus = Online  | VprocState -> ONLINE<br>ConfigStatus = Online                                                 | VprocState = OFFLINE/FATAL/<br>NONODE\<br>ConfigStatus -> Down | This combination is inconsistent and will be transitioned as follows:<br>VprocState -> ONLINE<br>ConfigStatus = Online |
| Hold (AMP vprocs only)                     | VprocState = ONLINE<br>ConfigStatus -> Online | This combination is inconsistent and will be transitioned as follows:<br>VprocState -> ONLINE | VprocState = OFFLINE/FATAL/<br>NONODE                          | This combination is inconsistent and will be transitioned as follows:                                                  |

| CONFIG-STATUS<br>(prior to System startup) | VprocState (prior to System startup)                                                                                                                                                                                                                                                                 |                                                                                                                                                                                                                                |                                                           |                                                                                                                          |
|--------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
|                                            | ONLINE                                                                                                                                                                                                                                                                                               | UTILITY (AMP vprocs only)                                                                                                                                                                                                      | OFFLINE/FATAL / NONODE                                    | NEWPROC                                                                                                                  |
|                                            |                                                                                                                                                                                                                                                                                                      | ConfigStatus -> Online                                                                                                                                                                                                         | ConfigStatus -> Down                                      | VProcState -> ONLINE<br>ConfigStatus -> Online                                                                           |
| Catchup (AMP vprocs only)                  | This combination is inconsistent and will be transitioned as follows:<br>IF (NOT RcvJrnl) THEN<br>VprocState -> OFFLINE<br>ConfigStatus -> Down<br>ELSE IF (RestartKind = COLDWAIT) THEN<br>VprocState = ONLINE<br>ConfigStatus -> Online<br>ELSE<br>VprocState -> UTILITY<br>ConfigStatus = Catchup | IF (NOT RcvJrnl) THEN<br>VprocState -> OFFLINE<br>ConfigStatus -> Down<br>ELSE<br>IF (RestartKind = COLDWAIT) THEN<br>VprocState -> ONLINE<br>ConfigStatus -> Online<br>ELSE<br>VprocState = UTILITY<br>ConfigStatus = Catchup | VprocState = OFFLINE/FATAL/NONODE<br>ConfigStatus -> Down | This combination is inconsistent and will be transitioned as follows:<br>VprocState -> UTILITY<br>ConfigStatus = Catchup |
| Down                                       | IF (VprocType = PE) THEN<br>VprocState = ONLINE<br>ConfigStatus -> Online<br>ELSE IF (VprocType = AMP) THEN IF (NOT RECONFIG) AND RcvJrnl THEN<br>VprocState -> UTILITY<br>ConfigStatus -> Catchup<br>ELSE IF (NOT RcvJrnl) THEN                                                                     | This vproc is currently undergoing an ALL TABLES REBUILD using the Table Rebuild utility program, therefore:<br>VprocState = UTILITY<br>ConfigStatus = Down<br>Table Rebuild will set to it ONLINE/Online when it is complete. | VprocState = OFFLINE/FATAL/NONODE<br>ConfigStatus = Down  | This combination is inconsistent and will be transitioned as follows<br>VprocState -> OFFLINE<br>ConfigStatus = Down     |

| CONFIG-STATUS<br>(prior to System startup) | VprocState (prior to System startup)                                                                                         |                                                                                                                                                                                                             |                                                               |                                                  |
|--------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------|--------------------------------------------------|
|                                            | ONLINE                                                                                                                       | UTILITY (AMP vprocs only)                                                                                                                                                                                   | OFFLINE/FATAL / NONODE                                        | NEWPROC                                          |
|                                            | VprocState<br>-> OFFLINE<br>ConfigStatus = Down                                                                              |                                                                                                                                                                                                             |                                                               |                                                  |
| NewReady                                   | This combination is inconsistent and will be transitioned as follows:<br>VProcState<br>-> NEWPROC<br>ConfigStatus = NewReady | IF RECONFIG THEN<br>VprocState = UTILITY<br>ConfigStatus = NewReady<br>ELSE<br>This combination is inconsistent and will be transitioned as follows:<br>VprocState<br>-> NEWPROC<br>ConfigStatus = NewReady | VprocState = OFFLINE/FATAL/ NONODE<br>ConfigStatus -> NewDown | VprocState = NEWPROC<br>ConfigStatus = NewReady  |
| NewDown                                    | This combination is inconsistent and will be transitioned as follows:<br>VprocState<br>-> OFFLINE<br>ConfigStatus = NewDown  | This combination is inconsistent and will be transitioned as follows:<br>VprocState --> OFFLINE<br>ConfigStatus = NewDown                                                                                   | VprocState = OFFLINE/FATAL/ NONODE<br>ConfigStatus = NewDown  | VprocState = NEWPROC<br>ConfigStatus -> NewReady |

## Vproc Manager Command Syntax

Vproc Manager presents a command-line environment that allows the entry of Vproc Manager commands, which have the following general syntax:

```
command options arguments [ ; ]
```

### Syntax Elements

#### **command**

A command in [Vproc Manager Commands](#).

#### **options arguments**

VprocID, VprocState, or RestartKind.

---

**Note:**

All arguments can be abbreviated to their shortest unique string.

---

---

**Note:**

The Parallel Database Extensions (PDE) must be running in order for Vproc Manager to run. Vantage does not need to be running.

---

## Vproc Manager Commands

The commands are described in more detail in the sections that follow.



## BOOT

The BOOT command initialize and start the database partitions of an AMP. This command applies only to AMPs with VprocState of FATAL and ConfigStatus of Down.

### Syntax

```
{ BOOT | B } vprocid
```

### Syntax Elements

#### *vprocid*

The number that identifies the vproc to be rebooted. For a description of vprocIDs, see [Definitions of Terms Used in Vproc Manager](#).

### Usage Notes

The boot process asynchronously invokes the AMP startup task on the specified AMP. Vproc Manager considers the boot complete after it has invoked the AMP startup task successfully in the startup partition on the specified AMP. The AMP startup task itself determines whether the system and the AMP are in the appropriate state for a boot and logs messages to the system console.

Use this command in conjunction with Table Rebuild when you need to rebuild all tables on an AMP. For more information, see [Table Rebuild \(rebuild\)](#).

### Example: Booting a specified AMP

The following example shows how to boot up AMP vproc #63.

```
Enter a command, Help, or Quit:
```

```
BOOT 63
```

```
WARNING: This command will destroy all user and dictionary data of vproc 63's
vdisk. To recover from this may require an ALL TABLES REBUILD!
Are you sure you want to do this (Y/N)?
```

```
Y
```

```
The AMP startup task has been successfully invoked of vproc 63.
Please refer to the system console for additional information.
```

## CLEARMVAMP

Clears the status of AMPs that were previously configured as target AMPs in move AMP reconfig operation. Do not use this command during move or add AMP reconfig operations. Running this command will cause the File System to be reinitialized on the target AMPs of the last reconfiguration MOVE AMP operation.

### Syntax

```
{ CLEARMVAMP | CL }
```

## HELP

The HELP command provides general help for Vproc Manager or detailed help if you specify an option.

If no command option is specified, a brief introduction to Vproc Manager is displayed, followed by instructions on how to receive additional help.

### Syntax

```
{ HELP | H } [ ALL | keyword ]
```

### Syntax Elements

#### ALL

Used to display the help text of the Vproc Manager in its entirety.

#### *keyword*

Either a command name, a VprocState, or a RestartKind.

### Example: Vproc Manager online help

Enter a command, Help, or Quit:

```
help
```

The Vproc Manager utility program provides a means to manage/manipulate various vproc attributes. The general command syntax is:

```
<COMMAND> <Options> <Arguments> [;]
```

That is, a command followed by its specific options and/or arguments and terminated with an optional semi-colon. All commands, options, and arguments may be abbreviated to the shortest unique string.

Valid commands are:

```
BOOT, HELP, INITVDISK, QUIT, RESTART, SET, and STATUS.
```

Enter "HELP <CommandName>" for detailed information on each command or type "HELP ALL" for the help text in its entirety.

### Example: Vproc Manager help for the QUIT command

Enter a command, Help, or Quit:

```
help quit
```

QUIT

- o This command causes the VprocManager utility program to exit.

**Example: Vproc Manager help for COLD restart**

Enter a command, Help, or Quit:

help cold

COLD and COLDWAIT are used to specify the types of restart to perform, either as an option in the RESTART command or as a value supplied in the SET RESTART command. They are defined as follows:

- o COLD - A full restart, but transaction recovery will be deferred.
- o COLDWAIT - A full restart, but DBS startup will be held up until transaction recovery is complete.

## INITVDISK

The INITVDISK command initializes the Teradata File System on the storage associated with a specific AMP, also known as the AMP's vdisk. This applies only to AMPs with a VprocState of FATAL or NEWPROC. The state of the specified AMP remains unchanged.

### Syntax

```
{ INITVDISK | I } vprocid
```

### Syntax Elements

#### *vprocid*

The number that identifies the vproc to be initialized. For a description of vprocIDs, see [Definitions of Terms Used in Vproc Manager](#).

### Usage Notes

The initialization is accomplished by invoking File System Initialization Task in the Application partition on the specified AMP. InitVdisk will abort if the Application partition is in use already.

The initialization of the storage associated with a NEWPROC AMP is only relevant prior to database startup.

System startup implicitly initializes the storage for NEWPROC AMPs (if necessary).

### Example: Initializing the storage associated with a specified AMP

```
Enter a command, Help, or Quit:
```

```
INITVDISK 3
```

```
WARNING: This command will destroy all user and dictionary data of vproc 3's
vdisk. To recover from this may require an ALL TABLES REBUILD!
```

```
Are you sure you want to do this (Y/N)?
```

```
Y
```

```
Starting the file system initialization task on vproc 3.
The Vdisk associated with vproc 3 has been initialized.
```

## QUIT

The QUIT command causes Vproc Manager to exit.

### Syntax

```
{ QUIT | Q }
```

## RESTART

The RESTART command forces a database restart.

The RESTART command implicitly causes Vproc Manager to exit.

### Syntax

```
{ RESTART | R }
  [ TPA ]
  [ NODUMP | DUMP { YES | NO } ]
  [ restartkind ]
  [ restart_comment ]
```

### Syntax Elements

#### TPA

This option has no effect on the RESTART command. It is included for compatibility.

#### NODUMP

A system dump will not occur. This is the default.

#### DUMP

Specifies whether a dump will occur.

#### *restartkind*

Specifies the type of restart to perform:

- COLD indicates a full restart, including the database (DBS) component, however, transaction recovery will be deferred.
- COLDWAIT indicates a full restart, however, DBS startup will be deferred until transaction recovery is complete.

If you do not specify a RestartKind, the current system setting is used.

#### *restart\_comment*

States the reason for the restart.

### Usage Notes

To set the desired RestartKind or Restart Type, see or [SET RESTART](#) or the RESTART TPA command in [Database Window \(xdbw\)](#).

To view the current system setting for RestartKind or Restart Type, see [STATUS DBS](#).

For additional information on stopping and restarting the system, see *Teradata Vantage™ - Database Administration*, B035-1093.

### Example: Performing a COLD restart

The following example shows how to perform a COLD restart and specifies a reason why.

```
Enter a command, Help, or Quit:

RESTART COLD This is a test.

The system will be restarted:
  Dump          : NO
  RestartKind    : COLD
  Reason         : This is a test.
```

### Example: Restarting with a dump

The following example shows how to restart using the current system setting for RestartKind and specifies a reason why.

```
Enter a command, Help, or Quit:

RESTART DUMP = YES This is a test.

The system will be restarted:
  Dump          : YES
  RestartKind    : COLD
  Reason         : This is yet another test.
```

### Example: Restarting using current restart settings

The following example shows how to restart using the current system setting for RestartKind and the default restart comment.

```
Enter a command, Help, or Quit:

RESTART

The system will be restarted:
  DUMP          : NO
  RestartKind    : COLD
  Reason         : System restarted by VprocManager.
```



## SET RESTART

The SET RESTART command sets the restart type to use during the next restart of the system.

This function does not force an immediate database restart.

### Syntax

```
{ SET | S } { RESTART | R } [=] restartkind
```

### Syntax Elements

#### *restartkind*

Specifies the type of restart to perform:

- COLD indicates a full restart, including the database (DBS) component, however, transaction recovery will be deferred.
- COLDWAIT indicates a full restart, however, DBS startup will be deferred until transaction recovery is complete.

If you do not specify a RestartKind, the current system setting is used.

For additional information, see [RESTART](#).

### Usage Notes

This function does not force an immediate database restart.

### Example: Setting the restart type

Enter a command, HELP or QUIT:

```
SET RESTART = COLDWAIT
```

The System RestartKind has been set to COLDWAIT.

# SET *vproclist*

The SET *vproclist* command sets the new state of a vproc or the new states for a list of vprocs.

## Syntax

```
{ SET | S } [ /V ] vproclist_spec [[,]...]
```

## Syntax Elements

*vproclist\_spec*

```
vproclist [=] vprocstate
```

*/V*

Specifies the verbose mode of output, which includes more information.

*vproclist*

A list of one or more vproc identifier numbers. For examples, see [Definitions of Terms Used in Vproc Manager](#).

*vprocstate*

Indicates the PDE system state of a vproc, such as ONLINE, OFFLINE, or FATAL.

## Usage Notes

| If the VprocState of an AMP or PE is changed from ...               | After the next system restart, the VprocState/ ConfigStatus is ...                                            |
|---------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|
| either OFFLINE or FATAL to ONLINE, and its ConfigStatus is Down     | ONLINE/Online. The AMP will be in UTILITY/Catchup state during recovery, immediately prior to ONLINE /Online. |
| either OFFLINE or FATAL to NEWPROC, and its ConfigStatus is NewDown | NEWPROC/NewReady.                                                                                             |
| ONLINE to either OFFLINE or FATAL                                   | OFFLINE/Down or FATAL/Down.                                                                                   |
| NEWPROC to either OFFLINE or FATAL                                  | OFFLINE/NewDown or FATAL/NewDown.                                                                             |

### Note:

Any allowed VprocState transition becomes effective immediately. With the exception of AMPs being recovered using the Recovery Control Task, no changes are made to the ConfigStatus, the static message group, or whether the vproc is the Control AMP. These changes are deferred until the next system restart.

The SET *VprocList* command is governed by the allowable VprocState transitions shown in the following table.

| Current VprocState | Desired VprocState |                                                                |                                                         |                                     |                                                                 |              |
|--------------------|--------------------|----------------------------------------------------------------|---------------------------------------------------------|-------------------------------------|-----------------------------------------------------------------|--------------|
|                    | NONODE             | OFFLINE                                                        | ONLINE                                                  | NEWPROC                             | FATAL                                                           | UTILITY      |
| NONODE             | No change.         | Not allowed.                                                   | Not allowed.                                            | Not allowed.                        | Not allowed.                                                    | Not allowed. |
| OFFLINE            | Not allowed.       | No change.                                                     | Allowed.<br>If Config-Status = Down, initiate recovery. | Allowed if Config-Status = NewDown. | Allowed.                                                        | Not allowed. |
| ONLINE             | Not allowed.       | Allowed.<br>ConfigStatus will be Down after system restart.    | No change.                                              | Not allowed.                        | Allowed.<br>ConfigStatus will be Down after system restart.     | Not allowed. |
| NEWPROC            | Not allowed.       | Allowed.<br>ConfigStatus will be NewDown after system restart. | Not allowed.                                            | No change.                          | Allowed.<br>Config Status will be NewDown after system restart. | Not allowed. |
| FATAL              | Not allowed.       | Allowed.                                                       | Allowed.<br>If ConfigStatus = Down, initiate recovery.  | Allowed if ConfigStatus = NewDown.  | No change.                                                      | Not allowed. |
| UTILITY            | Not allowed.       | Not allowed.                                                   | Not allowed.                                            | Not allowed.                        | Not allowed.                                                    | No change.   |

### Example: Taking specified vprocs offline

This example shows how to take vprocs 0 and 1 offline.

Enter a command, HELP or QUIT:

```
set /v 0 to 1 offline
```

Vproc 0's VprocState has been set from ONLINE to OFFLINE.

NOTE: Vproc 0 will be forced down (i.e., its ConfigStatus will be Down) after the next system restart.

Vproc 1's VprocState has been set from ONLINE to OFFLINE.

NOTE: Vproc 1 will be forced down (i.e., its ConfigStatus will be Down) after the next system restart.

**Example: Setting a vproc online**

This example shows how to take vproc 1 online.

```
Enter a command, HELP or QUIT:
```

```
set /v 1 online
```

```
Vproc 1's VprocState has been set from OFFLINE to ONLINE.
```

```
NOTE: Vproc 1 will begin recovery in the background via  
the Recovery Control Task and will assume a  
VprocState/ConfigStatus of UTILITY/Catchup.  
Please refer to the System Console for  
additional status information.
```

## STATUS

The STATUS command reports the status of the database, PDE, and contiguous maps configuration.

### Syntax

```
{ STATUS | ST }
```

### Usage Notes

The Database Status Table includes a status row for each vproc that is defined on the system. When applicable, a footnote will follow the Database Status Table, indicating which AMP vproc has been selected as the Control AMP.

| Column                      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Vproc Number                | Uniquely identifies the vproc across the entire system.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Rel. Vproc#                 | Represents the number of the vproc relative to the Node upon which the vproc resides.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Node ID                     | Identifies the node upon which the vproc resides. The Node ID is formatted as CCC-MM, where CCC denotes the cabinet number, and MM denotes the module number.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Can Move                    | This indicates whether the vproc can be migrated to another node in its defined clique if its primary node fails.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Crash Count                 | Represents the number of times the vproc has crashed. The count increments with every attempted system restart. When the system restart succeeds, Crash Count is reset to 0 for all vprocs.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Vproc State                 | Represents the current PDE system state of the vproc.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Config Status               | Displays the Teradata Logical Configuration Map Status of the vproc.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| Config Type                 | Represents the Teradata Logical Configuration Map Type of a vproc.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Global Map Cluster/Host No. | Displays the Cluster Number if the Config Type is AMP or displays the Host No. if the Config Type is PE.<br>Cluster is the cluster number for the AMP as defined during a system configuration or reconfiguration. The valid range of cluster numbers is 0 to 8099.<br>Host No. is the host number that was assigned to the PE during a system configuration or reconfiguration. The valid range of host numbers is 1 to 1023.                                                                                                                                                                                                                                                                                                      |
| RcvJrnl/Host Type           | <ul style="list-style-type: none"> <li>Displays the RcvJrnl (that is, Recovery Journaling) flag if the Config Type is AMP. The RcvJrnl flag is Off if an AMP is down and the other AMPs in its cluster are not to create a recovery journal for the down AMP.</li> </ul> <p><b>Note:</b></p> <p>If you anticipate that an AMP will be down for a long period of time, then Teradata recommends an offline rebuild of all tables on the AMP (after the RcvJrnl flag has been set to Off).</p> <ul style="list-style-type: none"> <li>Displays the Host Type if the Config Type is PE. The Host Type is the host type for the PE as defined during a system configuration or reconfiguration, and is one of the following:</li> </ul> |

| Column    | Description                                                                                                                 |
|-----------|-----------------------------------------------------------------------------------------------------------------------------|
|           | <ul style="list-style-type: none"> <li>◦ IBM</li> <li>◦ COP</li> <li>◦ ATT3B</li> <li>◦ BULLHN</li> <li>◦ OS1100</li> </ul> |
| TVS Vproc | For AMP vprocs, displays the associated TVS vproc.                                                                          |

The map status table identifies the clustering used for the AMP vprocs with respect to the maps that determine how data is distributed among the AMPs. In the example output below, AMPs 0 and 2 constitute cluster 0 in all maps that include those AMPs. AMPs 1 and 3 constitute cluster 1.

The PDE Status Table includes a status row for each node that is defined on the system, as follows:

| Column        | Description                                                                                                                                                                                                                                           |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Node ID       | The Node ID as defined using the Parallel Upgrade Tool (PUT). For more information, see the <i>Parallel Upgrade Tool (PUT) Reference</i> . The Node ID is formatted as CCC-MM, where CCC denotes the cabinet number and MM denotes the module number. |
| Node State    | The current state of the node, which is either ONLINE, DOWN, or STANDBY.                                                                                                                                                                              |
| Clique Number | The clique number for the node as defined using the Parallel Upgrade Tool (PUT). For more information, see the <i>Parallel Upgrade Tool (PUT) Reference</i> .                                                                                         |
| CPUs          | The number of CPUs on the node.                                                                                                                                                                                                                       |
| Memory (MB)   | The total memory size in megabytes for the node (rounded up to the nearest integer).                                                                                                                                                                  |
| CHANs         | The number of channels attached to the node.                                                                                                                                                                                                          |
| LANs          | The number of LANs attached to the node.                                                                                                                                                                                                              |
| AMPs          | The number of AMPs running on the node.                                                                                                                                                                                                               |
| Node Name     | The network name for the node.                                                                                                                                                                                                                        |

When applicable, a footnote will follow the PDE Status Table, indicating which node is defined as a Non-TPA Node (for STANDBY node).

### Example: Displaying the database, map, and PDE status

```
Enter a command, HELP or QUIT:
status
```

```
SYSTEM NAME: teradata-1
```

#### DBS LOGICAL CONFIGURATION

| Vproc<br>Number | Rel.<br>Vproc# | Node<br>ID | Can<br>Move | Crash<br>Count | Vproc<br>State | Config<br>Status | Config<br>Type | Global<br>Map<br>Cluster/<br>Host No. | Rcv<br>Jrn1/<br>Host<br>Type | TVS<br>Vproc |
|-----------------|----------------|------------|-------------|----------------|----------------|------------------|----------------|---------------------------------------|------------------------------|--------------|
| -----           | -----          | -----      | -----       | -----          | -----          | -----            | -----          | -----                                 | -----                        | -----        |

|       |    |      |     |   |        |        |     |   |     |       |
|-------|----|------|-----|---|--------|--------|-----|---|-----|-------|
| 0*    | 1  | 1-01 | Yes | 0 | ONLINE | Online | AMP | 0 | On  | 28671 |
| 1     | 2  | 1-01 | Yes | 0 | ONLINE | Online | AMP | 1 | On  | 28671 |
| 2     | 3  | 1-01 | Yes | 0 | ONLINE | Online | AMP | 0 | On  | 28670 |
| 3     | 4  | 1-01 | Yes | 0 | ONLINE | Online | AMP | 1 | On  | 28670 |
| 26623 | 8  | 1-01 | No  | 0 | ONLINE | N/A    | RSG | 0 | N/A | N/A   |
| 22528 | 7  | 1-01 | No  | 0 | ONLINE | N/A    | GTW | 1 | COP | N/A   |
| 28670 | 9  | 1-01 | Yes | 0 | ONLINE | N/A    | TVS | 0 | N/A | N/A   |
| 28671 | 10 | 1-01 | Yes | 0 | ONLINE | N/A    | TVS | 0 | N/A | N/A   |
| 30718 | 5  | 1-01 | Yes | 0 | ONLINE | Online | PE  | 1 | COP | N/A   |
| 30719 | 6  | 1-01 | Yes | 0 | ONLINE | Online | PE  | 1 | COP | N/A   |

\* DBS Control AMP

DBS State: Logons are enabled - The system is quiescent  
DBS RestartKind: COLD

#### MAP CONFIGURATION

Slot 0: TD\_GlobalMap  
MAP State: Online

Slot 1: TD\_DataDictionaryMap  
MAP State: Online

Slot 2: TD\_Map1  
MAP State: Online

| AMP<br>Number | Slot 0<br>Cluster<br>No | Slot 1<br>Cluster<br>No | Slot 2<br>Cluster<br>No |
|---------------|-------------------------|-------------------------|-------------------------|
| 0             | 0                       | 0                       | 0                       |
| 1             | 1                       | 1                       | 1                       |
| 2             | 0                       | 0                       | 0                       |
| 3             | 1                       | 1                       | 1                       |

#### PDE PHYSICAL CONFIGURATION

| Node<br>ID | Node<br>State | Clique<br>Number | CPUs | Memory<br>(MB) | CHANS | LANs | AMPs | Node Name |
|------------|---------------|------------------|------|----------------|-------|------|------|-----------|
| 1-01       | ONLINE        | 0                | 1    | 7740           | 0     | 1    | 4    | localhost |

PDE State: RUN/STARTED

## STATUS DBS

The STATUS DBS command reports the status of the database.

### Syntax

```
{ STATUS | ST } DBS
```

### Usage Notes

The STATUS DBS command returns a status report for the Database Status Table only. The Database Status Table includes a status row for each vproc that is defined on the system. When applicable, a footnote will follow the Database Status Table, indicating which AMP vproc has been selected as the Control AMP.

#### Vproc Status Columns

| Column            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Vproc Number      | Uniquely identifies the vproc across the entire system.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Rel. Vproc#       | Represents the number of the vproc number relative to the Node upon which the vproc resides.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Node ID           | Identifies the Node upon which the vproc resides. The Node ID is formatted as CCC-MM, where CCC denotes the cabinet number and MM denotes the module number.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Can Move          | This indicates whether the vproc can be migrated to another node in its defined clique if its primary node fails.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Crash Count       | Represents the number of times the vproc has crashed.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Vproc State       | Represents the current PDE system state of a vproc.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Config Status     | Displays the Teradata Logical Configuration Map Status of a vproc.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Config Type       | Represents the Teradata Logical Configuration Map Type of a vproc.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Cluster/Host No.  | Displays the Cluster Number if the Config Type is AMP or displays the Host No. if the Config Type is PE.<br>Cluster is the cluster number for the AMP as defined during a system configuration or reconfiguration. The valid range of cluster numbers is 0 to 8099.<br>Host No. is the host number that was assigned to the PE during a system configuration or reconfiguration. The valid range of host numbers is 1 to 1023.                                                                                                                                                                                                                                                                                                      |
| RcvJrnl/Host Type | <ul style="list-style-type: none"> <li>Displays the RcvJrnl (that is, Recovery Journaling) flag if the Config Type is AMP. The RcvJrnl flag is Off if an AMP is down and the other AMPs in its cluster are not to create a recovery journal for the down AMP.</li> </ul> <p><b>Note:</b></p> <p>If you anticipate that an AMP will be down for a long period of time, then Teradata recommends an offline rebuild of all tables on the AMP (after the RcvJrnl flag has been set to Off).</p> <ul style="list-style-type: none"> <li>Displays the Host Type if the Config Type is PE. The Host Type is the host type for the PE as defined during a system configuration or reconfiguration, and is one of the following:</li> </ul> |



| Column    | Description                                                                                                           |
|-----------|-----------------------------------------------------------------------------------------------------------------------|
|           | <ul style="list-style-type: none"><li>◦ IBM</li><li>◦ COP</li><li>◦ ATT3B</li><li>◦ BULLHN</li><li>◦ OS1100</li></ul> |
| TVS Vproc | For AMP vprocs, displays the associated TVS vproc.                                                                    |

**Example: Displaying the status of the Teradata system**

```
Enter a command, HELP or QUIT:

status dbs

SYSTEM NAME: teradata-1

                DBS LOGICAL CONFIGURATION
                -----

Vproc  Rel.  Node  Can  Crash Vproc  Config  Config Cluster/ Host  TVS
Number Vproc# ID   Move Count State Status Type  Host No. Type  Vproc
-----
    0*    1    1-01  Yes    0  ONLINE  Online   AMP      0  On  10238
      1    2    1-01  Yes    0  ONLINE  Online   AMP      0  On  10237
   8192    4    1-01  No     0  ONLINE  N/A      GTW      1  COP  N/A
  10237    5    1-01  Yes    0  ONLINE  N/A      TVS      0  N/A  N/A
  10238    6    1-01  Yes    0  ONLINE  N/A      TVS      0  N/A  N/A
  16383    3    1-01  Yes    0  ONLINE  Online   PE       1  COP  N/A
-----
*   DBS Control AMP

DBS State: Logons are enabled - The system is quiescent

DBS RestartKind: COLD
```

## STATUS DBS ORDERBY CN

The STATUS DBS ORDERBY CN command reports the status of Online and NotOnline Vantage AMP vprocs, optionally ordered by cluster number.

### Syntax

```
{ STATUS | ST } DBS [ ORDERBY CN ]
```

### Usage Notes

The STATUS DBS Orderby CN command returns the Online and NotOnline Vantage AMP vprocs ordered by the cluster status tables only. When applicable, a footnote will follow the Database Status Table, indicating which AMP vproc has been selected as the Control AMP.

#### ONLINE vproc Columns

| Column          | Description                                                                                                                                                                                                                                                                                                                  |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Vproc-Ids Range | Uniquely identifies the Vproc-id or Vproc-ids individually or in a range across the entire system. Vproc numbers in the range of 0 through 30719 are used exclusively by the Vantage configuration. Vproc numbers greater than 30719 are used exclusively by PDE and are not included in the Teradata logical configuration. |
| Crash Count     | Represents the number of times the vproc has crashed.                                                                                                                                                                                                                                                                        |
| CN No.          | Displays the cluster number for the AMP as defined during a system configuration or reconfiguration. The valid range of cluster numbers is 0 to 8099.                                                                                                                                                                        |

#### NOTONLINE vproc Columns

| Column          | Description                                                                                                                                                                                                                                                                                                                  |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Vproc-Ids Range | Uniquely identifies the Vproc-id or Vproc-ids individually or in a range across the entire system. Vproc numbers in the range of 0 through 30719 are used exclusively by the Vantage configuration. Vproc numbers greater than 30719 are used exclusively by PDE and are not included in the Teradata logical configuration. |
| Crash Count     | Represents the number of times the vproc has crashed.                                                                                                                                                                                                                                                                        |
| Vproc State     | Represents the current PDE system state of a vproc if the state is not online.                                                                                                                                                                                                                                               |
| Config Status   | Displays the Teradata Logical Configuration Map Status of a vproc.                                                                                                                                                                                                                                                           |
| Cluster Number  | Displays the cluster number for the AMP as defined during a system configuration or reconfiguration. The valid range of cluster numbers is 0 to 8099.                                                                                                                                                                        |

### Example: Displaying a cluster-ordered list of AMP status

```
Enter a command, HELP or QUIT:
status dbs orderby cn
```

SYSTEM NAME: ippd

DBS LOGICAL CONFIGURATION  
-----

ONLINE AMP Vprocs ordered by Cluster No. -

| Vproc-Ids | Crash | CN   | Vproc-Ids | Crash | CN   |
|-----------|-------|------|-----------|-------|------|
| Range     | Cnt   | NO   | Range     | Cnt   | NO   |
| -----     | ---   | ---- | -----     | ---   | ---- |
| 0*-1      | 0     | 0    | 2-3       | 0     | 1    |
| 4-5       | 0     | 2    | 6-7       | 0     | 3    |
| -----     | ---   | ---- | -----     | ---   | ---- |

\* DBS Control AMP

DBS State: Logons are enabled - Users are logged on

DBS RestartKind: COLD

## STATUS DBS ORDERBY HN

The STATUS DBS ORDERBY HN command reports the status of Online and NotOnline Vantage PE vprocs, optionally ordered by host number.

### Syntax

```
{ STATUS | ST } DBS [ ORDERBY HN ]
```

### Usage Notes

The STATUS DBS ORDERBY HN command is used to return the Online and NotOnline Vantage PE vprocs ordered by the host status tables only.

#### ONLINE vproc Columns

| Column          | Description                                                                                                                                                                                                                                                                                                                  |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Vproc-Ids Range | Uniquely identifies the Vproc-id or Vproc-ids individually or in a range across the entire system. Vproc numbers in the range of 0 through 30719 are used exclusively by the Vantage configuration. Vproc numbers greater than 30719 are used exclusively by PDE and are not included in the Teradata logical configuration. |
| Crash Count     | Represents the number of times the vproc has crashed.                                                                                                                                                                                                                                                                        |
| Can Move        | Indicates whether the PE vproc can be migrated to another node in its defined clique if its primary node fails.                                                                                                                                                                                                              |
| Host Number     | Displays the host number that was assigned to the PE during a system configuration or reconfiguration. The valid range of host numbers is 1 to 1023.                                                                                                                                                                         |
| Host Type       | Displays the host type for the PE as defined during a system configuration or reconfiguration, and is one of the following: IBM, COP, ATT3B, BULLHN, or OS1100.                                                                                                                                                              |

#### NOTONLINE vproc Columns

| Column          | Description                                                                                                                                                                                                                                                                                                                  |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Vproc-Ids Range | Uniquely identifies the Vproc-id or Vproc-ids individually or in a range across the entire system. Vproc numbers in the range of 0 through 30719 are used exclusively by the Vantage configuration. Vproc numbers greater than 30719 are used exclusively by PDE and are not included in the Teradata logical configuration. |
| Crash Count     | Represents the number of times the vproc has crashed.                                                                                                                                                                                                                                                                        |
| Can Move        | Indicates whether the PE vproc can be migrated to another node in its defined clique if its primary node fails.                                                                                                                                                                                                              |
| Vproc State     | Represents the current PDE system state of a vproc if the state is not online.                                                                                                                                                                                                                                               |
| Config Status   | Displays the Teradata Logical Configuration Map Status of a vproc.                                                                                                                                                                                                                                                           |
| Host Number     | Displays the host number that was assigned to the PE during a system configuration or reconfiguration. The valid range of host numbers is 1 to 1023.                                                                                                                                                                         |

| Column    | Description                                                                                                                                                    |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Host Type | Displays the host type for the PE as defined during a system configuration or reconfiguration and is one of the following: IBM, COP, ATT3B, BULLHN, or OS1100. |

**Example:Displaying a host-ordered list of PE status**

```
Enter a command, HELP or QUIT:
status dbs orderby hn

SYSTEM NAME: nirvana

                                DBS LOGICAL CONFIGURATION
                                -----
ONLINE PE vprocs ordered by Host No. - -
Vproc-Ids      Crash      Host      Host
Range          Count      Moveable   Number   Type
-----
16368-16382      0        Yes        52       COP
16345-16351      0        No         282      IBM
16353, 16357, 16359, 0        No         286      IBM
.
.
.
16362, 16366
-----
NOTONLINE PE vprocs ordered by Host No. - -
Vproc-Ids      Crash      Vproc      Config      Host      Host
Range          Count      Can Move   State       Status     Number    Type
-----
16383           0        Yes        NONODE      Down       52        COP
16344           0        No         NONODE      Down       282       IBM
16355, 16367    0        No         NONODE      Down       286       IBM
16352, 16364    0        No         NONODE      Down       289       IBM
-----
DBS State: Logons are enabled - Users are logged on
DBS RestartKind: COLD
```

## STATUS DBS ORDERBY NODEID

The STATUS DBS ORDERBY NODEID command reports the status of Online and NotOnline Vantage vprocs, optionally ordered by node id.

### Syntax

```
{ STATUS | ST } DBS [ ORDERBY NODEID ]
```

### Usage Notes

The STATUS DBS *Orderby Nodeid* command is used to return the Online and NotOnline Vantage vprocs ordered by the node id status tables only. When applicable, a footnote will follow the Database Status Table, indicating which AMP vproc has been selected as the Control AMP.

#### ONLINE vproc Columns

| Column          | Description                                                                                                                                                                                                                                                                                                                  |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Vproc-Ids Range | Uniquely identifies the Vproc-id or Vproc-ids individually or in a range across the entire system. Vproc numbers in the range of 0 through 30719 are used exclusively by the Vantage configuration. Vproc numbers greater than 30719 are used exclusively by PDE and are not included in the Teradata logical configuration. |
| Crash Count     | Represents the number of times the vproc crashed.                                                                                                                                                                                                                                                                            |
| Node Ids        | Identifies the Node upon which the vproc resides. The Node ID is formatted as CCC-MM, where CCC denotes the cabinet number and MM denotes the module number.                                                                                                                                                                 |

#### NOTONLINE vproc Columns

| Column          | Description                                                                                                                                                                                                                                                                                                                  |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Vproc-Ids Range | Uniquely identifies the Vproc-id or Vproc-ids individually or in a range across the entire system. Vproc numbers in the range of 0 through 30719 are used exclusively by the Vantage configuration. Vproc numbers greater than 30719 are used exclusively by PDE and are not included in the Teradata logical configuration. |
| Crash Count     | Represents the number of times the vproc crashed.                                                                                                                                                                                                                                                                            |
| Vproc State     | Represents the current PDE system state of a vproc if the state is not online.                                                                                                                                                                                                                                               |
| Config Status   | Displays the Teradata Logical Configuration Map Status of a vproc.                                                                                                                                                                                                                                                           |
| Node Ids        | Identifies the Node upon which the vproc resides. The Node ID is formatted as CCC-MM, where CCC denotes the cabinet number and MM denotes the module number.                                                                                                                                                                 |

### Example: Displaying a node-ID-ordered list of vproc status

Enter a command, HELP or QUIT:

```
status dbs orderby nodeid
```

SYSTEM NAME: nirvana

DBS LOGICAL CONFIGURATION

ONLINE AMP and PE vprocs ordered by Node ID - -

| Vproc-Ids<br>Range | Crash<br>Count | Node<br>Ids | Vproc-Ids<br>Range | Crash<br>Count | Node<br>Ids |
|--------------------|----------------|-------------|--------------------|----------------|-------------|
| 3, 7, 11, 15       | 0              | 9-00        | 16366              | 0              | 9-00        |
| 2, 6, 10, 14       | 0              | 9-02        | 1, 5, 9, 13        | 0              | 10-00       |
| 16362-16363, 16381 | 0              | 10-00       | 0*, 4, 8, 12       | 0              | 10-02       |
| .                  |                |             |                    |                |             |
| .                  |                |             |                    |                |             |
| .                  |                |             |                    |                |             |
| 16369              | 0              | 17-00       | 16368              | 0              | 17-02       |

NOTONLINE AMP and PE vprocs orderby Node ID - -

| Vproc-Ids<br>Range | Crash<br>Count | Vproc<br>State | Config<br>Status | Node<br>IDs |
|--------------------|----------------|----------------|------------------|-------------|
| 16367, 16383       | 0              | NONODE         | Down             | 9-00        |
| 16364              | 0              | NONODE         | Down             | 9-02        |
| 16355              | 0              | NONODE         | Down             | 11-02       |
| 17                 | 0              | UTILITY        | CATCHUP          | 11-02       |
| 21                 | 0              | NEWPROC        | NewReady         | 11-02       |
| .                  |                |                |                  |             |
| .                  |                |                |                  |             |
| .                  |                |                |                  |             |
| .                  |                |                |                  |             |
| 63                 | 0              | FATAL**        | Down             | 15          |

- \* DBS Control AMP
- \*\* The storage associated with the AMP has an I/O error or can't be opened. Vproc can be brought online after fixing the I/O error and restarting the database.

# STATUS MAP

The STATUS MAP command reports the contiguous map configuration, including a list of the maps, their slots, and the AMP clusters defined by the maps.

## Syntax

```
{ STATUS | ST } MAP
```

## Example: Displaying the contiguous maps configuration

Enter a command, HELP or QUIT:  
status map

SYSTEM NAME: localhost17/03/17 14:03:37

MAP CONFIGURATION  
-----

Slot 0: TD\_GlobalMap  
MAP State: Online

Slot 1: TD\_DataDictionaryMap  
MAP State: Online

Slot 2: TD\_Map1  
MAP State: Online

Slot 3: TD\_Map2  
MAP State: Online

| AMP    | Slot 0  | Slot 1  | Slot 2  | Slot 3  |
|--------|---------|---------|---------|---------|
| Number | Cluster | Cluster | Cluster | Cluster |
|        | No      | No      | No      | No      |
| 0      | 0       | 0       | 0       | 0       |
| 1      | 0       | 0       | 0       | 0       |
| 2      | 1       | 1       | 1       | 1       |
| 3      | 1       | 1       | 1       | 1       |



## STATUS NOTONLINE

The STATUS NOTONLINE command reports the status row for vprocs and nodes that are not fully online.

### Syntax

```
{ STATUS | ST } { NOTONLINE | NO }
```

### Usage Notes

The following table shows the columns of the vproc status row.

| Column                      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Vproc Number                | Uniquely identifies the vproc across the entire system.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Rel. Vproc#                 | Represents the number of the vproc relative to the Node upon which the vproc resides.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Node ID                     | Identifies the node upon which the vproc resides. The Node ID is formatted as CCC-MM, where CCC denotes the cabinet number, and MM denotes the module number.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Can Move                    | This indicates whether the vproc can be migrated to another node in its defined clique if its primary node fails.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Crash Count                 | Represents the number of times the vproc has crashed. The count increments with every attempted system restart. When the system restart succeeds, Crash Count is reset to 0 for all vprocs.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Vproc State                 | Represents the current PDE system state of the vproc.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Config Status               | Displays the Teradata Logical Configuration Map Status of the vproc.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| Config Type                 | Represents the Teradata Logical Configuration Map Type of a vproc.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Global Map Cluster/Host No. | Displays the Cluster Number if the Config Type is AMP or displays the Host No. if the Config Type is PE.<br>Cluster is the cluster number for the AMP as defined during a system configuration or reconfiguration. The valid range of cluster numbers is 0 to 8099.<br>Host No. is the host number that was assigned to the PE during a system configuration or reconfiguration. The valid range of host numbers is 1 to 1023.                                                                                                                                                                                                                                                                                                                                                                                      |
| RcvJrnl/Host Type           | <ul style="list-style-type: none"> <li>Displays the RcvJrnl (that is, Recovery Journaling) flag if the Config Type is AMP. The RcvJrnl flag is Off if an AMP is down and the other AMPs in its cluster are not to create a recovery journal for the down AMP.</li> </ul> <p><b>Note:</b></p> <p>If you anticipate that an AMP will be down for a long period of time, then Teradata recommends an offline rebuild of all tables on the AMP (after the RcvJrnl flag has been set to Off).</p> <ul style="list-style-type: none"> <li>Displays the Host Type if the Config Type is PE. The Host Type is the host type for the PE as defined during a system configuration or reconfiguration, and is one of the following:             <ul style="list-style-type: none"> <li>IBM</li> <li>COP</li> </ul> </li> </ul> |

| Column    | Description                                                                               |
|-----------|-------------------------------------------------------------------------------------------|
|           | <ul style="list-style-type: none"><li>◦ ATT3B</li><li>◦ BULLHN</li><li>◦ OS1100</li></ul> |
| TVS Vproc | For AMP vprocs, displays the associated TVS vproc.                                        |

**Example: Displaying a list of vprocs and nodes that are not online**

```
Enter a command, HELP, or QUIT:
status notonline

SYSTEM NAME: teradata-1

                DBS LOGICAL CONFIGURATION
                -----

Vproc  Rel.  Node   Can   Crash Vproc  Config  Config  Global  Rcv
Number Vproc# ID    Move  Count State  Status  Type   Map    Jrn1/
-----
      5    6   2-14  Yes    0  OFFLINE Online  AMP    2    On   10237
-----
```

## STATUS ONLINE

The STATUS ONLINE command reports whether all AMPs are online. If any AMP is not online, STATUS ONLINE shows a more detailed list of status for online vprocs.

### Syntax

```
{ STATUS | ST } { ONLINE | ON }
```

### Usage Notes

If all vprocs are online, the STATUS ONLINE command simply reports that.

If one or more vprocs are offline, the STATUS ONLINE command displays a STATUS row for each of the vprocs that is online. When applicable, a footnote will follow the Database Status Table, indicating which AMP vproc has been selected as the Control AMP.

#### Vproc Status Columns

| Column                      | Description                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Vproc Number                | Uniquely identifies the vproc across the entire system.                                                                                                                                                                                                                                                                                                                                                                        |
| Rel. Vproc#                 | Represents the number of the vproc relative to the Node upon which the vproc resides.                                                                                                                                                                                                                                                                                                                                          |
| Node ID                     | Identifies the node upon which the vproc resides. The Node ID is formatted as CCC-MM, where CCC denotes the cabinet number, and MM denotes the module number.                                                                                                                                                                                                                                                                  |
| Can Move                    | This indicates whether the vproc can be migrated to another node in its defined clique if its primary node fails.                                                                                                                                                                                                                                                                                                              |
| Crash Count                 | Represents the number of times the vproc has crashed. The count increments with every attempted system restart. When the system restart succeeds, Crash Count is reset to 0 for all vprocs.                                                                                                                                                                                                                                    |
| Vproc State                 | Represents the current PDE system state of the vproc.                                                                                                                                                                                                                                                                                                                                                                          |
| Config Status               | Displays the Teradata Logical Configuration Map Status of the vproc.                                                                                                                                                                                                                                                                                                                                                           |
| Config Type                 | Represents the Teradata Logical Configuration Map Type of a vproc.                                                                                                                                                                                                                                                                                                                                                             |
| Global Map Cluster/Host No. | Displays the Cluster Number if the Config Type is AMP or displays the Host No. if the Config Type is PE.<br>Cluster is the cluster number for the AMP as defined during a system configuration or reconfiguration. The valid range of cluster numbers is 0 to 8099.<br>Host No. is the host number that was assigned to the PE during a system configuration or reconfiguration. The valid range of host numbers is 1 to 1023. |
| RcvJrnl/Host Type           | <ul style="list-style-type: none"> <li>Displays the RcvJrnl (that is, Recovery Journaling) flag if the Config Type is AMP. The RcvJrnl flag is Off if an AMP is down and the other AMPs in its cluster are not to create a recovery journal for the down AMP.</li> </ul>                                                                                                                                                       |

| Column    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|           | <p><b>Note:</b></p> <p>If you anticipate that an AMP will be down for a long period of time, then Teradata recommends an offline rebuild of all tables on the AMP (after the RcvJrnl flag has been set to Off).</p> <ul style="list-style-type: none"> <li>Displays the Host Type if the Config Type is PE. The Host Type is the host type for the PE as defined during a system configuration or reconfiguration, and is one of the following: <ul style="list-style-type: none"> <li>IBM</li> <li>COP</li> <li>ATT3B</li> <li>BULLHN</li> <li>OS1100</li> </ul> </li> </ul> |
| TVS Vproc | For AMP vprocs, displays the associated TVS vproc.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

The PDE Status Table includes a status row for each node that is defined on the system. The columns of the node status row are shown in the following table.

| Column        | Description                                                                                                                                                                                                                                           |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Node ID       | The Node ID as defined using the Parallel Upgrade Tool (PUT). For more information, see the <i>Parallel Upgrade Tool (PUT) Reference</i> . The Node ID is formatted as CCC-MM, where CCC denotes the cabinet number and MM denotes the module number. |
| Node State    | The current state of the node, which is either ONLINE, DOWN, or STANDBY.                                                                                                                                                                              |
| Clique Number | The clique number for the node as defined using the Parallel Upgrade Tool (PUT). For more information, see the <i>Parallel Upgrade Tool (PUT) Reference</i> .                                                                                         |
| CPUs          | The number of CPUs on the node.                                                                                                                                                                                                                       |
| Memory (MB)   | The total memory size in megabytes for the node (rounded up to the nearest integer).                                                                                                                                                                  |
| CHANS         | The number of channels attached to the node.                                                                                                                                                                                                          |
| LANs          | The number of LANs attached to the node.                                                                                                                                                                                                              |
| AMPs          | The number of AMPs running on the node.                                                                                                                                                                                                               |
| Node Name     | The network name for the node.                                                                                                                                                                                                                        |

When applicable, a footnote will follow the PDE Status Table, indicating which node is defined as a Non-TPA Node (for STANDBY node).

### Example: Displaying the status of online vprocs

If all vprocs are online, the STATUS ONLINE command simply reports that.

```
Enter a command, HELP or QUIT:
status online

SYSTEM NAME: ztest

All DBS vprocs are fully online.

All PDE nodes are fully online.
```

If any vproc is not online, the STATUS ONLINE command displays a more detailed status screen for all vprocs and nodes that are ONLINE. In the following example, vproc number 2 has been disabled and is offline, so it is not shown in the list of online vprocs.

```
Enter a command, HELP or QUIT:
status online

SYSTEM NAME: localhost

                                DBS LOGICAL CONFIGURATION
                                -----

                                Global   Rcv
                                Map      Jrn1/
Vproc Rel.  Node   Can   Crash Vproc  Config  Config Cluster/ Host  TVS
Number Vproc# ID     Move  Count State  Status  Type   Host No.  Type  Vproc
-----
   0*    1    1-01  Yes    0  ONLINE Online  AMP     0   On  28671
   1     2    1-01  Yes    0  ONLINE Online  AMP     1   On  28671
   3     4    1-01  Yes    0  ONLINE Online  AMP     1   On  28670
26623    8    1-01  No     0  ONLINE N/A     RSG     0  N/A  N/A
22528    7    1-01  No     0  ONLINE N/A     GTW     1  COP  N/A
28670    9    1-01  Yes    0  ONLINE N/A     TVS     0  N/A  N/A
28671   10    1-01  Yes    0  ONLINE N/A     TVS     0  N/A  N/A
30718    5    1-01  Yes    0  ONLINE Online  PE      1  COP  N/A
30719    6    1-01  Yes    0  ONLINE Online  PE      1  COP  N/A
-----
*   DBS Control AMP

    DBS State: Logons are enabled - The system is quiescent

    DBS RestartKind: COLD

All PDE nodes are fully online.
```



# STATUS PDE

The STATUS PDE command reports the status of the PDE.

## Syntax

```
{ STATUS | ST } PDE
```

## Usage Notes

The STATUS PDE command returns the PDE Status Table only. The PDE Status Table includes a status row for each node that is defined on the system.

### Node Status Columns

| Column        | Description                                                                                                                                                                                                                                           |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Node ID       | The Node ID as defined using the Parallel Upgrade Tool (PUT). For more information, see the <i>Parallel Upgrade Tool (PUT) Reference</i> . The Node ID is formatted as CCC-MM, where CCC denotes the cabinet number and MM denotes the module number. |
| Node State    | The current state of the node, which is either ONLINE, DOWN, or STANDBY.                                                                                                                                                                              |
| Clique Number | The clique number for the node as defined using the Parallel Upgrade Tool (PUT). For more information, see the <i>Parallel Upgrade Tool (PUT) Reference</i> .                                                                                         |
| CPUs          | The number of CPUs on the node.                                                                                                                                                                                                                       |
| Memory (MB)   | The total memory size in megabytes for the node (rounded up to the nearest integer).                                                                                                                                                                  |
| CHANs         | The number of channels attached to the node.                                                                                                                                                                                                          |
| LANs          | The number of LANs attached to the node.                                                                                                                                                                                                              |
| AMPs          | The number of AMPs running on the node.                                                                                                                                                                                                               |
| Node Name     | The network name for the node.                                                                                                                                                                                                                        |

## Example: Displaying the status of PDE

```
SYSTEM NAME: default_vconfig

                                PDE PHYSICAL CONFIGURATION
                                -----

Node   Node   Clique   Memory
ID     State  Number  CPUs   (MB)   CHANs  LANs  AMPs  Node Name
-----
-----
  4-06  ONLINE    0       4    2043    0      1     4  puthod1_bynet
  4-07  ONLINE    0       2    2043    0      1     4  puthod2_bynet
```

|                        |            |               |             |      |       |      |      |               |
|------------------------|------------|---------------|-------------|------|-------|------|------|---------------|
| 4-08                   | ONLINE     | 0             | 2           | 2043 | 0     | 1    | 4    | puthod3_bynet |
| -----                  |            |               |             |      |       |      |      |               |
| Node ID                | Node State | Clique Number | Memory CPUs | (MB) | CHANs | LANs | AMPs | Node Name     |
| -----                  |            |               |             |      |       |      |      |               |
| 4-09^                  | STANDBY    | 0             | 0           | 0    | 0     | 1    | 4    | puthod4_bynet |
| -----                  |            |               |             |      |       |      |      |               |
| ^ Non-TPA Node         |            |               |             |      |       |      |      |               |
| PDE State: RUN/STARTED |            |               |             |      |       |      |      |               |



## STATUS RESTART

The STATUS RESTART command reports the current system RestartKind setting.

### Syntax

```
{ STATUS | ST } { RESTART | R }
```

### Example: Displaying the current restart kind setting

Enter a command, HELP, or QUIT:

```
status restart
```

```
DBS RestartKind = COLD.
```

## STATUS SYSSTATE

The STATUS SYSSTATE command reports the current system state and the System Debugger, if it is attached.

### Syntax

```
{ STATUS | ST } { SYSSTATE | SYSS }
```

### Example: Displaying the current database and PDE status

Enter a command, HELP, or QUIT:

```
status sysstate
```

```
DBS State: Logons are enabled - The system is quiescent
```

```
PDE State: RUN/STARTED
```

## STATUS *vproclst*

The STATUS *vproclst* command reports the vproc status table row for the specified vproc or vprocs and allow a maximum of 30720 or 16384 vprocs, depending on the system, to be specified. This command does not display a vproc status row for undefined vprocs.

### Syntax

```
{ STATUS | ST } vproclst
```

### Syntax Elements

#### *vproclst*

A list of one or more vproc identifiers. For examples, see [Definitions of Terms Used in Vproc Manager](#).

### Usage Notes

#### Vproc Status Columns

| Column                      | Description                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Vproc Number                | Uniquely identifies the vproc across the entire system.                                                                                                                                                                                                                                                                                                                                                                        |
| Rel. Vproc#                 | Represents the number of the vproc number relative to the Node upon which the vproc resides.                                                                                                                                                                                                                                                                                                                                   |
| Node ID                     | Identifies the Node upon which the vproc resides. The Node ID is formatted as CCC-MM, where CCC denotes the cabinet number and MM denotes the module number.                                                                                                                                                                                                                                                                   |
| Can Move                    | This indicates whether the vproc can be migrated to another node in its defined clique if its primary node fails.                                                                                                                                                                                                                                                                                                              |
| Crash Count                 | Represents the number of times the vproc has crashed.                                                                                                                                                                                                                                                                                                                                                                          |
| Vproc State                 | Represents the current PDE system state of a vproc.                                                                                                                                                                                                                                                                                                                                                                            |
| Config Status               | Displays the Teradata Logical Configuration Map Status of a vproc.                                                                                                                                                                                                                                                                                                                                                             |
| Config Type                 | Represents the Teradata Logical Configuration Map Type of a vproc.                                                                                                                                                                                                                                                                                                                                                             |
| Global Map Cluster/Host No. | Displays the Cluster Number if the Config Type is AMP or displays the Host No. if the Config Type is PE.<br>Cluster is the cluster number for the AMP as defined during a system configuration or reconfiguration. The valid range of cluster numbers is 0 to 8099.<br>Host No. is the host number that was assigned to the PE during a system configuration or reconfiguration. The valid range of host numbers is 1 to 1023. |
| RcvJrnl/Host Type           | <ul style="list-style-type: none"> <li>Displays the RcvJrnl (that is, Recovery Journaling) flag if the Config Type is AMP. The RcvJrnl flag is Off if an AMP is down and the other AMPs in its cluster are not to create a recovery journal for the down AMP.</li> </ul>                                                                                                                                                       |

| Column    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|           | <p><b>Note:</b></p> <p>If you anticipate that an AMP will be down for a long period of time, then Teradata recommends an offline rebuild of all tables on the AMP (after the RcvJrnl flag has been set to Off).</p> <ul style="list-style-type: none"><li>• Displays the Host Type if the Config Type is PE. The Host Type is the host type for the PE as defined during a system configuration or reconfiguration, and is one of the following:<ul style="list-style-type: none"><li>◦ IBM</li><li>◦ COP</li><li>◦ ATT3B</li><li>◦ BULLHN</li><li>◦ OS1100</li></ul></li></ul> |
| TVS Vproc | For AMP vprocs, displays the associated TVS vproc.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

**Example: Displaying the status of specified vprocs**

```
Enter a command, HELP or QUIT:
status 1, 2, 8192 to 16383

SYSTEM NAME: teradata-1

                                DBS LOGICAL CONFIGURATION
                                -----

Vproc  Rel.  Node   Can   Crash Vproc  Config  Config  Global  Rcv
Number Vproc# ID    Move Count State Status Type   Map    Jrnl/
-----
    0*    1    1-01  Yes     0  ONLINE Online   AMP     0    On   10238
    1     2    1-01  Yes     0  ONLINE Online   AMP     0    On   10237
   8192    4    1-01  No      0  ONLINE  N/A     GTW     1    COP  N/A
  10237    5    1-01  Yes     0  ONLINE  N/A     TVS     0    N/A  N/A
  10238    6    1-01  Yes     0  ONLINE  N/A     TVS     0    N/A  N/A
  16383    3    1-01  Yes     0  ONLINE Online   PE      1    COP  N/A
-----
*   DBS Control AMP

    DBS State: Logons are enabled - The system is quiescent

    DBS RestartKind: COLD
```

# How to Read Syntax

This document uses the following syntax conventions.

| Syntax Convention | Meaning                                                                                                                                                                                                                                                                                                                                                                  |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| KEYWORD           | Keyword. Spell exactly as shown.<br>Many environments are case-insensitive. Syntax shows keywords in uppercase unless operating system restrictions require them to be lowercase or mixed-case.                                                                                                                                                                          |
| <i>variable</i>   | Variable. Replace with actual value.                                                                                                                                                                                                                                                                                                                                     |
| <i>number</i>     | String of one or more digits. Do not use commas in numbers with more than three digits.<br>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.<br>Example: x x x<br>See note after table.                                                                                                                                                                                                                                                                         |
| x [, ...]         | You can repeat x, separating occurrences with commas.<br>Example: x, x, x<br>See note after table.                                                                                                                                                                                                                                                                       |
| x [delimiter...]  | You can repeat x, separating occurrences with specified delimiter.<br>Examples:<br><ul style="list-style-type: none"> <li>If <i>delimiter</i> is semicolon:<br/>x; x; x</li> <li>If <i>delimiter</i> is { ,   OR }, you can do either of the following: <ul style="list-style-type: none"> <li>x, x, x</li> <li>x OR x OR x</li> </ul> </li> </ul> 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
-

## Starting the Utilities

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

| Interface             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Database Window (DBW) | <p>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.</p> <p><b>Note:</b></p> <p>Operators must be members of the tdtrusted user group to run console utilities, or must be logged in as root. Non-tdtrusted 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 <i>Teradata Vantage™ - Database Utilities</i>, B035-1102.</p> <p>For low bandwidth connections, command-line interfaces to CNS are available, such as cnstern and cnstool. Online documentation is available for cnstern and cnstool in the form of Linux man pages.</p> <p>A subset of the console utilities can be run from the Remote Console portlet of Teradata Viewpoint. For more information, see <i>Teradata® Viewpoint User Guide</i>, B035-2206.</p> |
| 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:

```
tatcmd
```

at the Linux command line.

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

```
xdbw -display displayspec &
```

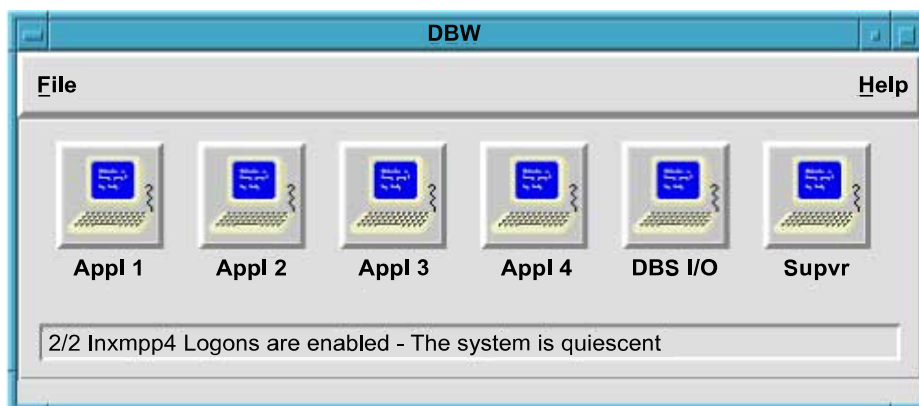
where *displayspec* is the name or IP address of the local machine, followed by a colon and the server number, typically 0 or 0.0. For example:

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

or

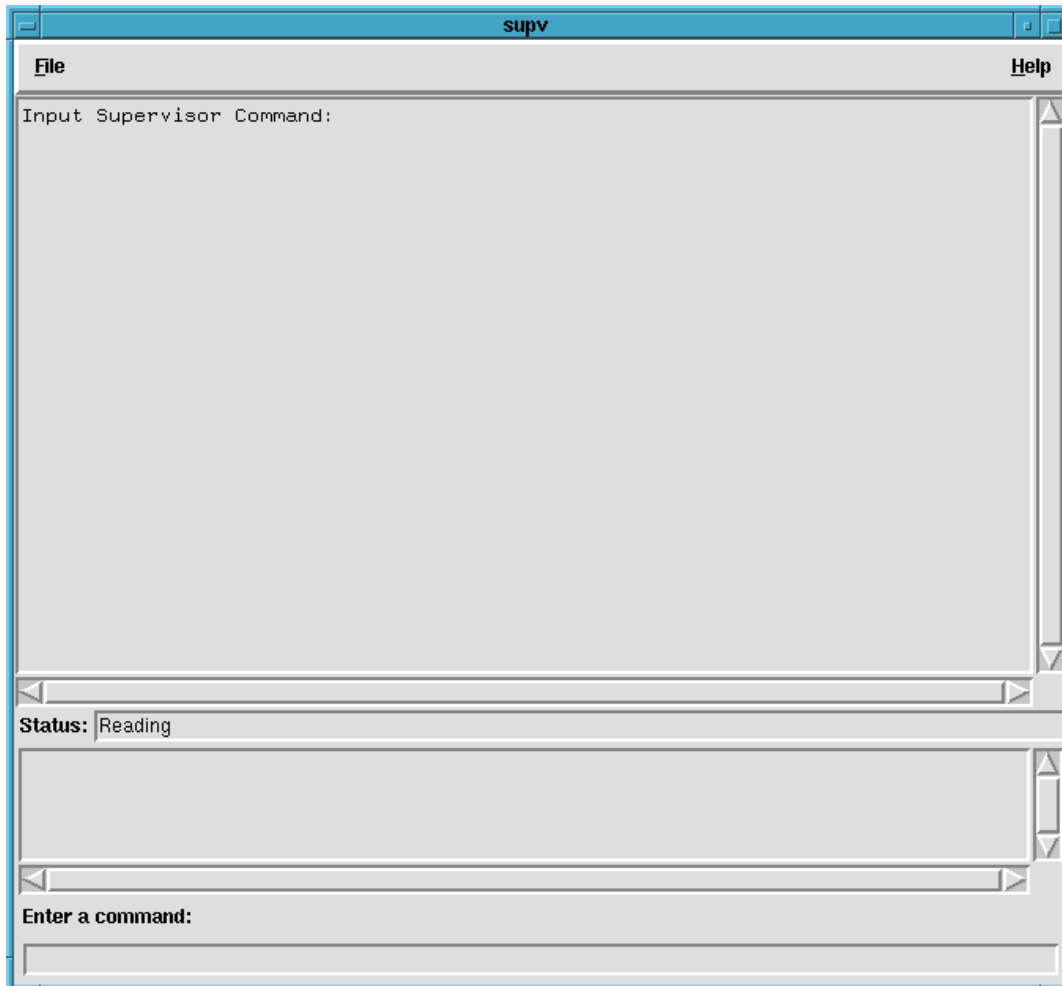
```
xdbw -display 192.0.2.24:0.0 &
```

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 `cnstern` 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'
```

# Session States, Events, and Actions

This section contains descriptions of the valid states, events, and actions shown in the output of the Gateway Global (gtwglobal) utility DISPLAY SESSION command.

## Session State Definitions

The following table shows the valid states that can occur for a session.

| State                      | Represents                                                                                                  |
|----------------------------|-------------------------------------------------------------------------------------------------------------|
| CS_ASNRSP                  | an assign response received from the assign task.                                                           |
| CS_ASNRSPSSO               | an assign response received from the assign task with saved SSO.                                            |
| CS_ASNRSPWAIT              | waiting for an assign response from the assign task.                                                        |
| CS_ASNRSPWAITERR           | waiting for an assign response from the assign task after an error occurred.                                |
| CS_ASNRSPWAITERRCLOSED     | waiting for an assign response from the assign task with the socket closed.                                 |
| CS_ASNRSPWAITSSO           | waiting for an assign response from the assign task with saved SSO.                                         |
| CS_CLIENTWAITDUNMSG        | waiting for a DUN start message from the client.                                                            |
| CS_CLIENTWAITELICITDATARSP | waiting for a response from the client after deferred transfer of lob data.                                 |
| CS_CLIENTWAITINDOUBT       | waiting for a request from the client after an in doubt transaction.                                        |
| CS_CLIENTWAITINTRAN        | waiting for a request from the client after a transaction.                                                  |
| CS_CLIENTWAITINTTRANSECPOL | a recursive state that handles security policy violation while waiting for client request in a transaction. |
| CS_CLIENTWAITNOTRAN        | waiting for a request from the client after no transaction.                                                 |
| CS_CLOSEWAIT               | waiting for the client to close the virtual circuit.                                                        |
| CS_CLOSEWAITTERM           | waiting for the client to close the virtual circuit and notify assign to free context.                      |

| State                         | Represents                                                                                          |
|-------------------------------|-----------------------------------------------------------------------------------------------------|
| CS_CLOSEWAITTERMDISC          | waiting for the client to close the virtual circuit and notify assign to keep context.              |
| CS_CLOSEWAITTERMGONE          | waiting for the client to close the virtual circuit and notify assign to force the context.         |
| CS_CONSSOWAIT                 | waiting for an SSO connect request from the client.                                                 |
| CS_CONTMSGGRSPINDOUBTLOGOFFOK | a recursive state that handles a continue message response after logoff in an in doubt transaction. |
| CS_CONTMSGGRSPINTRANLOGOFFOK  | a recursive state that handles a continue message response after logoff in a transaction.           |
| CS_CONTMSGGRSPNOTRANLOGOFFOK  | a recursive state that handles a continue message response after logoff not in a transaction.       |
| CS_CONWAIT                    | waiting for a connect request from the client.                                                      |
| CS_DIRECTMSGGRSPBAD           | a recursive state that handles a bad direct response from DBS.                                      |
| CS_DIRECTMSGGRSPOK            | a recursive state that handles a correct direct response from DBS.                                  |
| CS_DUNRSP                     | waiting for message from client or database.                                                        |
| CS_DUNSQLMSG                  | a recursive state that handles a DUN SQL message received from DBS.                                 |
| CS_ELICITDATAMSGREQ           | a recursive state that handles elicit data request from DBS.                                        |
| CS_ERRWAITFORCLOSE            | waiting for the client to close the connection after being sent an error message.                   |
| CS_ERRWAITFORLOGOFF           | waiting for the logoff to complete after the client closed the connection due to an error.          |
| CS_ESTABLISHED                | an idle logged on session.                                                                          |
| CS_ESTABLISHEDELICITSTARTRSP  | an idle logged on session that uses recoverable protocol.                                           |
| CS_ESTSECPOLVIOL              | a recursive state that handles a security policy violation on a logged on session.                  |
| CS_IDLE                       | a new session waiting for its first message from the Client.                                        |
| CS_LOGOFFRSPERR               | a recursive state that handles an error response for a logoff from the database.                    |

| State                         | Represents                                                                                                                                                                 |
|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CS_LOGOFFRSPERRINDOUBT        | a recursive state that handles an error response for a logoff from the database when it was in an in doubt transaction.                                                    |
| CS_LOGOFFRSPERRINTRAN         | a recursive state that handles an error response for a logoff from the database when it was in a transaction.                                                              |
| CS_LOGOFFRSPERRNOTRAN         | a recursive state that handles an error response from the database for a logoff when it was not in a transaction.                                                          |
| CS_LOGOFFRSPOK                | a recursive state that handles a logoff response from the database.                                                                                                        |
| CS_LOGOFFRSPOKINDOUBT         | a recursive state that handles a logoff response from the database when it was in an in doubt transaction.                                                                 |
| CS_LOGOFFRSPOKINTRAN          | a recursive state that handles a logoff response from the database when it was in a transaction.                                                                           |
| CS_LOGOFFRSPOKNOTRAN          | a recursive state that handles a logoff response from the database when it was not in a transaction.                                                                       |
| CS_LOGOFFRSPWAITUCABTRSP      | waiting for a UC abort response after sending a UC abort request during logoff processing.                                                                                 |
| CS_LOGOFFWAITRSP              | waiting for a logoff response from session control.                                                                                                                        |
| CS_LOGOFFWAITRSPCLOSECONTIMER | waiting for a logoff response from session control and the virtual circuit was timed out.                                                                                  |
| CS_LOGOFFWAITRSPCLOSED        | waiting for a logoff response from session control and the virtual circuit was closed.                                                                                     |
| CS_LOGOFFWAITRSPCLOSEDDISC    | waiting for a logoff response from session control and the virtual circuit was disconnected.                                                                               |
| CS_LOGOFFWAITRSPERR           | waiting for a logoff response from session control.                                                                                                                        |
| CS_LOGOFFWAITRSPFORCED        | waiting for a logoff response from session control and the client was killed.                                                                                              |
| CS_LOGOFFWAITRSPINDOUBT       | waiting for a logoff response from session control after the client sent a logoff request when it was in an in doubt transaction.                                          |
| CS_LOGOFFWAITRSPINDOUBTDISC   | waiting for a logoff response from session control after the client sent a logoff request when it was in an in doubt transaction and the virtual circuit was disconnected. |

| State                         | Represents                                                                                                                                                           |
|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CS_LOGOFFWAITRSPINDOUBTFORCED | waiting for a logoff response from session control after the client sent a logoff request when it was in an in doubt transaction and the session was killed.         |
| CS_LOGOFFWAITRSPINDOUBTTTERM  | waiting for a logoff response from session control after the client sent a logoff request when it was in an in doubt transaction and the virtual circuit was closed. |
| CS_LOGOFFWAITRSPINTRAN        | waiting for a logoff response from session control after the client sent a logoff request when it was in a transaction.                                              |
| CS_LOGOFFWAITRSPINTRANDISC    | waiting for a logoff response from session control after the client sent a logoff request when it was in a transaction and the virtual circuit was disconnected.     |
| CS_LOGOFFWAITRSPINTRANFORCED  | waiting for a logoff response from session control after the client sent a logoff request when it was in a transaction and the session was killed.                   |
| CS_LOGOFFWAITRSPINTRANSECPOL  | a recursive state that handles a security policy violation while waiting for the response to a logoff request when it was in a transaction.                          |
| CS_LOGOFFWAITRSPINTRANTERM    | waiting for a logoff response from session control after the client sent a logoff request when it was in a transaction and the virtual circuit was closed.           |
| CS_LOGOFFWAITRSPNOTRAN        | waiting for a logoff response from session control after the client sent a logoff request when it was not in a transaction.                                          |
| CS_LOGOFFWAITRSPNOTRANDISC    | waiting for a logoff response from session control after the client sent a logoff request when it was not in a transaction and the virtual circuit was disconnected. |
| CS_LOGOFFWAITRSPNOTRANFORCED  | waiting for a logoff response from session control after the client sent a logoff request when it was not in a transaction and the session was killed.               |
| CS_LOGOFFWAITRSPNOTRANTERM    | waiting for a logoff response from session control after the client sent a logoff request when it was not in a transaction and the virtual circuit was closed.       |
| CS_LOGOFFWAITRSPSECPOL        | a recursive state that handles a security policy violation while waiting for the response to a logoff request.                                                       |
| CS_LOGONFAILRSP               | a recursive state that handles a response from session control to a logon fail request.                                                                              |

| State                    | Represents                                                                                                       |
|--------------------------|------------------------------------------------------------------------------------------------------------------|
| CS_LOGONFAILWAIT         | waiting for a response from session control to a logon fail request.                                             |
| CS_LOGONFAILWAITCLOSED   | waiting for a response from session control to a logon fail request after the client has gone away.              |
| CS_LOGONFAILWAITCONTIMER | waiting for a response from session control to a logon fail request after the client has timed out.              |
| CS_LOGONFAILWAITERR      | waiting for a response from session control to a logon fail request after an error has occurred.                 |
| CS_LOGONRSPBAD           | a recursive state that handles bad Logon response.                                                               |
| CS_LOGONRSPOK            | a recursive state that handles correct Logon response.                                                           |
| CS_LOGONWAIT             | waiting for a response from session control to a logon request.                                                  |
| CS_LOGONWAITCLOSED       | waiting for a response from session control to a logon request after the client has gone away.                   |
| CS_LOGONWAITCLOSEDFORCED | waiting for a response from session control to a logon request after the client has gone away and was killed.    |
| CS_LOGONWAITERR          | waiting for a response from session control to a logon request after an error has occurred.                      |
| CS_OFFWAITUCABTRSP       | waiting for a UCAbort request after a successful logoff request.                                                 |
| CS_OFFWAITUCABTRSPFORCED | waiting for a UCAbort request after a successful logoff request where the client was killed.                     |
| CS_OFFWAITUCABTRSPSECPOL | a recursive state that handles security policy violation while waiting for the response to UCAbort after logoff. |
| CS_OFFWAITUCABTRSPTERM   | waiting for a UCAbort request after a successful logoff request where the client closed the virtual circuit.     |
| CS_REASNRSP              | reassign response received from the assign task.                                                                 |
| CS_REASNRSPSSO           | reassign response received from the assign task with saved SSO parcel.                                           |
| CS_REASNRSPWAIT          | waiting for a reassign response from the assign task.                                                            |
| CS_REASNRSPWAITSSO       | waiting for a reassign response from the assign task with saved SSO parcel.                                      |

| State                                  | Represents                                                                                                                                        |
|----------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| CS_REASNWAITCLOSED                     | waiting for a reassign response from the assign task after an error has occurred.                                                                 |
| CS_REASNWAITCLOSEDDISC                 | waiting for a reassign response from the assign task after an error has occurred and the client disconnected.                                     |
| CS_REASNWAITCLOSEDKILL                 | waiting for a reassign response from the assign task after receiving a GONE/KILL message.                                                         |
| CS_REASNWAITERR                        | waiting for a reassign response from the assign task after an error has occurred.                                                                 |
| CS_RECONFALLOFFWAITRSP                 | waiting for a logoff response from session control after reconnect failed.                                                                        |
| CS_RECONSSOWAIT                        | waiting for a reconnect response from the client using SSO.                                                                                       |
| CS_RECONWAIT                           | waiting for a reconnect response from the client.                                                                                                 |
| CS_RESSOWAIT                           | waiting for an SSO response from the client as part of reconnect.                                                                                 |
| CS_SECPOLVIOLLOFFWAITRSP               | waiting for a logoff response after a Security Policy Violation.                                                                                  |
| CS_SECPOLVIOLLOFFWAITRSPINTRAN         | waiting for a logoff response or client connection close after a Security Policy Violation. The logoff request was received inside a transaction. |
| CS_SECPOLVIOLLOFFWAITRSPINTRANTERM     | waiting for a logoff response after a Security Policy Violation. The logoff request was received inside a transaction.                            |
| CS_SECPOLVIOLLOFFWAITRSPWAITTERM       | waiting for the client to close the connection after a Security Policy Violation during logoff processing.                                        |
| CS_SECPOLVIOLOFFWAITUCABTRSP           | waiting for a UC abort response or client connection close after a Security Policy Violation during the UC abort processing as part of a logoff.  |
| CS_SECPOLVIOLOFFWAITUCABTRSPWAITABTRSP | waiting for a UC abort response after a Security Policy Violation during the UC abort processing as part of a logoff.                             |
| CS_SECPOLVIOLWAITABORTRSP              | waiting for abort response after Security Policy Violation.                                                                                       |
| CS_SECPOLVIOLWAITABORTRSPTERM          | waiting for abort response gateway has got terminate or disconnect from client after Security Policy Violation.                                   |



| State                                  | Represents                                                                                                                                            |
|----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| CS_SECPOLVIOLWAITCONTRSPINTRAN         | waiting for a continue response or client connection close after a Security Policy Violation. The continue request was received inside a transaction. |
| CS_SECPOLVIOLWAITCONTRSPINTRANWAITTERM | waiting for a continue response after a Security Policy Violation. The continue request was received inside a transaction.                            |
| CS_SECPOLVIOLWAITCONTRSPNOTRAN         | waiting for a continue response or client connection close after a Security Policy Violation.                                                         |
| CS_SECPOLVIOLWAITCONTRSPNOTRANWAITTERM | waiting for a continue response after a Security Policy Violation.                                                                                    |
| CS_SECPOLVIOLWAITDIRECT                | waiting for DBS response from a Directed request after Security Policy Violation.                                                                     |
| CS_SECPOLVIOLWAITDIRECTLOGOFF          | waiting for DBS response from a Directed request after Security Policy Violation.                                                                     |
| CS_SECPOLVIOLWAITSTARTRSPLOGOFF        | waiting for a start response or client connection close after a Security Policy Violation during logoff processing.                                   |
| CS_SECPOLVIOLWAITSTARTRSPLOGOFFTERM    | waiting for a start response after a Security Policy Violation during logoff processing.                                                              |
| CS_SECPOLVIOLWAITUCABTRSP              | waiting for Ucabort response after Security Policy Violation.                                                                                         |
| CS_SECPOLVIOLWAITUCABTRSPTERM          | waiting for Ucabort response gateway has got terminate or disconnect from client after Security Policy Violation.                                     |
| CS_SSOWAIT                             | waiting for an SSO response from the client.                                                                                                          |
| CS_STARTMSGGRSPINDOUBT                 | a recursive state that handles a start message response in an in doubt transaction.                                                                   |
| CS_STARTMSGGRSPINTRAN                  | a recursive state that handles a start message response in a transaction.                                                                             |
| CS_STARTMSGGRSPNOTRAN                  | a recursive state that handles a start message response when not in a transaction.                                                                    |
| CS_WAITABORTRSP                        | waiting for a DBS response to an ABORT request.                                                                                                       |
| CS_WAITCONTINUEACKINDOUBT              | waiting for an HSCMSGRECEIVED message as acknowledgement was received for a CONTINUE message that is in an in doubt transaction.                      |

| State                       | Represents                                                                                                                      |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| CS_WAITCONTINUEACKINTRAN    | waiting for an HSCMSGRECEIVED message as acknowledgement was received for a CONTINUE message that is in a transaction.          |
| CS_WAITCONTINUEACKNOTRAN    | waiting for an HSCMSGRECEIVED message as acknowledgement was received for a CONTINUE message that is not in a transaction.      |
| CS_WAITCONTINUERSPINDOUBT   | waiting for a DBS response from the Continue Request in an in doubt transaction.                                                |
| CS_WAITCONTINUERSPINTRAN    | waiting for a DBS response from the Continue Request in a transaction.                                                          |
| CS_WAITCONTINUERSPINNOTRAN  | waiting for a DBS response from the Continue Request not in a transaction.                                                      |
| CS_WAITCONTRSPINDOUBTDISC   | waiting for a DBS response from the Continue Request in an in doubt transaction after disconnect.                               |
| CS_WAITCONTRSPINDOUBTFORCED | waiting for a DBS response from the Continue Request in an in doubt transaction after kill.                                     |
| CS_WAITCONTRSPINDOUBTLOGOFF | waiting for a DBS response from the Continue Request in an in doubt transaction after logoff.                                   |
| CS_WAITCONTRSPINDOUBTTTERM  | waiting for a DBS response from the Continue Request in an in doubt transaction after terminate.                                |
| CS_WAITCONTRSPINTRANDISC    | waiting for a DBS response from the Continue Request in a transaction after disconnect.                                         |
| CS_WAITCONTRSPINTRANFORCED  | waiting for a DBS response from the Continue Request in a transaction after kill.                                               |
| CS_WAITCONTRSPINTRANLOGOFF  | waiting for a DBS response from the Continue Request in a transaction after logoff.                                             |
| CS_WAITCONTRSPINTRANSECPOL  | a recursive state that handles security policy violation while waiting for the response to a continue request in a transaction. |
| CS_WAITCONTRSPINTRANTERM    | waiting for a DBS response from the Continue Request in a transaction after terminate.                                          |
| CS_WAITCONTRSPNOTRANDISC    | waiting for a DBS response from the Continue Request not in a transaction after disconnect.                                     |
| CS_WAITCONTRSPNOTRANFORCED  | waiting for a DBS response from the Continue Request not in a transaction after kill.                                           |
| CS_WAITCONTRSPNOTRANLOGOFF  | waiting for a DBS response from the Continue Request not in a transaction after logoff.                                         |

| State                        | Represents                                                                                                       |
|------------------------------|------------------------------------------------------------------------------------------------------------------|
| CS_WAITCONTRSPNOTRANSECPOL   | a recursive state that handles a security policy violation while waiting for the response to a continue request. |
| CS_WAITCONTRSPNOTRANTERM     | waiting for a DBS response from the Continue Request not in a transaction after terminate.                       |
| CS_WAITDIRECTACK             | waiting for an HSCMSGRECEIVED message as acknowledgement that the DIRECT message was received.                   |
| CS_WAITDIRECTRSP             | waiting for a DBS response from a Directed Request.                                                              |
| CS_WAITDIRECTRSPDISC         | waiting for a DBS response from a Directed Request after a disconnect.                                           |
| CS_WAITDIRECTRSPLOGOFF       | waiting for a DBS response from a Directed Request after logoff.                                                 |
| CS_WAITDIRECTRSPLOGOFFFORCED | waiting for a DBS response from a Directed Request after logoff due to a kill.                                   |
| CS_WAITDIRECTRSPLOGOFFTERM   | waiting for a DBS response from a Directed Request after logoff due to a terminate.                              |
| CS_WAITDIRECTRSPSECPOL       | a recursive state that handles security policy violation while waiting for the response to a directed request.   |
| CS_WAITDUNABORTRSP           | waiting for DBS response to an ABORT request for DUN transaction.                                                |
| CS_WAITDUNRSP                | waiting for DUN response from database.                                                                          |
| CS_WAITFORTAKEOVERRSP        | waiting for TakeoverRsp from assign task.                                                                        |
| CS_WAITFORTAKEOVERRSPCLOSED  | waiting for TakeoverRsp from assign task after close.                                                            |
| CS_WAITSTARTACK              | waiting for an HSCMSGRECEIVED message as acknowledgement that the START message was received.                    |
| CS_WAITSTARTDBSRSPDISC       | waiting for a DBS response after a disconnect.                                                                   |
| CS_WAITSTARTDBSRSPFORCED     | waiting for a DBS response after a kill.                                                                         |
| CS_WAITSTARTDBSRSPTERM       | waiting for a DBS response after a terminate.                                                                    |
| CS_WAITSTARTRSP              | waiting for a DBS response from a Start Request.                                                                 |
| CS_WAITSTARTRSPLOGOFF        | waiting for a start response after receiving a logoff request.                                                   |

| State                         | Represents                                                                                                   |
|-------------------------------|--------------------------------------------------------------------------------------------------------------|
| CS_WAITSTARTRSPLOGOFFSECPOL   | a recursive state that handles security policy violation while waiting for the response to an abort request. |
| CS_WAITSTARTSECPOL            | a recursive state that handles security policy violation while waiting for start response.                   |
| CS_WAITSTRTDBSRSPFORCEDSECPOL | a recursive state that handles security policy violation while waiting for the response to an abort request. |
| CS_WAITUCABTRSP               | waiting for a DBS response to a UCAbort Request.                                                             |
| CS_WAITUCABTRSPDISC           | waiting for a DBS response to a UCAbort Request after a disconnect.                                          |
| CS_WAITUCABTRSPFORCED         | waiting for a DBS response to a UCAbort Request after a kill.                                                |
| CS_WAITUCABTRSPSECPOL         | a recursive state that handles security policy violation while waiting for the response to UCAbort.          |
| CS_WAITUCABTRSPTERM           | waiting for a DBS response to a UCAbort Request after a terminate.                                           |

## Session Event Definitions

Two types of session events occur:

- Fundamental external events
- Events that occur only if the given Action occurs in the State

### Fundamental External Events

The following table shows the fundamental external events that can occur for a session.

| Valid Events   | Event Type | Definition                                         |
|----------------|------------|----------------------------------------------------|
| CE_ABORTMSG    | Network    | Abort Msg was received.                            |
| CE_ASNMSG      | Network    | [Re]Assign Msg was received.                       |
| CE_ASSIGNRSP   | Gateway    | [Re]Assign response received from the assign task. |
| CE_CONFIGMSG   | Network    | Config message was received.                       |
| CE_CONNECTMSG  | Network    | Connect message was received.                      |
| CE_CONTIMEOUT  | Gateway    | Connect timer is timed out.                        |
| CE_CONTINUEMSG | Network    | Continue message was received.                     |

| Valid Events          | Event Type | Definition                                              |
|-----------------------|------------|---------------------------------------------------------|
| CE_COPMSGRECEIVED     | Network    | Message received message was received from client.      |
| CE_DBSRSP             | Database   | DBS Response was received.                              |
| CE_DIRECTMSG          | Network    | Directed Message was received.                          |
| CE_DIRECTMSGRSP       | Database   | DBS response to a directed request was received.        |
| CE_DISCONNECT         | Network    | Error occurred on virtual circuit.                      |
| CE_DUNCONTINUEMSG     | Network    | DUN continue message from client.                       |
| CE_DUNDBSRSP          | Database   | DUN message response from the database.                 |
| CE_DUNSTARTMSG        | Network    | DUN start message from client.                          |
| CE_ELICITDATAMSGRSP   | Network    | Elicitdata Message was received.                        |
| CE_ELICITSTARTRSP     | Network    | ElicitStartResponse Message was received.               |
| CE_ENCRYPTERR         | Gateway    | Unable to encrypt a message.                            |
| CE_GONE               | Database   | PMPC abort session message was received.                |
| CE_HDRERR             | Network    | A Client message header error occurred.                 |
| CE_HSCMSGRECEIVED     | Database   | Message received message was received from DBS.         |
| CE_KILL               | Gateway    | Gtwwglobal kill command occurred.                       |
| CE_LOGOFFMSG          | Network    | Logoff Request was received.                            |
| CE_LOGONLOGOFFRSP     | Database   | DBS response to a logon/logoff request received.        |
| CE_METHODMSG          | Network    | Methods Msg was received.                               |
| CE_RESUMEACTIONS      | Gateway    | Restore the current position in a state transition.     |
| CE_SAVEACTIONS        | Gateway    | Save the current position in a state transition.        |
| CE_SECPOLICYVIOLMSG   | Network    | A security policy violation message was received.       |
| CE_SECURITYPOLICYVIOL | Network    | A message failed to follow security policy.             |
| CE_SENDDISCONNECT     | Network    | A network send failed after the socket disconnected.    |
| CE_SENDTERMINATE      | Network    | A network send failed after the session was terminated. |
| CE_SSOMSG             | Network    | Single Sign-on Message was received.                    |
| CE_STARTMSG           | Network    | Start message has been received.                        |
| CE_TAKEOVER           | Gateway    | Received takeover request from assign task.             |
| CE_TAKEOVERRSP        | Gateway    | Received takeover response from assign task.            |
| CE_TERMINATE          | Network    | Orderly release on virtual circuit.                     |

| Valid Events | Event Type | Definition                 |
|--------------|------------|----------------------------|
| CE_TESTMSG   | Network    | Test message was received. |

### Events Occurring if the Action Occurs in the State

The following table shows the events that can occur for a session if the given action occurs in the state.

| Valid Events         | Event Type | Action         | Definition                                                |
|----------------------|------------|----------------|-----------------------------------------------------------|
| CE_ABORTMSGAUTH      | Recursive  | CA_ABORTMSG    | Abort Message had invalid security.                       |
| CE_ABORTMSGAUTH      | Recursive  | CA_DUNABORTMSG | DUN Abort Message had invalid security.                   |
| CE_ABORTMSGERR       | Recursive  | CA_ABORTMSG    | Abort Message contained an error.                         |
| CE_ABORTMSGERR       | Recursive  | CA_DUNABORTMSG | DUN Abort Message contained an error.                     |
| CE_ABORTMSGOK        | Recursive  | CA_ABORTMSG    | Abort Message was processed and was valid.                |
| CE_ABORTMSGOK        | Recursive  | CA_DUNABORTMSG | DUN Abort Message was processed and was valid.            |
| CE_ABORTMSGSKIP      | Recursive  | CA_ABORTMSG    | Abort Message was not for current request.                |
| CE_ABORTMSGSKIP      | Recursive  | CA_DUNABORTMSG | DUN Abort Message was not for current request.            |
| CE_ABORTMSGUCOK      | Recursive  | CA_ABORTMSG    | Abort Message contained a pclucabort.                     |
| CE_ABORTMSGUCOK      | Recursive  | CA_DUNABORTMSG | DUN Abort Message contained a pclucabort.                 |
| CE_ASNMSGERR         | Recursive  | CA_ASNMSG      | [Re]Assign message contained an error.                    |
| CE_ASNMSGOK          | Recursive  | CA_ASNMSG      | Assign Msg was processed and was valid.                   |
| CE_ASNMSGSSOOK       | Recursive  | CA_ASNMSG      | Assign Msg with SSO was processed and was valid.          |
| CE_ASNRSPERR         | Recursive  | CA_ASSIGNRSP   | Error [Re]Assign response received from the assign task.  |
| CE_ASNRSPOK          | Recursive  | CA_ASSIGNRSP   | Successful Assign response received from the assign task. |
| CE_ASNRSPSSOCONTINUE | Recursive  | CA_ASNRSPSSO   | AsnRspOk and SSO parcel needs continue.                   |

| Valid Events             | Event Type | Action               | Definition                                                      |
|--------------------------|------------|----------------------|-----------------------------------------------------------------|
| CE_ASNRSPSSOERR          | Recursive  | CA_ASNRSPSSO         | AsnRspOk and SSO parcel is in error.                            |
| CE_ASNRSPSSOFAIL         | Recursive  | CA_ASNRSPSSO         | AsnRspOk and SSO parcel failed authentication.                  |
| CE_ASNRSPSSOOK           | Recursive  | CA_ASNRSPSSO         | AsnRspOk and SSO parcel is Complete.                            |
| CE_BADSTATE              | Recursive  | CA_ASSIGNRSP         | Badstate Assign response received from the assign task.         |
| CE_CONNECTMSGAUTH        | Recursive  | CA_CONNECTMSG        | Connect message had invalid security.                           |
| CE_CONNECTMSGERR         | Recursive  | CA_CONNECTMSG        | Connect message contained an error.                             |
| CE_CONNECTMSGOK          | Recursive  | CA_CONNECTMSG        | Connect message was processed and was valid.                    |
| CE_CONTINUEMSGACKREQ     | Recursive  | CA_CONTINUEMSG       | Continue message with ack requested was processed and is valid. |
| CE_CONTINUEMSGAUTH       | Recursive  | CA_CONTINUEMSG       | Continue message had invalid security.                          |
| CE_CONTINUEMSGERR        | Recursive  | CA_CONTINUEMSG       | Continue message contained an error.                            |
| CE_CONTINUEMSGOK         | Recursive  | CA_CONTINUEMSG       | Continue message was processed and was valid.                   |
| CE_CONTINUEMSGGRSPERR    | Recursive  | CA_CONTINUEMSGGRSP   | Continue response from the DBS contained an error.              |
| CE_CONTINUEMSGGRSPOK     | Recursive  | CA_CONTINUEMSGGRSP   | Continue response from the DBS was OK.                          |
| CE_COPMSGRECEIVEDMSGAUTH | Recursive  | CA_COPMSGRECEIVEDMSG | COPMsgReceived message had invalid security.                    |
| CE_COPMSGRECEIVEDMSGERR  | Recursive  | CA_COPMSGRECEIVEDMSG | COPMsgReceived message contained an error.                      |
| CE_COPMSGRECEIVEDMSGOK   | Recursive  | CA_COPMSGRECEIVEDMSG | COPMsgReceived message is OK.                                   |
| CE_DIRECTMSGACKREQ       | Recursive  | CA_DIRECTMSG         | Direct message with ack request was processed and is valid.     |
| CE_DIRECTMSGAUTH         | Recursive  | CA_DIRECTMSG         | Direct message had invalid security.                            |
| CE_DIRECTMSGERR          | Recursive  | CA_DIRECTMSG         | Direct message contained an error.                              |

| Valid Events            | Event Type | Action              | Definition                                                             |
|-------------------------|------------|---------------------|------------------------------------------------------------------------|
| CE_DIRECTMSGOK          | Recursive  | CA_DIRECTMSG        | Direct message was processed and was valid.                            |
| CE_DIRECTMSGRSPERR      | Recursive  | CA_DIRECTMSGRSP     | Direct Message response contained an error.                            |
| CE_DIRECTMSGRSPOK       | Recursive  | CA_DIRECTMSGRSP     | Direct message response was OK.                                        |
| CE_DUNABORTMSGOK        | Recursive  | CA_DUNABORTMSG      | DUN Abort Message if the abort is for one transaction in DUN protocol. |
| CE_DUNCONTINUEMSGAUTH   | Recursive  | CA_DUNCONTINUEMSG   | DUN continue message has invalid security.                             |
| CE_DUNCONTINUEMSGERR    | Recursive  | CA_DUNCONTINUEMSG   | DUN continue message contained an error.                               |
| CE_DUNCONTINUEMSGOK     | Recursive  | CA_DUNCONTINUEMSG   | DUN continue message was OK.                                           |
| CE_DUNMSGRSP            | Recursive  | CA_DUNDBSRSP        | DUN message response from database.                                    |
| CE_DUNMSGRSPERR         | Recursive  | CA_DUNDBSRSP        | DUN message from database has an error.                                |
| CE_DUNSQLMSG            | Recursive  | CA_DUNDBSRSP        | DUN SQL message from database.                                         |
| CE_DUNSTARTMSGAUTH      | Recursive  | CA_DUNSTARTMSG      | DUN start message has invalid security.                                |
| CE_DUNSTARTMSGERR       | Recursive  | CA_DUNSTARTMSG      | DUN start message contained an error.                                  |
| CE_DUNSTARTMSGOK        | Recursive  | CA_DUNSTARTMSG      | DUN start message is OK.                                               |
| CE_ELICITDATAMSGACKREQ  | Recursive  | CA_ELICITDATAMSGRSP | Elicit data message with ack request was processed and is valid.       |
| CE_ELICITDATAMSGREQ     | Recursive  | CA_STARTMSGRSP      | An Elicit Data request was received instead of a Start response.       |
| CE_ELICITDATAMSGRSPAUTH | Recursive  | CA_ELICITDATAMSGRSP | Elicit data message response had invalid security.                     |
| CE_ELICITDATAMSGRSPERR  | Recursive  | CA_ELICITDATAMSGRSP | Elicit data message response contained an error.                       |
| CE_ELICITDATAMSGRSPOK   | Recursive  | CA_ELICITDATAMSGRSP | Elicit data message response was OK.                                   |
| CE_ELICITSTARTRSPAUTH   | Recursive  | CA_ELICITSTARTRSP   | ElicitStartResponse message had invalid security.                      |



| Valid Events         | Event Type | Action            | Definition                                                       |
|----------------------|------------|-------------------|------------------------------------------------------------------|
| CE_ELICITSTARTRSPERR | Recursive  | CA_ELICITSTARTRSP | ElicitStartResponse message contained an error.                  |
| CE_ELICITSTARTRSPOK  | Recursive  | CA_ELICITSTARTRSP | ElicitStartResponse message is OK.                               |
| CE_LOGOFFMSGRSPERR   | Recursive  | CA_LOGOFFMSGRSP   | An error response received from session control.                 |
| CE_LOGOFFMSGRSPOK    | Recursive  | CA_LOGOFFMSGRSP   | A successful response received from session control.             |
| CE_LOGONERR          | Recursive  | CA_SENDLOGON      | An error in the logon message.                                   |
| CE_LOGONFAIL         | Recursive  | CA_SENDLOGON      | An error in building logon message.                              |
| CE_LOGONFAILERR      | Recursive  | CA_SENDLOGONFAIL  | An error in trying to send a failed logon message.               |
| CE_LOGONMSGRSPERR    | Recursive  | CA_LOGONMSGRSP    | An error response received from session control.                 |
| CE_LOGONMSGRSPOK     | Recursive  | CA_LOGONMSGRSP    | A successful response received from session control.             |
| CE_LOGONMSGRSPRUN    | Recursive  | CA_LOGONMSGRSP    | A bad runrsp was received from session control.                  |
| CE_LOGONSSOERR       | Recursive  | CA_SENDLOGONSSO   | An error in the SSO logon message.                               |
| CE_METHODMSGAUTH     | Recursive  | CA_METHODMSG      | Method Msg had invalid security.                                 |
| CE_METHODMSGERR      | Recursive  | CA_METHODMSG      | Method Msg contained an error.                                   |
| CE_METHODMSGOK       | Recursive  | CA_METHODMSG      | Method Msg was valid.                                            |
| CE_NODEFULL          | Recursive  | CA_ASSIGNRSP      | Node full assign response was received from assign.              |
| CE_REASNMSGOK        | Recursive  | CA_ASNMSG         | ReAssign Message was processed and was valid.                    |
| CE_REASNMSGSSOOK     | Recursive  | CA_ASNMSG         | ReAssign Message with SSO was processed and was valid.           |
| CE_REASNRSPOK        | Recursive  | CA_ASSIGNRSP      | Successful ReAssign response received from Assign task.          |
| CE_REASNRSPOKSSO     | Recursive  | CA_ASSIGNRSP      | Successful ReAssign response with SSO received from Assign task. |

| Valid Events           | Event Type | Action                 | Definition                                                       |
|------------------------|------------|------------------------|------------------------------------------------------------------|
| CE_REASNRSPSSOCONTINUE | Recursive  | CA_REASNRSPSSO         | AsnRspOk and SSO parcel needs continue.                          |
| CE_REASNRSPSSOERR      | Recursive  | CA_REASNRSPSSO         | AsnRspOk and SSO parcel is in error.                             |
| CE_REASNRSPSSOFAIL     | Recursive  | CA_REASNRSPSSO         | AsnRspOk and SSO parcel failed authentication.                   |
| CE_REASNRSPSSOOK       | Recursive  | CA_REASNRSPSSO         | AsnRspOk and SSO parcel is Complete.                             |
| CE_RECONERR            | Recursive  | CA_RECON               | Reconnect validation unsuccessful.                               |
| CE_RECONNECTMSGOK      | Recursive  | CA_CONNECTMSG          | ReConnect message was processed and was valid.                   |
| CE_RECONOK             | Recursive  | CA_RECON               | Reconnect validation successful.                                 |
| CE_RECONSSOERR         | Recursive  | CA_RECONSSO            | Reconnect validation unsuccessful.                               |
| CE_RECONSSOOK          | Recursive  | CA_RECONSSO            | Reconnect validation successful.                                 |
| CE_RECONSSOTAKEOVERERR | Recursive  | CA_RECONSSO            | Reconnect validation unsuccessful for a taking over session.     |
| CE_RECONSSOTAKEOVEROK  | Recursive  | CA_RECONSSO            | Takeover needs to be sent to takeover already connected session. |
| CE_RECOVERABLELEVEL0   | Recursive  | CA_GETRECOVERABLELEVEL | Client does not support RecoverableProtocol.                     |
| CE_RECOVERABLELEVEL1   | Recursive  | CA_GETRECOVERABLELEVEL | Client supports RecoverableProtocol level 1.                     |
| CE_RESSOCONTINUE       | Recursive  | CA_SENDRSSORSP         | The Reconnect SSO authentication required more exchanges.        |
| CE_RESSOERR            | Recursive  | CA_SENDRSSORSP         | The Reconnect SSO authentication was not successful.             |
| CE_SSOCONTINUE         | Recursive  | CA_SENDSSORSP          | The SSO authentication required more exchanges.                  |
| CE_SSOERR              | Recursive  | CA_SENDSSORSP          | The SSO authentication was not successful.                       |
| CE_SSOFAIL             | Recursive  | CA_SENDSSORSP          | The SSO authentication failed.                                   |
| CE_SSOMSGAUTH          | Recursive  | CA_SSOMSG              | An SSO message had invalid security.                             |

| Valid Events          | Event Type | Action           | Definition                                                              |
|-----------------------|------------|------------------|-------------------------------------------------------------------------|
| CE_SSOMSGERR          | Recursive  | CA_SSOMSG        | An error was encountered processing the SSO message.                    |
| CE_SSOMSGOK           | Recursive  | CA_SSOMSG        | An SSO message has been received and is OK.                             |
| CE_STARTMSGACKREQ     | Recursive  | CA_STARTMSG      | A Start message with ack request was processed and is valid.            |
| CE_STARTMSGAUTH       | Recursive  | CA_STARTMSG      | A Start message had invalid security.                                   |
| CE_STARTMSGERR        | Recursive  | CA_STARTMSG      | An error was encountered processing the start message.                  |
| CE_STARTMSGOK         | Recursive  | CA_STARTMSG      | A start message has been received and is OK.                            |
| CE_STARTMSGRSPERR     | Recursive  | CA_STARTMSGRSP   | A start request response is in error.                                   |
| CE_STARTMSGRSPINDOUBT | Recursive  | CA_STARTMSGRSP   | A start request response for an in doubt transaction has been received. |
| CE_STARTMSGRSPINTRAN  | Recursive  | CA_STARTMSGRSP   | A start request response in a transaction has been received.            |
| CE_STARTMSGRSPNOTRAN  | Recursive  | CA_STARTMSGRSP   | A start request response not in a transaction has been received.        |
| CE_UCABORTMSGRSPERR   | Recursive  | CA_UCABORTMSGRSP | UCAbort Response was not received.                                      |
| CE_UCABORTMSGRSPOK    | Recursive  | CA_UCABORTMSGRSP | UCAbort Response was received.                                          |

## Session Action Definitions

The following table shows the valid actions and definitions that can occur for a session.

### Note:

Actions of the form CA\_ERRGTWxxxx will cause the CA\_SENDERR\_inline function to be called with the value specified by ERRGTWxxxx.

| Valid Action | Definition                              |
|--------------|-----------------------------------------|
| CA_ABORTMSG  | Validate abort request from the client. |

| Valid Action           | Definition                                                      |
|------------------------|-----------------------------------------------------------------|
| CA_ASNMSG              | Validate assign request from the client.                        |
| CA_ASNRSP              | Process valid assign response from assign task.                 |
| CA_ASNRSPSSO           | Process valid assign response from assign task with SSO parcel. |
| CA_ASSIGNRSP           | Process [re]assign response from assign task.                   |
| CA_CLEANDUNINFO        | Clean the DUN info structure.                                   |
| CA_CLOSE               | Close the file description in connect task.                     |
| CA_CONFIGMSG           | Process a config message.                                       |
| CA_CONNECTMSG          | Validate connect request from the client.                       |
| CA_CONTINUEMSG         | Validate continue request from the client.                      |
| CA_CONTINUEMSGRSP      | Process continue response from the DBS.                         |
| CA_COPMSGRECEIVEDMSG   | Validate COPMsgRecieved message from the client.                |
| CA_DIRECTMSG           | Validate directed request from the client.                      |
| CA_DIRECTMSGRSP        | Process the response to a directed message.                     |
| CA_DUNABORTMSG         | Validate the abort request from the client in DUN protocol.     |
| CA_DUNCONTINUEMSG      | Process the continue message from client in DUN protocol.       |
| CA_DUNDBSRSP           | Process the response from DBS in DUN protocol.                  |
| CA_DUNSTARTMSG         | Process the start message from client in DUN protocol.          |
| CA_ELICITDATAMSGRSP    | Process the response from the client to an elicit data request. |
| CA_ELICITSTARTRSP      | Process the elicitstartresponse from the client.                |
| CA_FORWARDEVENT        | Forwards the event to new state.                                |
| CA_FREE                | Free the session context.                                       |
| CA_GETRECOVERABLELEVEL | Generate an event based on RecoverableProtocol level.           |
| CA_ILLEGALEVENT        | An unexpected event occurred.                                   |
| CA_LOG                 | Log an event.                                                   |
| CA_LOGAUTHERR          | Log an authentication error.                                    |
| CA_LOGOFF              | Build and send a logoff message to session control.             |
| CA_LOGOFFMSGRSP        | Process logoff response from the DBS.                           |
| CA_LOGONMSGRSP         | Process logon response from the DBS.                            |
| CA_METHODMSG           | Validate method request from client.                            |

| Valid Action          | Definition                                                        |
|-----------------------|-------------------------------------------------------------------|
| CA_NOP                | Placeholder - do nothing.                                         |
| CA_REASNRSP           | Process valid reassign response from assign task.                 |
| CA_REASNRSPSSO        | Process valid reassign response from assign task with SSO parcel. |
| CA_RECON              | Process reconnect message from client.                            |
| CA_RECONSSO           | Process reconnect message from client using SSO.                  |
| CA_SECPOLICYVIOLMSG   | Process security policy violation message.                        |
| CA_SENDABORT          | Send an abort request to the DBS.                                 |
| CA_SENDASN            | Build and send assign request to assign task.                     |
| CA_SENDASNRSP         | Build and send assign response to the client.                     |
| CA_SENDASNRSPSSO      | Build and send assign response and SSO parcel to the client.      |
| CA_SENDCONFIGRSP      | Send the config response to the client.                           |
| CA_SENDCONRSP         | Send the connect response to the client.                          |
| CA_SENDCONTINUE       | Send the continue request to the DBS.                             |
| CA_SENDCOPMSGRECEIVED | Send a message received message to the client.                    |
| CA_SENDDBSRSP         | Send the DBS response to the client.                              |
| CA_SENDDIRECT         | Send the directed request to the DBS.                             |
| CA_SENDDIRECTRSP      | Send the Direct Request response to the client.                   |
| CA_SENDDUNABORT       | Send the abort request to database in DUN protocol.               |
| CA_SENDDUNCONTINUE    | Send the DUN continue message to database.                        |
| CA_SENDDUNMSGRSP      | Send the DUN message response from database to client.            |
| CA_SENDDUNSQLMSG      | Send the DUN SQL message to client.                               |
| CA_SENDDUNSTART       | Send the DUN start message to database.                           |
| CA_SENDELICITDATA     | Send the Elicit Data to the DBS.                                  |
| CA_SENDELICITDATAREQ  | Send the ElicitData Request to the client.                        |
| CA_SENDELICITSTARTRSP | Send ElicitStartResponse to the DBS.                              |
| CA_SENDERR            | Build and send error parcel to the client.                        |
| CA_SENDERR_inline     | Build and send error parcel from statetable to the client.        |
| CA_SENDHSCMSGRECEIVED | Send a message received message to the DBS.                       |
| CA_SENDLOGOFFOKRSP    | Build OK logoff response and forward to the client.               |

| Valid Action       | Definition                                                      |
|--------------------|-----------------------------------------------------------------|
| CA_SENDLOGOFFRSP   | Forward logoff response to the client.                          |
| CA_SENDLOGON       | Send the normal logon request to session control.               |
| CA_SENDLOGONFAIL   | Send a logon failed message to session control.                 |
| CA_SENDLOGONSSO    | Build and send SSO logon request to session control.            |
| CA_SENDMETHODRSP   | Build and send methods response to the client.                  |
| CA_SENDPERR        | Forward error parcel from session control to the client.        |
| CA_SENDREASN       | Build and send reassign request to assign task.                 |
| CA_SENDREASNRSP    | Build and send reassign response to the client.                 |
| CA_SENDREASNRSPSSO | Build and send reassign response with SSO parcel to the client. |
| CA_SENDRECONRSP    | Build and send reconnect response to the client.                |
| CA_SENDRESSORSP    | Build and send reconnect SSO response to the client.            |
| CA_SENDRNPUCABORT  | Send a UC abort request to the DBS for RNP session.             |
| CA_SENDSSORSP      | Build and send SSO response to the client.                      |
| CA_SENDSTART       | Forward start request to the DBS.                               |
| CA_SENDTAKEOVER    | Send Takeover message to assign task.                           |
| CA_SENDTERM        | Build and send terminate request to assign task.                |
| CA_SENDTERMDISC    | Build and send a disconnect terminate request to assign task.   |
| CA_SENDTERMFORCE   | Build and send a force terminate request to assign task.        |
| CA_SENDTERMFREE    | Build and send a free terminate request to assign task.         |
| CA_SENDTERMGONE    | Build and send a gone terminate request to assign task.         |
| CA_SENDTERMTIMEOUT | Build and send a timeout terminate request to assign task.      |
| CA_SENDOCABORT     | Send a UC abort request to the DBS.                             |
| CA_SHUTDOWN        | Shutdown the virtual circuit.                                   |
| CA_SSOMSG          | Validate SSOMSG.                                                |
| CA_STARTCONTIMER   | Start connect timer.                                            |
| CA_STARTMSG        | Validate Start message.                                         |
| CA_STARTMSGRSP     | Process Start Response from DBS.                                |
| CA_STOPCONTIMER    | Stop connect timer.                                             |
| CA_TESTMSG         | Process the Test Message from the client.                       |

| Valid Action     | Definition                                |
|------------------|-------------------------------------------|
| CA_UCABORTMSGRSP | Validate if DBS response is to a UCABORT. |

# PDE Tools

## PDE Tools

Parallel Database Extensions (PDE) is a software interface layer that lies between the operating system and Advanced SQL Engine. PDE supports the parallelism across system nodes that contributes to Advanced SQL Engine speed and linear scalability. Many diagnostic and troubleshooting utilities work at the PDE level.

PDE tools are a collection of PDE-level utilities that come with Advanced SQL Engine. They are not documented in Utilities because PDE tools have online documentation accessible from a system console using the "man" and "pdehelp" commands.

This section presents a list of some of the useful PDE tools you can use. These tools and more are located in the /usr/pde/bin directory.

### Note:

Not all of the PDE tools are suitable for customer use without appropriate guidance from the Teradata Support Center. Consult with Teradata Support before running PDE tools that are not listed here.

| PDE Tool Name | Description                                                                                                                       |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------|
| amplload      | Displays the AMP vproc usage of AMP Worker Tasks (AWTs) on Advanced SQL Engine, representing the load on each AMP vproc           |
| awtmon        | Collects and displays a user-friendly summary of the AMP Worker Task (AWT) in-use count snapshots for the local node or all nodes |
| cnstern       | PDE Console Subsystem Terminal                                                                                                    |
| cnstool       | PDE Console Subsystem Tool                                                                                                        |
| controlgdo    | Displays Control GDO contents                                                                                                     |
| csp           | Crashdump Save Program                                                                                                            |
| cspclean      | Clean up Crashdump Save Program's run time environment                                                                            |
| cspinfo       | Crashdump Save Program Information display utility                                                                                |
| csppeek       | Display useful information from a PDE dump                                                                                        |
| cspstop       | Graceful shut down of the Crashdump Save Program                                                                                  |
| cspwake       | Awaken the Crashdump Save Program                                                                                                 |
| dmpbust       | Decode PDE dump files                                                                                                             |
| evtool        | Displays the event message text for any event message provided by EMF                                                             |



| PDE Tool Name  | Description                                                                                                                                                                                                                                                                                           |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| gdomview       | Displays entries from the GDO changes log, to show configuration setting changes that were made to dbscontrol and ctl globally distributed objects within a given time range. Changes to these GDOs are made using the DBS Control (dbscontrol) and Control GDO Editor (ctl) utilities, respectively. |
| logview        | System Log and Dump Utility                                                                                                                                                                                                                                                                           |
| mboxchk        | A front-end script for puma -m command monitors Advanced SQL Engine mailbox messages to detect DBS hangs                                                                                                                                                                                              |
| modmpplist     | Modifies the node list (mpplist) file on local or all system nodes                                                                                                                                                                                                                                    |
| nodecheck      | Gathers and analyzes PDE vital information from local node                                                                                                                                                                                                                                            |
| pcl            | Parallel Command Language user interface                                                                                                                                                                                                                                                              |
| pdeconf        | PDE configuration settings                                                                                                                                                                                                                                                                            |
| pdegloba       | Displays the PDE global information                                                                                                                                                                                                                                                                   |
| pdehelp        | Displays Teradata tools online help                                                                                                                                                                                                                                                                   |
| pdeinfo        | Saves system-wide PDE related data                                                                                                                                                                                                                                                                    |
| pdedrv         | PDE device driver load/unload administration command                                                                                                                                                                                                                                                  |
| pdepath        | Displays paths of Teradata running environment                                                                                                                                                                                                                                                        |
| pdestat        | PDE System call Statistics Display Utility                                                                                                                                                                                                                                                            |
| pdestate       | Displays the current PDE state and substate                                                                                                                                                                                                                                                           |
| pdestop        | Brings PDE down immediately on the local node                                                                                                                                                                                                                                                         |
| syscheck       | Runs nodecheck on all system nodes to gather and analyze PDE vital information                                                                                                                                                                                                                        |
| tdatcmd        | Sets up a Teradata command shell or window                                                                                                                                                                                                                                                            |
| tdinfo         | Displays the Teradata system configuration                                                                                                                                                                                                                                                            |
| tdinstall      | Install a Teradata component software package                                                                                                                                                                                                                                                         |
| tdmaint        | Tool for marking the node for maintenance and change it back                                                                                                                                                                                                                                          |
| tdnstat        | Teradata Network (TDN) Statistics Display Utility displays or clears network statistics                                                                                                                                                                                                               |
| tdntune        | Teradata Network (TDN) dynamic tunables read/write utility                                                                                                                                                                                                                                            |
| tdpkgrm        | Removes old Teradata software versions                                                                                                                                                                                                                                                                |
| toscrashcount  | TPA Crash Count Utility                                                                                                                                                                                                                                                                               |
| tosgetpma      | Displays the contents of the tosgetpma structure                                                                                                                                                                                                                                                      |
| tpacontrolnode | Display PDE control node id, cabinet, and module                                                                                                                                                                                                                                                      |
| tpanode        | Display the PDE node and the node type.                                                                                                                                                                                                                                                               |

| PDE Tool Name | Description                                                                            |
|---------------|----------------------------------------------------------------------------------------|
| tpasetcfg     | Cancels PDE's NETCONFIG start-up timeout                                               |
| tpasumtrace   | Summarizes TPA start-up and restart traces for all nodes                               |
| tpatrace      | Displays PDE start-up and restart traces for various subsystems                        |
| tpatracedump  | Extracts and displays TPA traces from a kernel panic dump                              |
| tsklist       | Displays information about PDE processes and their tasks                               |
| prginv        | PDE program invocation utility                                                         |
| prgreset      | PDE Partition Reset Utility                                                            |
| prgtrace      | Produces detailed traces of actions taken by PDE programs                              |
| psh           | Parallel SHell (P-SHell)                                                               |
| puma          | Prints PDE information and kernel memory addresses                                     |
| v2pgdo        | Displays the contents of the V2P GDO                                                   |
| vconfig       | Translates configuration definition to VCONFIG GDO                                     |
| verify_pdisks | verifies configuration definition of vconfig.gdo and matches the configuration on disk |
| vprocgdo      | Displays the contents of the VprocConfig GDO                                           |

# Additional Information

## Teradata Links

| Link                                                                                                    | Description                                                                                                                                                                                                                                                |
|---------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="https://docs.teradata.com/">https://docs.teradata.com/</a>                                     | Search Teradata Documentation, customize content to your needs, and download PDFs.<br>Customers: Log in to access Orange Books.                                                                                                                            |
| <a href="https://support.teradata.com">https://support.teradata.com</a>                                 | One-stop source for Teradata community support, software downloads, and product information.<br>Log in for customer access to: <ul style="list-style-type: none"><li>• Community support</li><li>• Software updates</li><li>• Knowledge articles</li></ul> |
| <a href="https://www.teradata.com/University/Overview">https://www.teradata.com/University/Overview</a> | Teradata education network                                                                                                                                                                                                                                 |
| <a href="https://support.teradata.com/community">https://support.teradata.com/community</a>             | Link to Teradata community                                                                                                                                                                                                                                 |