

Teradata Vantage™ - SystemFE Macros

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.

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

Chapter 1: Introduction to SystemFE Macros	5
Prerequisites	5
Changes and Additions	5
Chapter 2: SystemFE Macros Description	6
Determining Contents of the DBC.SW_Event_Log	6
Restoring the SystemFE Database from an Earlier Release	6
AllRestarts	7
BynetEvents	9
CDS_ViewExclusions/CDS_ViewExclusions_ANSI	12
CDS_AddExclusion/CDS_AddExclusion_ANSI	14
CDS_RemoveExclusion/CDS_RemoveExclusion_ANSI	15
CDS_ViewPercentData	15
CDS_ModifyPercentData	16
CDS_Object/CDS_Object_ANSI	17
CDS_Database/CDS_Database_ANSI	17
CDS_System/CDS_System_ANSI	19
CDS_PhysicalTableSizeEstimate/CDS_PhysicalTableSizeEstimate_ANSI	20
CDS_LogicalTableSizeEstimate/CDS_LogicalTableSizeEstimate_ANSI	21
ReconfigCheck	22
DiskEvents	23
EventCount	29
FallBack_All	31
FallBack_DB	35
ListErrorCodes	38
ListEvent	40
ListRestart_Logon_Events	42
ListSoftware_Event_Log	44
MiniCylPacks	48
NoFallBack_All	50
NoFallBack_DB	51
PackDisk	53
Chapter 3: Using SystemFE Macros	56
Database Preventive Maintenance	56
Analyzing Preventive Maintenance Reports	57
Other Resources for Analyzing System Messages	58
Determining Fallback Protection	59
Reconfiguring AMPs	59

Updating Vantage Software 59

Appendix A: How to Read Syntax 60

Appendix B: SystemFE Macro Listings 62

Appendix C: Additional Information 73

Introduction to SystemFE Macros

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.

This document is a basic guide to a set of macros for generating reports on system events and using them as a diagnostic tool.

Prerequisites

You should be familiar with Teradata® SQL and Basic Teradata Query (BTEQ) facility.

If you are not familiar with BTEQ or Teradata SQL, you can review the following documents:

- *Basic Teradata® Query Reference*, B035-2414
- *Teradata Vantage™ - SQL Fundamentals*, B035-1141
- *Teradata Vantage™ - SQL Data Manipulation Language*, B035-1146

Changes and Additions

Date	Description
July 2021	With this release, Teradata session mode macros are created in DIPPOST and ANSI session mode macros are created in DIPANSI. See macro descriptions in SystemFE Macros Description .
June 2020	<p>Added the Customer Data Space (CDS) Tools, macros that return the data space used in terabytes by tables, databases, and the system. The macros also let you exclude certain tables from the calculations and view the list of excluded tables. The following macros are added:</p> <ul style="list-style-type: none"> • CDS_AddExclusion/CDS_AddExclusion_ANSI • CDS_Database/CDS_Database_ANSI • CDS_LogicalTableSizeEstimate/CDS_LogicalTableSizeEstimate_ANSI • CDS_ModifyPercentData/CDS_ModifyPercentData_ANSI • CDS_Object/CDS_Object_ANSI • CDS_PhysicalTableSizeEstimate/CDS_PhysicalTableSizeEstimate_ANSI • CDS_RemoveExclusion/CDS_RemoveExclusions_ANSI • CDS_System/CDS_System_ANSI • CDS_ViewExclusions/CDS_ViewExclusions_ANSI • CDS_ViewPercentData/CDS_ViewPercentData_ANSI

SystemFE Macros Description

The SystemFE macros provide information about reported system events. You can use this information to monitor the health of the database. The SystemFE Macros are designed to be run using the BTEQ facility.

By default, the SystemFE macros reside in the SystemFE database. All macros that have *Fromdate* and *Todate* as parameters use inclusive dates. The default value for *Todate* is DATE (the current date).

You can use SystemFE macros to quickly produce reports relevant to system messages stored in the DBC.SW_Event_Log.

Teradata recommends that you use character dates as opposed to numeric dates for producing reports.

Recommended format: 'YYYY-MM-DD'

For example, for May 31, 2013 input '2013-05-31'

Format not recommended: (YYYY-1900) *10000+ m*100+d

For example, for May 31, 2013, 1130531 is correct but not recommended.

For more information on using character format and numeric format, see *Teradata Vantage™ - Data Types and Literals*, B035-1143.

For additional resources to consult for any problem or proactive health check activity, as well as advice on when to delete system messages from the DBC.SW_Event_Log, see [Analyzing Preventive Maintenance Reports](#).

If you do not have the SystemFE macros installed on your system, open Database Window (DBW). Then, run the Database Initialization Program (DIP) utility from the Supervisor window and select the SQL script file DIPSYSFE. For Teradata session mode CDS macros, select SQL script file DIPPOST and for ANSI session mode CDS macros, select SQL script file DIPANSI.

For information on DBW and the DIP utility, see *Teradata Vantage™ - Database Utilities*, B035-1102.

Determining Contents of the DBC.SW_Event_Log

To determine if data exists in the DBC.SW_Event_Log, execute the following SQL statement:

```
Select Count(*), Min(TheDate), Max(TheDate) From DBC.Software_Event_Log;
```

This returns a single row containing a count of events, the date of the earliest event, and the date of the most recent event. The count will be zero and the dates will be NULL if there are no events in the log.

Restoring the SystemFE Database from an Earlier Release

To restore the SystemFE database from an earlier release, you must first clean up macros that are no longer meaningful to the current system. After cleaning up the obsolete macros, you need to recreate the SystemFE.diskevents macro.

DIP must have already been run once to ensure that the supporting infrastructure is updated (for example, empty macros or views should already have been created). If DIP has been run at least once previously, run the appropriate DIP again to recreate the macro.

DIP files that you might have to run are:

- DIPSYSFE - creates the macro if it is NOT already present
- DIPPOST - creates Teradata session mode CDS macros if they are NOT already present
- DIPANSI - creates ANSI session mode CDS macros if they are NOT already present
- DIPPATCH - fills in for any release-dependent changes

For more information on these DIP files, see *Teradata Vantage™ - Database Utilities*, B035-1102.

AllRestarts

Returns all software restart events logged in the DBC.SW_Event_Log table for a set period of time.

Syntax

```
AllRestarts ( Fromdate [, Todate] ) ;
```

Syntax Elements

Fromdate

The earliest date of a restart that is to be listed in the report.

The format is 'YYYY-MM-DD'. This parameter can be coded either as a character string, for example, '2013-05-31', or as a numeric value, for example, 1130531.

Todate

The latest date of a restart that is to be listed in the report. The format is 'YYYY-MM-DD'.

This parameter can be coded either as a character string, for example, '2013-05-31', or as a numeric value, for example, 1130531. The default value for *Todate* is the current date.

Note:

This macro is not case sensitive. Uppercase and lowercase letters can be used interchangeably.

Usage Notes

The AllRestarts macro report is based on event code 033-13895-00.

For a complete description of these events, see the information about Parallel Database Extensions (PDE) messages in *Teradata Vantage™ - Database Messages*, B035-1096.

The output shows the date, time, and all lines of the DBC.SW_Event_Log connected with each software restart event. This macro also provides a summary of the causes of outage at the end of the report to help identify events that cause the most restarts.

Examples

Example: Generating a Report of All Restarts Up to the Current Date

The following statement generates a report of all restarts between May 5, 2013, and the date of today:

```
Exec AllRestarts ('2013-05-05',);
```

Example: Generating a Report of All Restarts in a Week

The following statement generates a report of all restarts for the last seven days:

```
Exec AllRestarts (Date-6,);
```

Example: Generating a Report of All Restarts Between Specific Dates

The following statement generates a report of all restarts between January 1, 2013 and March 18, 2013:

```
exec AllRestarts('2013-01-01','2014-03-18');
```

The system displays a report similar to this:

```
exec AllRestarts('2013-01-01','2014-03-18');
*** Echo accepted.
*** Total elapsed time was 1 second.
*** Echo accepted.
*** Echo accepted.
*** Echo accepted.
*** Echo accepted.
```

```

*** Echo accepted.
*** Echo accepted.
*** Echo accepted.
*** Echo accepted.
*** Query completed. 4 rows found. 5 columns returned.
14/04/04                               (Crashable Events)
11:45
      Between 2013-01-01 and      2014-03-18
Date      Time Message Text
-----
-----
14/03/14 16:54:28.912135 PDE: TPA restart initiated by this node, 001-01, for
event 10198.
      16:54:55.723604 PDE: TPA restart initiated by this node, 001-01, for
event 10198.
14/03/17 14:39:29.013986 PDE: TPA restart initiated by this node, 001-01, for
event 10198.
      15:05:51.651466 PDE: TPA restart initiated by this node, 001-01, for
event 10198.
*** Echo accepted.
*** Echo accepted.
*** Query completed. One row found. One column returned.
      Total Restarts
      Restarts
      -----
      4
*** Echo accepted.

```

BynetEvents

Returns all BYNET events logged in the DBC.SW_Event_Log table for a set period of time.

Note:

This macro is not case-sensitive. Uppercase and lowercase letters can be used interchangeably.

Syntax

```
BYNETEvents ( Fromdate [, Todate] ) ;
```

Syntax Elements

Fromdate

The date on which the search is to start. The format is 'YYYY-MM-DD'. This parameter can be coded either as a character string, for example, '2013-05-31', or as a numeric value, for example, 1130531.

Todate

The date on which the search is to stop. The format is 'YYYY-MM-DD'. This parameter can be coded either as a character string, for example, '2013-05-31', or as a numeric value, for example, 1130531. The default value for *Todate* is the current date.

Note:

This macro is not case sensitive. Uppercase and lowercase letters can be used interchangeably.

Usage Notes

The BynetEvents macro returns all messages based on event codes:

- 033-11100-00 to 033-11121-00 (TDN messages)
- 033-12900-00 to 033-12999-00 (BNS messages)
- 033-13000-00 to 033-13021-00 (CFG messages)
- And, for any event from any series that contains the string "bynet"

For a complete description of these events, see “Parallel Database Extensions (PDE) Messages” *Teradata Vantage™ - Database Messages*, B035-1096.

If you want to examine more information than the macro displays by default, run a SELECT query on the DBC.SW_Event_Log table by providing the necessary criteria and targeting against the field “TEXT.” Here is an example of the SELECT query:

```
select * from dbc.Software_Event_Log
where Text like '%restart%'
order by thedate, thetime
;
```

This causes the system to display all the available information for the events rather than a truncated version. In most cases, the latter version is sufficient to understand the nature of the problem.

Informational messages (that is, those requiring no action) are recorded to the DBC.Software_Event_Log view when one of the following occurs:

- A TPA restart occurs
- The BYNET initiates a TPA restart

When multiple messages are recorded over a very short time period, a BYNET event might have occurred forcing a node offline. If a node has been forced offline by the BYNET, you might need a Teradata field engineer to resolve the BYNET event.

Examples

Example: Generating a Report of BYNET Events in a Week

The following statement generates a report of all BYNET Events for the last seven days:

```
Exec BynetEvents (Date-6,);
```

Example: Generating a Report of BYNET Events Between October and February

The following statement generates a report of BYNET Events between October 20, 2012 and February 23, 2013:

```
Exec BynetEvents ('2012-10-20', '2013-02-23');
```

Example: Generating a Report of BYNET Events Between December and January

The following statement generates a report of BYNET Events between December 19, 2012 and January 1, 2013:

```
Exec BynetEvents ('2012-12-19', '2013-01-01');
```

The resulting report follows:

```
Exec BynetEvents ('2012-12-19','2013-01-01');

*** Echo accepted.
*** Total elapsed time was 1 second.

*** Echo accepted.
*** Echo accepted.
*** Echo accepted.
*** Echo accepted.
*** Echo accepted.
*** Echo accepted.
```

```

*** Echo accepted.
*** Echo accepted.
*** Query completed. 9 rows found. 5 columns returned.
13/02/29      BYNET Events Between 12/12/19 and 13/01/01      16:19
Date          Time Message Text
-----
12/12/28  12:49:04.000000 140001006|bynet |1|S|I|0|1|1|B|0|0|bya.c|65535|65535|
65535|10#BYA: BYA113-4 restarting to bring in late
arriver node 0xDE4 on port 0.

14:52:48.000000 140000806|bynet|1|H|I|0|1|2|B|0|0|bya.c|65535|65535|
65535|11#BYA: BYA112-4 input port error on
adapter port 1: unexpected command.
140000806|bynet|1|H|I|0|1|2|B|0|0|bya.c|65535|65535|
65535|11#BYA: BYA113-4 input port error on adapter
port 0: soft carrier loss.
15:02:05.000000 140001006|bynet |1|S|I|0|1|2|B|0|0|bya.c|65535|65535|
65535|12#BYA: BYA113-4 restarting to bring in late
arriver node 0xDE4 on port 0.
140001006|bynet |1|S|I|0|1|3|B|0|0|bya.c|65535|65535|
65535|12#BYA: BYA112-4 restarting to bring in late
arriver node 0xDE4 on port 1.
12/12/29  19:09:48.000000 140000806|bynet |1|H|I|0|1|3|B|0|0|bya.c|65535|65535|
65535|13#BYA: BYA113-4 input port error on adapter
port 0: soft carrier loss.
140000806|bynet |1|H|I|0|1|2|B|0|0|bya.c|65535|65535|
65535|13#BYA: BYA112-4 input port error on adapter
port 1: violation error.
19:19:06.000000 140001006|bynet |1|S|I|0|1|1|B|0|0|bya.c|65535|65535|
65535|14#BYA: BYA112-4 restarting to bring in late
arriver node 0xDE4 on port 1.
140001006|bynet |1|S|I|0|1|3|B|0|0|bya.c|65535|65535|
65535|14#BYA: BYA113-4 restarting to bring in late
arriver node 0xDE4 on port 0.

*** Echo accepted.

```

CDS_ViewExclusions/CDS_ViewExclusions_ANSI

Returns the list of objects to be excluded from data space computations for a database or system.

Note:

If you use ANSI session mode, use the `_ANSI` form of the macro.

Teradata session mode macros are created in DIPPOST and ANSI session mode macros are created in DIPANSI.

Syntax

```
[ SystemFE. ] { CDS_ViewExclusions | CDS_ViewExclusions_ANSI } ;
```

Usage Notes

- By default, all Data Dictionary and system tables are added to the exclusion list.
To add other tables or databases to the exclusion list, use `CDS_AddExclusion` or `CDS_AddExclusion_ANSI`.
- DBC.CDSTableSizeV does not report objects in exclusion list.

Example

The following example returns the list of objects excluded from customer data space computations on a system. An asterisk (*) means all tables of the database.

```
EXEC systemfe.CDS_ViewExclusions;
```

DataBaseName	TableName
DBC	*
dbcmngr	*
LockLogShredder	*
PDCRADMIN	*
PDCRDATA	*
PDCRSTG	*
SQLJ	*
SysAdmin	*
SYSBAR	*
SYSJDBC	*
SYSLIB	*
SYSSPATIAL	*
SYSTEMFE	*
SYSUDTLIB	*
Sys_Calendar	*
TDMaps	*

TDQCD	*
TDStats	*
tdwm	*
TD_SERVER_DB	*
TD_SYSFNLIB	*
TD_SYSXML	*

CDS_AddExclusion/CDS_AddExclusion_ANSI

Adds objects to a list to exclude from the customer data space computation for a database or system.

Note:

If you use ANSI session mode, use the `_ANSI` form of the macro.

Teradata session mode macros are created in DIPPOST and ANSI session mode macros are created in DIPANSI.

Syntax

```
[ SystemFE. ] { CDS_AddExclusion | CDS_AddExclusion_ANSI }
( TargetDatabaseName, TargetTableName ) ;
```

Syntax Elements

TargetDatabaseName

Database name of the table to exclude from data space computation.

TargetTableName

Table name to exclude from data space computation.

To exclude all tables of *TargetDatabaseName*, use '*' as the *TargetTableName* value.

Examples

The following example adds Test.Clobtable to the list of tables to exclude from the data space computation:

```
EXEC systemfe.CDS_AddExclusion ('test', 'clobtable');
```

The following example adds all tables of database Test to the list of tables to exclude from the data space computation:

```
EXEC systemfe.CDS_AddExclusion ('test', '*');
```

CDS_RemoveExclusion/CDS_RemoveExclusion_ANSI

Removes an object from the exclusion list.

Note:

If you use ANSI session mode, use the `_ANSI` form of the macro.

Teradata session mode macros are created in DIPPOST and ANSI session mode macros are created in DIPANSI.

Syntax

```
[ SystemFE. ] { CDS_RemoveExclusion | CDS_RemoveExclusion_ANSI } (
    TargetDatabaseName,
    TargetTableName
) ;
```

Syntax Elements

TargetDatabaseName

Name of the database on which the target object resides.

TargetTableName

Name of the target object.

Examples

The following example removes the table `test1.table1` from the exclusion list:

```
EXEC systemfe.CDS_RemoveExclusion('test1', 'table1');
```

CDS_ViewPercentData

Returns the value that determines the percent of data for which the macros `CDS_Database`, `CDS_Database_ANSI`, `CDS_System`, and `CDS_System_ANSI` collect statistics when called with `CollectSummaryStats ('Y')`.

Note:

Teradata session mode macros are created in DIPPOST.

Syntax

```
[ SystemFE. ] CDS_ViewPercentData ;
```

Examples

The following example returns the percent value for macros CDS_Database, CDS_Database_ANSI, CDS_System, and CDS_System_ANSI to use when collecting statistics:

```
EXEC systemfe.CDS_ViewPercentData;
```

```
*** Query completed. One row found. One column returned.
*** Total elapsed time was 1 second.
```

```
PercentValue
-----
          80.00
```

CDS_ModifyPercentData

Modifies the value that determines the percent of data for which the macros CDS_Database, CDS_Database_ANSI, CDS_System, and CDS_System_ANSI collect statistics when called with CollectSummaryStats ('Y').

Note:

Teradata session mode macros are created in DIPPOST.

Syntax

```
[ SystemFE. ] CDS_ModifyPercentData ( NewPercentValue ) ;
```

Syntax Elements

NewPercentValue

Percent value for macros CDS_Database, CDS_Database_ANSI, CDS_System, and CDS_System_ANSI to use when collecting statistics.

Examples

The following example changes the percent value for macros CDS_Database, CDS_Database_ANSI, CDS_System, and CDS_System_ANSI to use when collecting statistics to 90:

```
EXEC systemfe.CDS_ModifyPercentData(90);
```

CDS_Object/CDS_Object_ANSI

Returns the consumed customer data space in terabytes for the specified object.

Note:

If you use ANSI session mode, use the _ANSI form of the macro.

Teradata session mode macros are created in DIPPOST and ANSI session mode macros are created in DIPANSI.

Syntax

```
[ SystemFE. ] { CDS_Object | CDS_Object_ANSI }  
( InputDatabaseName, InputTableName ) ;
```

Syntax Elements

InputDatabaseName

The name of the database in which the table resides.

InputTableName

The name of the table to evaluate for customer data space consumption.

Examples

The following example returns the consumed data space in terabytes for the table Glass:

```
EXEC systemfe.CDS_Object('Materials', 'Glass');
```

CDS_Database/CDS_Database_ANSI

Returns the used customer data space in terabytes for a database.

Note:

If you use ANSI session mode, use the `_ANSI` form of the macro.

Teradata session mode macros are created in DIPPOST and ANSI session mode macros are created in DIPANSI.

Syntax

```
[ SystemFE. ] { CDS_Database | CDS_Database_ANSI }
( InputDatabaseName, CollectSummaryStats ) ;
```

Syntax Elements

InputDatabaseName

The name of the database to evaluate for customer data space consumption.

CollectSummaryStats

Flag that indicates whether to collect summary statistics on the largest tables in *InputDatabaseName* to return a more accurate estimate of CDS size. Value must be either 'Y' (yes) or 'N' (no).

Usage Notes

- To run this macro with *CollectSummaryStats* ('Y'), the SystemFE user needs the SELECT and STATS privileges on *InputDatabaseName*.
SELECT lets the user check whether the statistics are stale and STATS lets the user collect statistics on objects.
- CollectSummaryStats* ('Y') increases the accuracy of the CDS values that the macro returns for the database. However, if the statistics are stale or not already collected for *InputDatabaseName*, the macro must collect new statistics, which increases macro running time.

Examples

The following example returns the consumed data space in terabytes for the inputdb database:

```
EXEC systemfe.CDS_Database('inputdb', 'N');
```

The following example collects summary statistics on the objects in the inputdb database if the statistics are stale or not already collected, to return a more accurate estimate of CDS size:

```
EXEC systemfe.CDS_Database('inputdb', 'Y');
```

CDS_System/CDS_System_ANSI

Returns the consumed customer data space in terabytes for the system.

Note:

If you use ANSI session mode, use the `_ANSI` form of the macro.

Teradata session mode macros are created in DIPPOST and ANSI session mode macros are created in DIPANSI.

Syntax

```
[ SystemFE. ] { CDS_System | CDS_System_ANSI } ( CollectSummaryStats ) ;
```

Syntax Elements

CollectSummaryStats

Flag that indicates whether to collect summary statistics on the largest tables on the system to return a more accurate estimate of CDS size. Value must be either 'Y' (yes) or 'N' (no).

Usage Notes

- To run this macro with `CollectSummaryStats ('Y')`, the SystemFE user needs the SELECT and STATS privileges on all databases in the system.
SELECT lets the user check whether the statistics are stale and STATS lets the user collect statistics on objects.
- `CollectSummaryStats ('Y')` increases the accuracy of the CDS values that the macro returns for the system. However, if the statistics are stale or not already collected for database objects, the macro must collect new statistics, which increases macro running time.

Example

The following example returns the consumed data space in terabytes for the system:

```
EXEC SystemFE.CDS_System('N');
```

Note:

Use CDS_AddExclusion/CDS_AddExclusion_ANSI and CDS_ViewExclusions/CDS_ViewExclusions_ANSI to add to and view a list of objects to be excluded from data space computations. By default, some Data Dictionary tables are already in the exclusion list.

The following example collects summary statistics on the objects in the system if they are stale or not already collected, to return a more accurate estimate of CDS size:

```
EXEC systemfe.CDS_System('Y');
```

CDS_PhysicalTableSizeEstimate/ CDS_PhysicalTableSizeEstimate_ANSI

Returns the estimated physical table size in terabytes without fallback for the specified logical object size.

Note:

If you use ANSI session mode, use the _ANSI form of the macro.

Teradata session mode macros are created in DIPPOST and ANSI session mode macros are created in DIPANSI.

Syntax

```
[ SystemFE. ] { CDS_PhysicalTableSizeEstimate |  
CDS_PhysicalTableSizeEstimate_ANSI } (  
    TargetDatabaseName,  
    TargetTableName,  
    LogicalSizeTB  
) ;
```

Syntax Elements

TargetDatabaseName

Name of the database on which the target object resides.

TargetTableName

Name of the target object.

LogicalSizeTB

Logical size of the target object in terabytes.

Examples

The following example returns the estimated physical size in terabytes without fallback for the table test1.table1 if its logical size is 5 terabytes:

```
EXEC systemfe.CDS_PhysicalTableSizeEstimate('test1', 'table1', 5);
```

CDS_LogicalTableSizeEstimate/ CDS_LogicalTableSizeEstimate_ANSI

Returns the estimated logical table size in terabytes for the specified physical object size.

Note:

If you use ANSI session mode, use the _ANSI form of the macro.

Teradata session mode macros are created in DIPPOST and ANSI session mode macros are created in DIPANSI.

Syntax

```
[ SystemFE. ] { CDS_LogicalTableSizeEstimate |  
CDS_LogicalTableSizeEstimate_ANSI } (  
    TargetDatabaseName,  
    TargetTableName,  
    PhysicalSizeTB  
) ;
```

Syntax Elements

TargetDatabaseName

Name of the database on which the target object resides.

TargetTableName

Name of the target object.

PhysicalSizeTB

Physical size of the target object in terabytes, excluding fallback.

Examples

The following example returns the estimated logical size in terabytes for the table test1.table1 if its physical size is 2 terabytes without fallback:

```
EXEC systemfe.CDS_LogicalTableSizeEstimate('test1', 'table1', 2);
```

ReconfigCheck

Returns an estimated number of bytes for each access module process (AMP) required for a reconfiguration operation.

Syntax

```
ReconfigCheck ( DestAMPs ) ;
```

Syntax Elements

DestAMPs

The number of AMPs to which you would like to calculate the amount of bytes required per AMP to complete the reconfiguration operation.

Note:

This macro is not case sensitive. Uppercase and lowercase letters can be used interchangeably.

Usage Notes

The ReconfigCheck macro is useful for obtaining an estimate of the space required prior to Teradata Support performing a reconfiguration operation. Because of background system test, multiple executions of ReconfigCheck might return different results.

Examples

Example: Generating a Report to Provide the Estimated Space for a One-AMP System

The following statement generates a report which provides the estimated space required to reconfigure a one-AMP system:

```
Exec ReconfigCheck (1);
```

Example: Generating a Statement of Estimated Space per AMP in a Four-AMP System

The following statement generates a report which provides the estimated space required per AMP to reconfigure a four-AMP system:

```
Exec ReconfigCheck (4);
```

The resulting report follows:

```
*** Delete completed. One row removed.
*** Total elapsed time was 1 second.
*** Insert completed. One row added.
*** Echo accepted.
*** Echo accepted.
*** Echo accepted.
*** Echo accepted.
*** Echo accepted.
*** Echo accepted.
*** Echo accepted.
*** Echo accepted.
*** Query completed. One row found. 2 columns returned.
13/05/21          Bytes Required For 4 AMP System          10:50
                        Bytes
                        Required
                        Per AMP
                        -----
                        1,021,118,464
*** Echo accepted.
```

DiskEvents

Returns all disk events logged in the DBC.SW_Event_Log table for a set period of time.

Syntax

```
DiskEvents ( Fromdate [, Todate] ) ;
```

Syntax Elements

Fromdate

The date on which the search is to start. The format is 'YYYY-MM-DD'. This parameter can be coded either as a character string, for example, '2013-05-31', or as a numeric value, for example, 1130531.

Todate

The date on which the search is to stop. The format is 'YYYY-MM-DD'. This parameter can be coded either as a character string, for example, '2013-05-31', or as a numeric value, for example, 1130531. The default value for *Todate* is the current date.

Note:

This macro is not case sensitive. Uppercase and lowercase letters can be used interchangeably.

Usage Notes

The DiskEvents macro report is based on event codes 033-11036-00, 033-11038-00, and 033-11094-00.

For a complete description of these events, see the information about Parallel Database Extensions (PDE) messages in *Teradata Vantage™ - Database Messages*, B035-1096.

You can also use a SELECT query to view events in the DBC.SW_Event_Log (see [“Determining Contents of the DBC.SW_Event_Log”](#)).

Examples

Example: Generating a Report of Disk Events for a Week

The following statement generates a report of all Disk Events for the last seven days:

```
Exec DiskEvents (Date-6,);
```

Example: Generating a Report of Disk Events Between September and January

The following statement generates a report of Disk Events between September 30, 2012 and January 31, 2013:

```
Exec DiskEvents ('2012-09-30' , '2013-01-31');
```

Example: Generating a Report of Disk Events Between August and February

The following statement generates a report of Disk Events between August 9, 2012 and February 20, 2013:

```
Exec DiskEvents ('2012-08-09','2013-02-20');
```

The resulting report follows:

```
exec systemfe.diskevents('2012-08-09','2013-02-20');
*** Echo accepted.
*** Total elapsed time was 2 seconds.
*** Query completed. 691 rows found. 5 columns returned.
13/02/19          Disk Events Between 2012-01-01 and 2013-02-20
                12:09
      Date          Time Message Text
-----
-----
13/02/02 18:41:16.381378 Event number 33-11094-00 (severity 10, category 9)
Received OS I/O
                        error.
                        Extent driver error TVSA_E_SYSTEM: OS Error -5
                        Cylinder ID = 0x0x0001000D00000026 (38:13:1:0)
                        State          = 0x03, Sector    = 0x0080, Length    =
0x00000200
                        Attr          = 0x00, Vproc ID = 0x000D, Error      =
0x00000010
                        Class        = 0x01, Flags     = 0x0000, OS Error =
0xFFFFFFFFB
                18:41:16.757614 Event number 33-11094-00 (severity 10, category 9)
Received OS I/O
                        error.
                        Extent driver error TVSA_E_SYSTEM: OS Error -5
                        Cylinder ID = 0x0x0001000900000026 (38:9:1:0)
                        State          = 0x03, Sector    = 0x0080, Length    =
0x00000200
                        Attr          = 0x00, Vproc ID = 0x0009, Error      =
0x00000010
                        Class        = 0x01, Flags     = 0x0000, OS Error =
0xFFFFFFFFB
                18:41:16.860288 Event number 33-11094-00 (severity 10, category 9)
```

Received OS I/O

```

error.
Extent driver error TVSA_E_SYSTEM: OS Error -5
Cylinder ID = 0x0x0001000B00000024 (36:11:1:0)
State      = 0x03, Sector   = 0x0099, Length   =
0x00000200
Attr       = 0x00, Vproc ID = 0x000B, Error    =
0x00000010
Class      = 0x01, Flags    = 0x0000, OS Error =
0xFFFFFFFFB
18:41:16.892315 Event number 33-11094-00 (severity 10, category 9)

```

Received OS I/O

```

error.
Extent driver error TVSA_E_SYSTEM: OS Error -5
Cylinder ID = 0x0x0001000A00000025 (37:10:1:0)
State      = 0x03, Sector   = 0x0083, Length   =
0x00000200
Attr       = 0x00, Vproc ID = 0x000A, Error    =
0x00000010
Class      = 0x01, Flags    = 0x0000, OS Error =
0xFFFFFFFFB
18:41:16.981832 Event number 33-11094-00 (severity 10, category 9)

```

Received OS I/O

```

error.
Extent driver error TVSA_E_SYSTEM: OS Error -5
Cylinder ID = 0x0x0001000C00000024 (36:12:1:0)
State      = 0x03, Sector   = 0x008A, Length   =
0x00000400
Attr       = 0x00, Vproc ID = 0x000C, Error    =
0x00000010
Class      = 0x01, Flags    = 0x0000, OS Error =
0xFFFFFFFFB
18:41:17.066148 Event number 33-11094-00 (severity 10, category 9)

```

Received OS I/O

```

error.
Extent driver error TVSA_E_SYSTEM: OS Error -5
Cylinder ID = 0x0x0001000800000025 (37:8:1:0)
State      = 0x03, Sector   = 0x0083, Length   =
0x00000200
Attr       = 0x00, Vproc ID = 0x0008, Error    =
0x00000010
Class      = 0x01, Flags    = 0x0000, OS Error =
0xFFFFFFFFB
18:41:17.422217 Event number 33-11094-00 (severity 10, category 9)

```

Received OS I/O

error.

Extent driver error TVSA_E_SYSTEM: OS Error -5

Cylinder ID = 0x0x0001000700000025 (37:7:1:0)

State = 0x03, Sector = 0x0095, Length =

0x00000200

13/02/19

Disk Events Between 2012-01-01 and 2013-02-20

12:09

Date

Time Message Text

13/02/02 18:41:17.422217 Attr = 0x00, Vproc ID = 0x0007, Error =
0x00000010

Class = 0x01, Flags = 0x0000, OS Error =

0xFFFFFFFFB

13/02/03 14:16:44.075227 Event number 33-11094-00 (severity 10, category 9)

Received OS I/O

error.

Extent driver error TVSA_E_SYSTEM: OS Error -5

Cylinder ID = 0x0x0001000A00000026 (38:10:1:0)

State = 0x03, Sector = 0x042A, Length =

0x00000200

Attr = 0x00, Vproc ID = 0x000A, Error =

0x00000010

Class = 0x01, Flags = 0x0000, OS Error =

0xFFFFFFFFB

14:16:44.155212 Event number 33-11094-00 (severity 10, category 9)

Received OS I/O

error.

Extent driver error TVSA_E_SYSTEM: OS Error -5

Cylinder ID = 0x0x0001000C00000026 (38:12:1:0)

State = 0x03, Sector = 0x0455, Length =

0x00000400

Attr = 0x00, Vproc ID = 0x000C, Error =

0x00000010

Class = 0x01, Flags = 0x0000, OS Error =

0xFFFFFFFFB

14:16:44.203279 Event number 33-11094-00 (severity 10, category 9)

Received OS I/O

error.

Extent driver error TVSA_E_SYSTEM: OS Error -5

Cylinder ID = 0x0x0001000D00000027 (39:13:1:0)

State = 0x03, Sector = 0x0461, Length =

0x00000200

```

Attr          = 0x00, Vproc ID = 0x000D, Error      =
0x00000010
Class         = 0x01, Flags      = 0x0000, OS Error =
0xFFFFFFFFB
14:16:44.207242 Event number 33-11094-00 (severity 10, category 9)
Received OS I/O
error.
Extent driver error TVSA_E_SYSTEM: OS Error -5
Cylinder ID = 0x0x0001000800000029 (41:8:1:0)
State        = 0x03, Sector     = 0x0442, Length    =
0x000000600
Attr          = 0x00, Vproc ID = 0x0008, Error      =
0x00000010
Class         = 0x01, Flags      = 0x0000, OS Error =
0xFFFFFFFFB
14:16:44.523432 Event number 33-11094-00 (severity 10, category 9)
Received OS I/O
error.
Extent driver error TVSA_E_SYSTEM: OS Error -5
Cylinder ID = 0x0x0001000900000025 (37:9:1:0)
State        = 0x03, Sector     = 0x0432, Length    =
0x000000400
Attr          = 0x00, Vproc ID = 0x0009, Error      =
0x00000010
Class         = 0x01, Flags      = 0x0000, OS Error =
0xFFFFFFFFB
14:16:44.655585 Event number 33-11094-00 (severity 10, category 9)
Received OS I/O
error.
Extent driver error TVSA_E_SYSTEM: OS Error -5
Cylinder ID = 0x0x0001000700000027 (39:7:1:0)
State        = 0x03, Sector     = 0x0427, Length    =
0x000000400
Attr          = 0x00, Vproc ID = 0x0007, Error      =
0x00000010
Class         = 0x01, Flags      = 0x0000, OS Error =
0xFFFFFFFFB
14:16:44.875817 Event number 33-11094-00 (severity 10, category 9)
Received OS I/O
error.
Extent driver error TVSA_E_SYSTEM: OS Error -5
.

```

•
•

EventCount

Returns a count of each type of event in DBC.SW_Event_Log which occurred in the set period.

Syntax

```
EventCount ( Fromdate [, Todate ] ) ;
```

Syntax Elements

Fromdate

The date on which the search is to start. The format is 'YYYY-MM-DD'. This parameter can be coded either as a character string, for example, '2013-05-31', or as a numeric value, for example, 1130531.

Todate

The date on which the search is to stop. The format is 'YYYY-MM-DD'. This parameter can be coded either as a character string, for example, '2013-05-31', or as a numeric value, for example, 1130531. The default value for *Todate* is the current date.

Note:

This macro is not case sensitive. Uppercase and lowercase letters can be used interchangeably.

Usage Notes

The EventCount macro takes into account event code 033-13895-00, or whose severity is greater than 20.

For a complete description of these events, see the information about Parallel Database Extensions (PDE) messages in *Teradata Vantage™ - Database Messages*, B035-1096.

This macro does not report counts for events, which are informational in nature, but rather only those events that will give you an idea of the health of Vantage.

From this macro, you can observe event code counts and plan maintenance accordingly.

Examples

Example: Generating a Statement of Events Between May 5 and the Current Date

The following statement provides counts of events between May 5, 2013, and the date of today:

```
Exec EventCount ('2013-05-05',);
```

Example: Generating a Statement of Event Codes Between October and February

The following statement provides counts of event codes between October 28, 2012 and February 12, 2013:

```
Exec EventCount ('2012-10-28','2013-02-12');
```

Example: Generating a Statement of Events in the Past 60 Days

The following statement provides counts of events for the last sixty days:

```
Exec EventCount (Date-60,);
```

The resulting report follows:

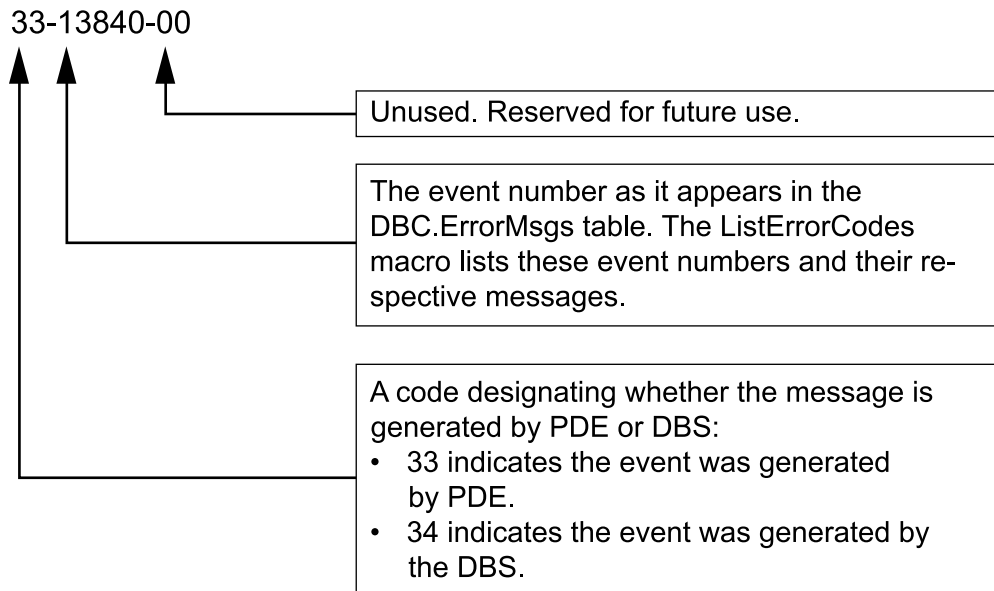
```
EXECUTE EventCount (Date-60,);
*** Echo accepted.
*** Total elapsed time was 1 second.
*** Echo accepted.
*** Echo accepted.
*** Echo accepted.
*** Echo accepted.
*** Echo accepted.
*** Echo accepted.
*** Echo accepted.
*** Query completed. One row found. 4 columns returned.
14/04/04                      Total Of Each Event                      10:44
                             Between 2014-02-03 and 2014-04-04
                                   Total
                             Event Tag  Occurrences
```

```
*** Echo accepted.
```

```
-----
33-10198-00
```

```
5
```

The following figure shows how to interpret the Event Tags included in the report.



FallBack_All

Returns all tables in the system which have fallback protection.

Syntax

```
FallBack_All ;
```

Note:

This macro is not case sensitive. Uppercase and lowercase letters can be used interchangeably.

Usage Notes

The FallBack_All macro reports the names of the database and tables, and the size of the table in bytes.

This macro is useful for obtaining all the tables on the system which have fallback copies.

For a discussion of fallback, see *Teradata Vantage™ - Database Design*, B035-1094.

Example: Generating a Statement of Tables with the Fallback Option

The following statement generates a report of all tables with the fallback option:

```
Exec FallBack_All;
```

The resulting report follows:

```
BTEQ -- Enter your DBC/SQL request or BTEQ command:
EXECUTE FallBack_All;
EXECUTE FallBack_All;
*** Echo accepted.
*** Total elapsed time was 1 second.
*** Echo accepted.
*** Echo accepted.
*** Echo accepted.
*** Echo accepted.
*** Echo accepted.
*** Echo accepted.
*** Echo accepted.
*** Echo accepted.
*** Query completed. 147 rows found. 3 columns returned.
13/05/21          FallBack Tables In All Databases          10:20
DataBase          Table                                     Bytes
-----
DBC               AccessRights                                     468,992
                  AccLogRuleTb                                   8,192
                  AccLogTbl                                       4,096
                  Accounts                                         5,120
                  All                                           4,078,191,616
                  ArchiveLoggingObjsTb                         4,096
                  CollationTbl                                   14,336
                  ColumnCorrelation                             4,096
                  ConstantDefs                                   302,080
                  ConstantValues                               177,152
                  ConstraintNames                               9,216
                  CostProfiles                                   23,552
                  CostProfileTypes                             8,192
                  Dbase                                           22,528
                  DBCAssociation                                4,096
                  DBCInfoTbl                                    5,120
                  DBQLExplainTb                                4,096
                  DBQLObjTbl                                    4,096
```

DBQLogTbl	10,240
DBQLRuleCountTbl	3,072
DBQLRuleTbl	4,096
DBQLSqlTbl	4,096
DBQLStepTbl	6,144
DBQLSummaryTbl	4,096
Dependency	4,096
ErrorMsgs	711,680
ErrorTbIs	4,096
EventLog	66,560
Global	3,072
Hosts	4,096
IdCol	5,120
Indexes	61,440
InDoubtResLog	4,096
JARS	4,096
JAR_JAR_USAGE	4,096
LogonRuleTbl	6,144
MDSRecoveryTbl	2,048
Migration	3,072
Next	5,120
OldPasswords	4,096
Owners	3,072
ParentChildCorrelation	4,096
Parents	5,120
PasswordRestrictions	141,312
Profiles	4,096
RCConfiguration	4,096
RCEvent	10,240
RCMedia	2,048
ReconfigInfo	6,144
ReconfigOrder	4,096
ReconfigTableStats	6,144
ReferencedTbIs	4,096
ReferencingTbIs	4,096
RepBatchStatus	2,048
RepGroup	4,096
RepGroupTables	2,048
ResUsageIpma	2,165,760
ResUsageIvpr	148,591,616
ResUsageSawt	34,482,176
ResUsageScpu	1,602,560
ResUsageShst	8,052,736
ResUsageSldv	24,064,000

	ResUsageSpdsk	15,404,032
	ResUsageSpma	3,770,368
	ResUsageSps	3,491,786,752
	ResUsageSvdsk	7,472,128
	ResUsageSvpr	311,285,760
	RoleGrants	4,096
	Roles	4,096
	ROUTINE_JAR_USAGE	4,096
	SessionTbl	8,192
	SW_Event_Log	4,096
	SysSecDefaults	5,120
	SystemQTbl	2,048
	TableConstraints	8,192
	TDWMEventHistory	4,096
	TDWMEventLog	4,096
	TDWMExceptionLog	4,096
	TDWMSummaryLog	4,096
	TempStatistics	4,096
	TempTables	4,096
	TextTbl	509,952
	Translation	72,704
	TriggersTbl	4,096
	TSETQueryText	4,096
	TVFields	3,725,312
	TVM	6,695,936
	UDFInfo	20,480
	UDTCast	4,096
	UDTInfo	4,096
	UDTTransform	4,096
	UnResolvedReferences	4,096
SysAdmin	FastLog	4,096
	HelpSyntax	1,381,376
SYSLIB	dem	4,096
	demddl	4,096
	dempart	4,096
	installsp	35,840
	loadpkg	226,304
	savepkg	57,344
SystemFe	CleanupQCFVer	7,168
	CreateQCFVer	119,808
	opt_cost_table	22,528
	Opt_DBSctl_Table	4,096
	opt_ras_table	24,576
	Temp_ReconfigSpace	2,048

```
Sys_Calendar      CALDATES      2,652,160
*** Echo accepted.
```

FallBack_DB

Returns all tables in the specified database with fallback protection.

Syntax

```
FallBack_DB ('DBName') ;
```

Syntax Elements

DBName

The name of the database you want to be listed. The name should be in single apostrophes as indicated.

Note:

This macro is not case sensitive. Uppercase and lowercase letters can be used interchangeably.

Usage Notes

The FallBack_DB macro reports the names of the database and tables, and the size of the table in bytes.

This macro is useful for obtaining all the tables in a particular database which have fallback copies.

For a discussion of fallback, see *Teradata Vantage™ - Database Design*, B035-1094.

Example: Generating a Report of Tables with the Fallback Option in the DBC Database

The following statement generates a report of all tables with the fallback option in the DBC database:

```
Exec FallBack_DB ('DBC');
```

The resulting report follows:

```
EXECUTE FallBack_DB ('DBC');
*** Echo accepted.
*** Total elapsed time was 1 second.
*** Echo accepted.
*** Echo accepted.
```

```

*** Echo accepted.
*** Echo accepted.
*** Echo accepted.
*** Echo accepted.
*** Echo accepted.
*** Query completed. 131 rows found. 3 columns returned.

```

```
13/05/21          FallBack Tables In Database DBC          10:31
```

DataBase	Table	Bytes
-----	-----	-----
DBC	AccessRights	468,992
	AccLogRuleTbl	8,192
	AccLogTbl	4,096
	Accounts	5,120
	All	4,078,191,616
	ArchiveLoggingObjsTbl	4,096
	CollationTbl	14,336
	ColumnCorrelation	4,096
	ConstantDefs	302,080
	ConstantValues	177,152
	ConstraintNames	9,216
	CostProfiles	23,552
	CostProfileTypes	8,192
	Dbase	22,528
	DBCAssociation	4,096
	DBCInfoTbl	5,120
	DBQLExplainTbl	4,096
	DBQLObjTbl	4,096
	DBQLLogTbl	10,240
	DBQLRuleCountTbl	3,072
	DBQLRuleTbl	4,096
	DBQLSqlTbl	4,096
	DBQLStepTbl	6,144
	DBQLSummaryTbl	4,096
	Dependency	4,096
	ErrorMsgs	711,680
	ErrorTbls	4,096
	EventLog	66,560
	Global	3,072
	Hosts	4,096
	IdCol	5,120
	Indexes	61,440
	InDoubtResLog	4,096
	JARS	4,096
	JAR_JAR_USAGE	4,096

LogonRuleTbl	6,144
MDSRecoveryTbl	2,048
Migration	3,072
Next	5,120
OldPasswords	4,096
Owners	3,072
ParentChildCorrelation	4,096
Parents	5,120
PasswordRestrictions	141,312
Profiles	4,096
RCConfiguration	4,096
RCEvent	10,240
RCMedia	2,048
ReconfigInfo	6,144
ReconfigOrder	4,096
ReconfigTableStats	6,144
ReferencedTbls	4,096
ReferencingTbls	4,096
RepBatchStatus	2,048
RepGroup	4,096
RepGroupTables	2,048
ResUsageIpma	2,165,760
ResUsageIvpr	148,591,616
ResUsageSawt	34,482,176
ResUsageScpu	1,602,560
ResUsageShst	8,052,736
ResUsageSldv	24,064,000
ResUsageSpdsk	15,404,032
ResUsageSpma	3,770,368
ResUsageSps	3,491,786,752
ResUsageSvdsk	7,472,128
ResUsageSvpr	311,285,760
RoleGrants	4,096
Roles	4,096
ROUTINE_JAR_USAGE	4,096
SessionTbl	8,192
SW_Event_Log	4,096
SysSecDefaults	5,120
SystemQTbl	2,048
TableConstraints	8,192
TDWMEventHistory	4,096
TDWMEventLog	4,096
TDWMEExceptionLog	4,096
TDWMSummaryLog	4,096

TempStatistics	4,096
TempTables	4,096
TextTbl	509,952
Translation	72,704
TriggersTbl	4,096
TSETQueryText	4,096
TVFields	3,725,312
TVM	6,695,936
UDFInfo	20,480
UDTCast	4,096
UDTInfo	4,096
UDTTransform	4,096
UnResolvedReferences	4,096

*** Echo accepted.

ListErrorCodes

Returns an output of all event codes in the DBC.ErrorMsgs table.

Syntax

```
ListErrorCodes ;
```

Note:

This macro is not case sensitive. Uppercase and lowercase letters can be used interchangeably.

Usage Notes

The ListErrorCodes macro can be used to quickly get a brief description of an error code.

The error codes are in ascending order with one line of associated text. It is recommended that this macro be run after installing maintenance tapes to check the DBC.ErrorMsgs table.

Example: Generating a Report of Error Code Descriptions

The following statement generates a report that includes a one-line description of each error code:

```
Exec ListErrorCodes;
```

Due the amount of codes actually displayed, the following example shows only a portion of an actual report:

*** Query completed. 2957 rows found. 2 columns returned.

13/05/21

Error Code Summary

09:24

ErrorCode ErrorText

```

-----
2,147 Internal error detected while comparing two strings.
2,148 Internal error detected while copying a string.
2,149 Internal error detected while comparing two strings.
2,504 Bad Parameter passed to file system.
2,507 Out of spool space on disk.
2,509 AMP internal error (see backtrace)
2,511 Bad locator parameter.
2,513 A data row is too long.
2,514 Operation not allowed: %DBID.%FSTR is being used to log ARC selected p
2,516 Error in calculating/updating disk space.
2,517 Empty DataBaseSpace Table Header.
2,518 USI row indexes non existent data row.
2,519 Data row indexed multiple times by USI.
2,520 A variable field offset exceeds the next variable offset or the row le
2,525 Fatal read error encountered during Reconfig.
2,530 End Transaction Error after space accounting.
2,531 No task found for task ID/Transaction ID.
2,532 Unrecoverable read error detected.  Retry.
2,533 Shut down DBS and rebuild the disk.
2,534 System recovering from disk read error on spool file.
.
.
.
.
8,551 LOT receive is returning data
8,552 LOT receive has returned the last byte of data
8,553 LOT has timed out waiting for an out-of-order segment
8,554 LOT has stopped a connection as requested by a caller
8,555 LOT was passed a null group identifier
8,556 LOT was passed a null last-channel identifier
8,557 LOT was passed a null link-channel identifier
8,558 LOT could not link a PDE mailbox to a channel
8,559 LOT could not signal on a PDE channel
8,560 LOT could not unlink a PDE mailbox from a channel
8,561 LOT record did not belong to the endpoint
8,570 Snapshot not found.
8,571 Snapshot already exists.
8,572 Maximum snapshot log limit exceeded.
9,000 Operation not allowed: the target table is being online archived.

```

```

9,001 Operation not allowed: the target table is being restored.
9,011 Table being Archived/Restored/Copied was dropped by some other transac
9,100 Unique index violation; errors logged in %TVMID where ETC_dbql_qid = %
9,101 Referencing violation; errors logged in %TVMID where ETC_dbql_qid = %F
9,127 Index violations detected; errors logged in %TVMID where ETC_dbql_qid
*** Echo accepted.

```

ListEvent

Returns a particular event code for the set period of time.

Syntax

```
ListEvent ( Event, Fromdate [, Todate ] ) ;
```

Syntax Elements

Event

The number of the event code for which you want to search. The format of *event* is an integer, as shown in Event Code (Message) Format Interpretation.

Fromdate

The date on which the search is to start. The format is 'YYYY-MM-DD'. This parameter can be coded either as a character string, for example, '2013-05-31', or as a numeric value, for example, 1130531.

Todate

The date on which the search is to stop. The format is 'YYYY-MM-DD'. This parameter can be coded either as a character string, for example, '2013-05-31', or as a numeric value, for example, 1130531. The default value for *Todate* is the current date.

Note:

This macro is not case sensitive. Uppercase and lowercase letters can be used interchangeably.

Usage Notes

The ListEvent macro is useful for obtaining a look at all occurrences of the desired event.

Examples

Example: Generating a Report of All 033-13895-00 Events Between May and the Current Date

The following statement generates a report of all 033-13895-00 events between May 5, 2013, and the date of today:

```
Exec ListEvent (0331389500, '2013-05-05',);
```

Example: Generating a Report of All 033-13895-00 Events for the Past Week

The following statement generates a report of all 033-13895-00 events for the last seven days:

```
Exec ListEvent (0331389500, Date-06,);
```

Example: Generating a Report of 033-12116-00 Events in May

The following statement generates a report of all 033-12116-00 events between May 5, 2013 and May 18, 2013:

```
Exec ListEvent (0331211600, '2013-05-05' , '2013-05-18');
```

The resulting report follows:

```
13/05/18 All 33-12116-00 Events Between 2013-05-05 and 2013-05-18      13:51
   Date           Time Message Text
-----
13/05/18 18:54:29.380000 13/05/18 18:54:26 Running DBS Version: 12m.00.00.00
                               Event number 34-02900-00 (severity 10, category 10)
18:54:29.420000 13/05/18 18:54:26 Running PDE Version: 12m.00.00.00
                               Event number 34-02900-00 (severity 10, category 10)
18:54:29.480000 13/05/18 18:54:26 AMP 0 has been selected as the Contro
                               Event number 34-02900-00 (severity 10, category 10)
18:54:29.520000 13/05/18 18:54:26 Redistribution minicasts are enabled
                               Event number 34-02900-00 (severity 10, category 10)
18:54:29.580000 13/05/18 18:54:26 The OCS System Profiles GDO has been
                               Event number 34-02900-00 (severity 10, category 10)
18:54:29.620000 Event number 34-02900-00 (severity 10, category 10)
```

```

13/05/18 18:54:27 Initializing DBS Vprocs
18:54:44.340000 13/05/18 18:54:44 Initializing DBS Configuration
Event number 34-02900-00 (severity 10, category 10)
18:54:45.010000 13/05/18 18:54:44 Configuration is operational
Event number 34-02900-00 (severity 10, category 10)
18:54:47.490000 13/05/18 18:54:47 Starting AMP partitions
Event number 34-02900-00 (severity 10, category 10)
18:54:48.060000 send_gtw_enable: sending
18:54:48.200000 May 05 18:54:48 Teradata DBS Gateway: [105]: error logg
18:54:49.960000 May 05 18:54:49 Teradata DBS Gateway: [105]: ycgsecurit
18:54:50.000000 May 05 18:54:50 Teradata DBS Gateway: [105]: ycgsecurit
May 05 18:54:50 Teradata DBS Gateway: [105]: ycgsecurit
May 05 18:54:50 Teradata DBS Gateway: [105]: ycgsecurit
May 05 18:54:50 Teradata DBS Gateway: [105]: ycgsecurit
18:54:54.740000 Event number 34-02900-00 (severity 10, category 10)
13/05/18 18:54:52 DBS RestartKind: COLD (Do not wait fo
18:54:54.850000 Event number 34-02900-00 (severity 10, category 10)

```

ListRestart_Logon_Events

Returns all software restart events and logon-enabled events logged in the DBC.SW_Event_Log table for the set period of time.

Syntax

```
ListRestart_Logon_Events ( Fromdate [, Todate ] ) ;
```

Syntax Elements

Fromdate

The earliest date of a restart/logon that is to be listed in the report. The format is 'YYYY-MM-DD'. This parameter can be coded either as a character string, for example, '2013-05-31', or as a numeric value, for example, 1130531.

Todate

The latest date of a restart/logon that is to be listed in the report.

The format is 'YYYY-MM-DD'. This parameter can be coded either as a character string, for example, '2013-05-31', or as a numeric value, for example, 1130531. The default value for *Todate* is the current date.

Note:

This macro is not case sensitive. Uppercase and lowercase letters can be used interchangeably.

Usage Notes

The ListRestart_Logon_Events report is based on event code 033-13895-00 for restart events, and the time the "Logons are enabled" message is reported in the Software_Event_Log table for logon-enabled events.

For a complete description of these events, see the information about Parallel Database Extensions (PDE) messages in *Teradata Vantage™ - Database Messages*, B035-1096.

The output shows the date, time, and all lines of the DBC.SW_Event_Log connected with each software restart event, and one line connected with the logon-enabled event. The logon-enabled event is the time the "Logons are enabled" message is reported in the DBC.SW_Event_Log table. This macro depicts the time Vantage recovered.

Note:

The column value "TheTime" in the DBC.SW_Event_log table is the time when the database came back up after the restart. It is *not* the time the reset starts because while the database is down, the system cannot access the SW_Event_log table to log the time.

Examples

Example: Generating a Report of All Restarts and Logon-Enabled Events Between May and the Current Date

The following statement generates a report of all restarts and logon-enabled events between May 5, 2013, and the date of today:

```
Exec ListRestart_Logon_Events ('2013-05-05',);
```

Example: Generating a Report of All Restarts and Logon-Enabled Events Between January and March

The following statement generates a report of all restarts and logon-enabled events between January 1, 2013 and March 9, 2013:

```
Exec ListRestart_Logon_Events('2013-01-01','2013-03-09');
```

Example: Generating a Report of All Restarts and Logon-Enabled Events in the past 11 Days

The following statement generates a report of all restarts and logon-enabled events for the last 11 days:

```
Exec ListRestart_Logon_Events(date-10,date);
```

The resulting report follows:

```
EXECUTE ListRestart_Logon_Events (date-10,date);
*** Query completed. 17 rows found. 5 columns returned.
13/05/29                (Crashable/logon-enabled Events)                09:42
                        Between 2013-05-19 and 2013-05-29
      Date              Time Message Text
-----
13/05/21 00:55:17.162969 13/05/21 00:55:17 Logons are enabled
13/05/26 03:06:04.232870 13/05/26 03:06:04 Logons are enabled
                22:40:15.943210 13/05/26 22:40:15 Logons are enabled
13/05/27                00:39:52.300814 13/05/27 00:39:52 Logons are enabled
                01:07:21.218686 13/05/27 01:07:21 Logons are enabled
                01:16:14.637825 13/05/27 01:16:14 Logons are enabled
```

ListSoftware_Event_Log

Returns all events (hardware and software) found in DBC.SW_Event_Log table for the set period of time.

Syntax

```
ListSoftware_Event_Log ( Fromdate [, Todate ] ) ;
```

Syntax Elements

Fromdate

The date on which the search is to start. The format is 'YYYY-MM-DD'. This parameter can be coded either as a character string, for example, '2013-05-31', or as a numeric value, for example, 1130531.

Todate

The date on which the search is to stop. The format is 'YYYY-MM-DD'. This parameter can be coded either as a character string, for example, '2013-05-31', or as a numeric value, for example, 1130531. The default value for *Todate* is the current date.

Note:

This macro is not case sensitive. Uppercase and lowercase letters can be used interchangeably.

Usage Notes

The ListSoftware_Event_Log macro is useful for obtaining a quick look at the contents of DBC.SW_Event_Log for a given time frame.

Note:

This report can get large very quickly.

Examples

Example: Generating a Copy of the Software_Event_Log Between January 1 and the Current Date

The following statement would give a copy of the Software_Event_Log between January 1, 2013, and the date of today:

```
Exec ListSoftware_Event_Log ('2013-01-01',);
```

Example: Generating a Copy of the Software_Event_Log for the Past Week

The following statement would give a copy of the Software_Event_Log for the last seven days:

```
Exec ListSoftware_Event_Log (Date-6,);
```

Example: Creating a Software_Event_Log Copy for a Specific Date

The following statement would give a copy of the Software_Event_Log for a specific date in the past, specifically May 21, 2013:

```
Exec ListSoftware_Event_Log ('2013-05-21', '2013-05-21');
```

The resulting report follows:

```

Exec ListSoftware_Event_Log ('2013-05-21', '2013-05-21');
*** Echo accepted.
*** Total elapsed time was 1 second.
*** Echo accepted.
*** Echo accepted.
*** Echo accepted.
*** Echo accepted.
*** Echo accepted.
*** Echo accepted.
*** Echo accepted.
*** Echo accepted.
*** Query completed. 30 rows found. 6 columns returned.
13/05/28   Software Event Log Listing Between 2013-05-21 and 2013-05-21
13:40
Date      Time          Event Tag                      Message
Text
-----
13/05/21   00:55:12.569072  34-02900-00          13/05/21 00:55:12
Starting AMP partitions
          00:55:13.049967  34-02900-00
13/05/21 00:55:13                                DBS RestartKind: COLD
(Do not wait for                                transaction recovery)
          00:55:13.090731  34-02900-00
13/05/21 00:55:13                                Voting for transaction
recovery
          00:55:13.221652  34-02900-00
13/05/21 00:55:13                                Recovery session 1
contains 181 rows on
          00:55:13.825460  34-02900-00
13/05/21 00:55:13                                Starting transaction
recovery
          00:55:13.978402  34-02900-00
13/05/21 00:55:13                                Starting PE partitions
          00:55:14.255917  34-02900-00
13/05/21 00:55:14                                Completed transaction
recovery
          00:55:16.582331  34-02900-00
13/05/21 00:55:16                                System is operational
          00:55:16.674762  34-08072-00
Event number 34-08072-00 (severity 0, category 10), occurred on

```

```

Thu May 21 00:55:16 2013, at 001-01 (Vproc
8192, partition 10,
task 9450) in system lnxsm6 in Module
gtwgateway, version
PDE:12r.00.00.00,TDBMS:12r.00.
34-08072-00
34-08072-00
gtwsecurity.c @334 (50332208): Thu May 21 00:55:16 2013
34-08072-00
tdgss library version: 12.0.0.0
34-08072-00
00:55:16.675132 34-08072-00
Event number 34-08072-00 (severity 0, category 10), occurred on
Thu May 21 00:55:16 2013, at 001-01 (Vproc
8192, partition 10,
task 9450) in system lnxsm6 in Module
gtwgateway, version
PDE:12r.00.00.00,TDBMS:12r.00.
34-08072-00
34-08072-00
gtwsecurity.c @376 (50332208): Thu May 21 00:55:16 2013
34-08072-00 tdgss
supported methods: TD1
34-08072-00
00:55:16.675308 34-08072-00
Event number 34-08072-00 (severity 0, category 10), occurred on
Thu May 21 00:55:16 2013, at 001-01 (Vproc
8192, partition 10,
task 9450) in system lnxsm6 in Module
gtwgateway, version
PDE:12r.00.00.00,TDBMS:12r.00.
34-08072-00
34-08072-00
gtwsecurity.c @376 (50332208): Thu May 21 00:55:16 2013
34-08072-00 tdgss
supported methods: TD2
34-08072-00
00:55:16.675465 34-08072-00
Event number 34-08072-00 (severity 0, category 10), occurred on
Thu May 21 00:55:16 2013, at 001-01 (Vproc
8192, partition 10,
task 9450) in system lnxsm6 in Module
gtwgateway, version
PDE:12r.00.00.00,TDBMS:12r.00.

```

```

34-08072-00
34-08072-00
gtwsecurity.c @376 (50332208): Thu May 21 00:55:16 2013
34-08072-00                                tdgss
supported methods: ldap
34-08072-00
00:55:17.162739 34-02900-00
13/05/21 00:55:17 Users are logged on
00:55:17.162969 34-02900-00
13/05/21 00:55:17 Logons are enabled
*** Echo accepted.

```

MiniCylPacks

Returns all MiniCylPacks related events logged in the DBC.Software_Event_Log table for a set period of time.

Syntax

```
MiniCylPacks ( Fromdate [, Todate ] ) ;
```

Syntax Elements

Fromdate

The date on which the search is to start. The format is 'YYYY-MM-DD'. This parameter can be coded either as a character string, for example, '2013-05-31', or as a numeric value, for example, 1130531.

Fromdate

The date on which the search is to stop.

The format is 'YYYY-MM-DD'. This parameter can be coded either as a character string, for example, '2013-05-31', or as a numeric value, for example, 1130531. The default value for *Todate* is the current date.

Note:

This macro is not case sensitive. Uppercase and lowercase letters can be used interchangeably.

Usage Notes

The MiniCylPacks report is based on event codes 034-05141-00, 034-05142-00, and 034-05143-00.

You can use the ListErrorCodes macro to get a brief description of these events. For a detailed description, see the information about database messages in *Teradata Vantage™ - Database Messages*, B035-1096.

This macro is useful to determine whether MiniCylPacks were done successfully, and whether they were performed in anticipation of their use or while there were other tasks waiting. This information is determined based on the specific event code, as follows:

Event Code	Description
5141	Indicates an anticipated, successful freeing of cylinders.
5142	Indicates a successful freeing of cylinders, but caused other tasks to wait.
5143	Indicates no success at freeing cylinders, and that the event caused other tasks to wait.

Note:

These event codes can also be viewed in the ListErrorCodes report (see [ListErrorCodes](#)).

Examples

Example: Generating a MiniCylPacks Report for the Last Seven Days

The following statement generates a MiniCylPacks report for the last seven days:

```
Exec MiniCylPacks (Date-6,);
```

Example: Generating a MiniCylPacks Report Between January and March

The following statement gives a report of the MiniCylPacks between January 1, 2013 and March 10, 2013:

```
Exec MiniCylPacks ('2013-01-01', '2013-03-10');
```

Example: Generating a MiniCylPacks Report Between January and the Current Date

The following statement gives a report of the MiniCylPacks between January 21, 2013, and the date of today:

```
Exec MiniCylPacks ('2013-01-21',);
```

The resulting report follows:

```

Exec MiniCylPacks ('2013-01-21',);

*** Query completed. 14 rows found. 6 columns returned.

13/01/21      MiniCylPacks Performed Between 2013-01-21 and 2013-01-21      21:49
Vproc        Date              Time              Event
-----
      1              13/01/21  21:29:26.610000  34-05142-00
***** Number of MiniCylPack log messages On vproc 1 =      1
      2              13/01/21  21:29:25.630000  34-05142-00
***** Number of MiniCylPack log messages On vproc 2 =      1
      3              13/01/21  21:29:24.950000  34-05142-00
***** Number of MiniCylPack log messages On vproc 3 =      1
      4              13/01/21  21:29:24.670000  34-05142-00
***** Number of MiniCylPack log messages On vproc 4 =      1
      5              13/01/21  21:29:24.660000  34-05142-00
***** Number of MiniCylPack log messages On vproc 5 =      1
      6              13/01/21  21:29:25.130000  34-05142-00
***** Number of MiniCylPack log messages On vproc 6 =      1
      7              13/01/21  21:29:25.250000  34-05142-00
***** Number of MiniCylPack log messages On vproc 7 =      1

*** Query completed. No rows found.
*** Echo accepted.

```

NoFallback_All

Returns all tables in the system which have no fallback protection.

Syntax

```
NoFallback_All ;
```

Note:

This macro is not case sensitive. Uppercase and lowercase letters can be used interchangeably.

Usage Notes

The NoFallback_All macro reports the names of the database and tables, and the size of the table in bytes.

This macro is useful for obtaining a look at all the tables on the system which do not have fallback copies.

For a discussion of fallback, see *Teradata Vantage™ - Database Design*, B035-1094.

Example: Generating a Report of Tables Without the Fallback Option

The following statement generates a report of all tables without the fallback option:

```
Exec NoFallback_All;
```

The resulting report follows:

```

      Exec NoFallback_All;

*** Query completed. 20 rows found. 3 columns returned.
13/05/21          NoFallback Tables In All Databases          09:36
DataBase          Table                                     Bytes
-----
DBC               Acctg                                     7,680
                  ChangedRowJournal                       2,048
                  DataBaseSpace                           104,448
                  LocalSessionStatusTable                 6,144
                  LocalTransactionStatusTable             2,048
                  OrdSysChngTable                         2,048
                  ReconfigJournal                         2,048
                  RecoveryLockTable                       2,048
                  RecoveryPJTable                         2,048
                  SavedTransactionStatusTable              2,048
                  SysRcvStatJournal                       25,088
                  TransientJournal                        15,859,712
                  UtilityLockJournalTable                 2,048

*** Echo accepted.
```

NoFallback_DB

Returns all tables in the specified database with no fallback protection.

Syntax

```
NoFallback_DB ('DBName') ;
```

Syntax Elements

DBName

The name of the database you want to be listed. The name should be in single apostrophes as indicated.

Note:

This macro is not case sensitive. Uppercase and lowercase letters can be used interchangeably.

Usage Notes

The NoFallback_DB macro reports the names of the database and tables, and the size of the table in bytes.

This macro is useful for obtaining a look at all the tables in a particular database which do not have fallback copies.

For a discussion of fallback, see *Teradata Vantage™ - Database Design*, B035-1094.

Example: Generating a Report of All Tables Without the Fallback Option in the DBC Database

The following statement generates a report of all tables without the fallback option in the DBC database:

```
Exec NoFallback_DB ('DBC');
```

The resulting report follows:

```
*** Query completed. 15 rows found. 3 columns returned.
13/05/21          NoFallback Tables In Database DBC          10:43
DataBase          Table                                       Byte
-----
DBC               Acctg                                       7,6
                  ChangedRowJournal                         2,0
                  DataBaseSpace                             104,4
                  LocalSessionStatusTable                   6,1
                  LocalTransactionStatusTable               2,0
                  OrdSysChngTable                           2,0
                  ReconfigJournal                           2,0
```

RecoveryLockTable	2,0
RecoveryPJTable	2,0
SavedTransactionStatusTable	2,0
SysRcvStatJournal	25,0
TransientJournal	15,859,7
UtilityLockJournalTable	2,0

PackDisk

Returns all user-issued PackDisk-related events logged in the DBC.Software_Event_Log table for a set period of time.

Syntax

```
PackDisk ( Fromdate [, Todate ] ) ;
```

Syntax Elements

Fromdate

The date on which the search is to start. The format is 'YYYY-MM-DD'. This parameter can be coded either as a character string, for example, '2013-05-31', or as a numeric value, for example, 1130531.

Todate

The date on which the search is to stop. The format is 'YYYY-MM-DD'. This parameter can be coded either as a character string, for example, '2013-05-31', or as a numeric value, for example, 1130531. The default value for *Todate* is the current date.

Note:

This macro is not case sensitive. Uppercase and lowercase letters can be used interchangeably.

Usage Notes

The PackDisk macro report is based on event codes 034-05144-00 and 034-05145-00.

For a complete description of these events, see the information about database messages in *Teradata Vantage™ - Database Messages*, B035-1096. You can also view the events in the ListErrorCodes report (see [“ListErrorCodes”](#)).

This macro is useful to list all PackDisk requests (indicated by event code 05144), if any, and whether these requests were completed successfully (indicated by event code 05145). You can initiate PackDisk requests using the Ferret utility.

For more information on Ferret, see *Teradata Vantage™ - Database Utilities*, B035-1102.

Examples

Example: Generating a PackDisk Report for the Last Seven Days

The following statement would give a report of the PackDisks for the last seven days:

```
Exec PackDisk (Date-6,);
```

Example: Generating a PackDisk Report Between January and March

The following statement would give a report of the PackDisk between January 1, 2013 and March 10, 2013:

```
Exec PackDisk ('2013-01-01' , '2013-03-10');
```

Example: Generating a PackDisk Report Between January and the Current Date

The following statement would give a report of the PackDisk between January 21, 2013, and the date of today.

```
Exec PackDisk ('2013-01-21',);
```

Note:

The date specified is the same as the date the macro was run.

The resulting report follows:

```
Exec PackDisk ('2013-01-21',);
*** Query completed. 23 rows found. 6 columns returned.
13/01/21      PackDisks Performed Between 2013-01-21 and 2013-01-21 21:49
Vproc      Date      Time      Event
-----
      0  13/01/21  21:30:20.390000  34-05144-00
                21:30:37.290000  34-05145-00
                -----
***** Number of PackDisks On vproc 0 =      1
      1  13/01/21  21:30:20.380000  34-05144-00
```

```

                21:30:34.740000    34-05145-00
                -----
***** Number of PackDisks On vproc 1 =      1
      2  13/01/21  21:30:20.380000    34-05144-00
                21:30:37.840000    34-05145-00
                -----
***** Number of PackDisks On vproc 2 =      1
      3  13/01/21  21:30:20.360000    34-05144-00
                21:30:53.320000    34-05145-00
                -----
***** Number of PackDisks On vproc 3 =      1
      4  13/01/21  21:30:20.360000    34-05144-00
                21:30:51.280000    34-05145-00
                -----
***** Number of PackDisks On vproc 4 =      1
      5  13/01/21  21:30:20.350000    34-05144-00
                21:30:39.520000    34-05145-00
                -----
***** Number of PackDisks On vproc 5 =      1
      6  13/01/21  21:30:20.320000    34-05144-00
                21:30:30.910000    34-05145-00
                -----
***** Number of PackDisks On vproc 6 =      1
      7  13/01/21  21:30:20.320000    34-05144-00
                -----
***** Number of PackDisks On vproc 7 =      1

```

Using SystemFE Macros

SystemFE Macros are designed to help you monitor Vantage. The macros can be organized into the categories shown in the following table. Information in the rest of this section describes specific scenarios and how the macros can help.

Category	Macro Name
Summary	FallBack_All FallBack_DB MiniCylPacks NoFallBack_All NoFallBack_DB PackDisk
Event logs	AllRestarts BynetEvents DiskEvents EventCount ListEvent ListRestart_Log_Events ListSoftware_Event_Log
Release related	ListErrorCodes
Reconfiguration	ReconfigCheck

Database Preventive Maintenance

Running Macros Weekly

For preventive maintenance, Teradata recommends you run the following macros in the following order at least on a weekly basis:

- EventCount
- AllRestarts
- DiskEvents
- MiniCylPacks
- PackDisk
- ListEvent

Generating the Preventive Maintenance Report

Executing the macros in the above order provides a Preventive Maintenance Report listing summary information followed by detailed information for the week. Used in this manner, the macros should be executed with 'Date-7' as the FromDate and 'Date-1' as the ToDate. For example, the following BTEQ statement provides a report listing all the events that occurred on the database during the specified time period (that is, the last 7 days excluding the date of today) with the number of occurrences:

```
Exec EventCount (Date-7, Date-1);
```

If you have access to a client user ID or access to the system console, you can write a BTEQ script in the above syntax. Then, you can run the script on a weekly basis without changing the date parameter.

Analyzing Preventive Maintenance Reports

In general, SystemFE macros are made available as a method for producing quick reports relevant to system messages stored in the DBC.SW_Event_Log. Informational, Warning, and Critical system messages are stored in the DBC.SW_Event_Log until they are manually deleted. Although not the only or definitive resource, this repository is good to consult for determining the relative health of the database platform.

Since system messages are stored in the DBC.SW_Event_Log until they are deleted manually, the System Administrator should develop a plan for deleting old messages from the Event_Log to keep it from growing to an unusually large size. Event_Log messages older than 90 days often are not useful. Proactive health checks should be performed every 30 days (not to exceed 90 days). On the reactive problem discovery side, 90-day-old messages rarely are helpful in determining current problems.

When the Preventive Maintenance Report is generated, the data must be analyzed. The order in which the macros were run is important. The analysis can be divided into three steps, each step encompassing more detail than the last. The sections below describe the steps.

Step 1 - Identifying High Frequency

The ListErrorCodes macro gives an output of all event codes in the DBC.ErrorMsgs table. The EventCount macro lists the codes and their frequency of occurrence during the specified time period. This aids in identifying any application or transient hardware problems that do not cause restarts. For example:

```
3610: Internal Error, Please do not resubmit last request
```

means that a request was aborted and a snapshot dump taken without creating a restart. You should report a high frequency of occurrence of this event to the Teradata Support Center so that they can determine the appropriateness of requesting the site to migrate to a later software release.

If further information about these events (such as processor number, and so forth) is needed, use the ListSoftware_Event_Log or ListEvent macro to list all the occurrences of that particular event code in full detail.

For information about backtrace, examine the message log files.

The location of the event information is in log file `/var/log/messages`.

Step 2 - Database Restarts

The most important events from the event log to consider are database restarts. They are listed as '33-13855-00' and '33-13892-00' messages on the reports obtained from the EventCount macro and the AllRestarts macro. Restarts indicate that problems have caused database outages. Note the event codes and perform problem determination. For all events, verify that the Teradata Support Center has opened a Call Log to report their occurrence. If the events are hardware restarts, you can perform a more detailed analysis by following step 3.

For a complete description and resolution of these events, see "Parallel Database Extensions (PDE) Messages" *Teradata Vantage™ - Database Messages*, B035-1096.

Step 3 - Subsystem Detail

The following macros display detailed reports concerning events related to the hardware processor subsystems:

- DiskEvents
- MiniCylPacks
- PackDisk
- ListEvent

You should examine these reports to reveal any evidence of intermittently failing hardware that has not caused database restarts.

Other Resources for Analyzing System Messages

Use the following resources to analyze system messages:

Resource	Description
Database Messages	An analysis of the message log file should always accompany any results found when using the SystemFE macros. The log file shows all system messages or activities leading up to and following those messages found in the DBC.SW_Event_Log. For information on the location of the message log file, see "Usage Notes" .
Administration Workstation (AWS)	Attached to all multi-node database systems, the AWS has a Customer Support Facility (CSF) knowledge base which often provides information useful for determining the best course of action to take when addressing system messages.
SHOWSPACE command (Ferret utility)	This command reports on the amount of used and available system space. For more information, see Ferret in <i>Teradata Vantage™ - Database Utilities</i> , B035-1102.
ResUsage macros	Executing these macros provides assistance in analyzing system performance. For more information, see <i>Teradata Vantage™ - Resource Usage Macros and Tables</i> , B035-1099.

Determining Fallback Protection

To determine which tables or tables within a database do not have fallback protection, run the NoFallback_All or NoFallback_DB macro. To determine which tables or tables within a database have fallback protection, run the FallBack_All or FallBack_DB macro. For a general discussion of fallback, see *Teradata Vantage™ - Database Design*, B035-1094.

Reconfiguring AMPs

To reconfigure the number of AMPs in your system, run the ReconfigCheck macro to get an estimated number of bytes per AMP required for a reconfiguration operation. Reconfigurations are performed by Teradata Services personnel.

Updating Vantage Software

After updating Vantage software, run the ListErrorCodes macro to obtain a current list of the error codes.

How to Read Syntax

This document uses the following syntax conventions.

Syntax Convention	Meaning
KEYWORD	Keyword. Spell exactly as shown. Many environments are case-insensitive. Syntax shows keywords in uppercase unless operating system restrictions require them to be lowercase or mixed-case.
<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. Example: 10045
[x]	x is optional.
[x y]	You can specify x, y, or nothing.
{ x y }	You must specify either x or y.
x [...]	You can repeat x, separating occurrences with spaces. Example: x x x See note after table.
x [, ...]	You can repeat x, separating occurrences with commas. Example: x, x, x See note after table.
x [delimiter...]	You can repeat x, separating occurrences with specified delimiter. Examples: <ul style="list-style-type: none"> If <i>delimiter</i> is semicolon: x; x; x If <i>delimiter</i> is { , OR }, you can do either of the following: <ul style="list-style-type: none"> x, x, x x OR x OR x See note after table.

Note:

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

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

You can repeat x. Do not repeat KEYWORD.

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

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

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

SystemFE Macro Listings

This section shows the BTEQ script listings used to create each macro for reference purposes. Refer to these listings to understand the source of the data of the macro and how the macro processes the data.

To obtain the most up-to-date listing of the macros that reside in the SystemFE database, use the following query:

```
select tvn.TVMNameI, tvn.TableKind
from dbc.dbase, dbc.tvn
where   dbase.DatabaseID = tvn.DatabaseID
        AND dbase.Databasename = 'SystemFE'
        AND tvn.TableKind = 'M'
order by 1;
```

To view the definition of a macro, use SHOW MACRO *macroname*.

AllRestarts

```
REPLACE MACRO SystemFE.AllRestarts
    (Fromdate (date,          FORMAT 'YYYY-MM-DD'),
     Todate   (date, DEFAULT DATE, FORMAT 'YYYY-MM-DD')) AS
(
echo 'set defaults;';
echo 'set format on;';
echo 'set width 80;';
echo 'set NULL as "-";';
echo 'set separator 2;';
echo 'set pagelength 60;';
echo 'set suppress on 3,4;';
echo 'set omit on 1,2;';
echo 'set heading "&DATE||(Crashable Events)||&TIME//Between &1 and      &2";';
Select   :Fromdate      (Format 'YYYY-MM-DD')
        , :Todate        (Format 'YYYY-MM-DD')
        , TheDate        (Title  'Date')
        , TheTime        (Title  'Time')
        , Text
From     DBC.Software_Event_Log
Where    TheDate between :fromdate and :todate
And      (Event_Tag = 0331385500 OR
          Event_Tag = 0331389200 )
```

```

Order By  TheDate, TheTime, Line;

echo 'set omit off;';
echo 'set heading "Total Restarts";';
Select    Count(*)                (Title 'Restarts')
From      DBC.Software_Event_Log
Where     TheDate between :fromdate and :todate
And       Line = 1
And       (Event_Tag = 0331385500 OR
           Event_Tag = 0331389200);

echo 'set defaults;';
);

```

BynetEvents

```

REPLACE MACRO SystemFE.BynetEvents
  (Fromdate (date,                FORMAT 'YYYY-MM-DD'),
   Todate   (date, DEFAULT DATE, FORMAT 'YYYY-MM-DD')) AS
(
echo 'set defaults;';
echo 'set format on;';
echo 'set width 254;';
echo 'set NULL as "-";';
echo 'set separator 2;';
echo 'set pagelength 60;';
echo 'set omit on 1,2;';
echo 'set suppress on 3,4;';
echo 'set heading "&DATE          BYNET Events Between &1 and &2          &TIME|| ||";';
Select    :Fromdate      (Format 'YYYY-MM-DD')
          , :Todate      (Format 'YYYY-MM-DD')
          , TheDate      (Title  'Date')
          , TheTime      (Title  'Time')
          , Text         (Title  'Message Text')
From      DBC.software_event_log
where     ((Event_tag>=0331110000 AND Event_tag<=0331112100)
          OR (Event_tag >= 0331290000 AND Event_tag <= 0331297800)
          OR (Event_tag >= 0331300000 AND Event_tag <= 0331302100)
          OR (Text like '%bynet%'))
And       TheDate BETWEEN :fromdate AND :Todate
Order by  TheDate, TheTime, Line;
echo 'set defaults;';
);

```

DiskEvents

```

REPLACE MACRO SystemFE.DiskEvents
  (Fromdate (date,                FORMAT 'YYYY-MM-DD'),
   Todate   (date, DEFAULT DATE, FORMAT 'YYYY-MM-DD')) AS
  (
    echo 'set defaults;';
    echo 'set format on;';
    echo 'set width 150;';
    echo 'set NULL as "-";';
    echo 'set separator 2;';
    echo 'set pagelength 60;';
    echo 'set omit on 1,2;';
    echo 'set suppress on 3,4,5;';
    echo 'set heading "&DATE||Disk Events Between &1 and &2||&TIME";';
    Select   :Fromdate      (Format 'YYYY-MM-DD')
             , :Todate      (Format 'YYYY-MM-DD')
             , TheDate      (Title  'Date')
             , TheTime      (Title  'Time')
             , Text         (Title  'Message Text')
    FROM     DBC.Software_Event_Log
    Where    Event_Tag in (0331103600, 0331103800, 0331109400)
    And      TheDate BETWEEN :fromdate AND :Todate
    Order by TheDate, TheTime;
    echo 'set defaults;';
  );

```

EventCount

```

REPLACE MACRO SystemFE.EventCount
  (Fromdate (date,                FORMAT 'YYYY-MM-DD'),
   Todate   (date, DEFAULT DATE, FORMAT 'YYYY-MM-DD')) AS
  (
    echo 'set defaults;';
    echo 'set format on;';
    echo 'set width 80;';
    echo 'set NULL as "-";';
    echo 'set separator 2;';
    echo 'set omit on 1,2;';
    echo 'set pagelength 60;';
    echo 'set heading "&DATE||Total Of Each Event||&TIME//Between &1 and  &2";';
  );

```

```

Select      :Fromdate      (Format 'YYYY-MM-DD')
            , :ToDate      (Format 'YYYY-MM-DD')
            , Event_Tag
            , Count(*)      (Title '    Total    //Occurrences')
From        DBC.Software_Event_Log
Where       TheDate between :fromdate and :todate
And         Line=1
And         ( Severity > 20
Or  Event_Tag = '033-13855-00'
Or  Event_Tag = '033-13892-00' )
Group By    Event_Tag
Order By    4 Desc;
echo 'set defaults;';
);

```

FallBack_All

```

REPLACE MACRO SystemFE.FallBack_All AS
(
echo 'set defaults;';
echo 'set format on;';
echo 'set width 80;';
echo 'set NULL as "-";';
echo 'set separator 2;';
echo 'set pagelength 60;';
echo 'set suppress on 1;';
echo 'set heading "&DATE||FallBack Tables In All Databases||&TIME";';
Select      DBC.AllSpace.DataBaseName
            (Title 'DataBase', Format 'X(15)')
            , DBC.AllSpace.TableName          (Title 'Table')
            , Sum(CurrentPerm) (Title 'Bytes')
From        DBC.AllSpace, DBC.Tables
Where       DBC.Allspace.Databasename = DBC.Tables.Databasename
And         DBC.Allspace.Tablename     = DBC.Tables.Tablename
And         DBC.Tables.ProtectionType = 'F'
And         DBC.Tables.TableKind ^= 'J'
Order By    1,2
Group By    1,2;
echo 'set defaults;';
);

```

FallBack_DB

```

REPLACE MACRO SystemFE.FallBack_DB (DBName (Char(30))) AS
(
echo 'set defaults;';
echo 'set format on;';
echo 'set width 82;';
echo 'set NULL as "-";';
echo 'set separator 2;';
echo 'set pagelength 60;';
echo 'set suppress on 1;';
echo 'set heading "&DATE||FallBack Tables In Database &1||&TIME";';
Select      DBC.AllSpace.DataBaseName
              (Title 'DataBase', Format 'X(30)')
            , DBC.AllSpace.TableName          (Title 'Table')
            , Sum(CurrentPerm) (Title 'Bytes')
From        DBC.AllSpace, DBC.Tables
Where       DBC.AllSpace.Databasename = :DBName
And         DBC.Allspace.Databasename = DBC.Tables.Databasename
And         DBC.Allspace.Tablename      = DBC.Tables.Tablename
And         DBC.Tables.ProtectionType = 'F'
Order By   1,2
Group By   1,2;
echo 'set defaults;';
);

```

ListErrorCodes

```

REPLACE MACRO SystemFE.ListErrorCodes AS
(
echo 'set defaults;';
echo 'set format on;';
echo 'set width 80;';
echo 'set NULL as "-";';
echo 'set separator 2;';
echo 'set pagelength 60;';
echo 'set heading "&DATE||Error Code Summary||&TIME";';
Select      ErrorCode (Format '---,--9')
            , ErrorText
From        DBC.ErrorMsgs
Where       ErrorCode > 0000
Order By   ErrorCode;

```

```
echo 'set defaults;';
);
```

ListEvent

```
REPLACE MACRO SystemFE.ListEvent
(Event(Integer),
  Fromdate (date,          FORMAT 'YYYY-MM-DD'),
  Todate   (date, DEFAULT DATE, FORMAT 'YYYY-MM-DD')) AS
(
echo 'set defaults;';
echo 'set format on;';
echo 'set width 80;';
echo 'set NULL as "-";';
echo 'set separator 2;';
echo 'set pagelength 60;';
echo 'set suppress on 3,4;';
echo 'set omit on 1,2,5;';
echo 'set heading "&DATE||All &5 Events Between &1 and &2||&TIME";';
Select      :Fromdate      (Format 'YYYY-MM-DD')
            , :Todate       (Format 'YYYY-MM-DD')
            , TheDate       (Title  'Date')
            , TheTime       (Title  'Time')
            , Event_Tag
            , Text
From        DBC.Software_Event_Log
Where       Event_Tag=:Event
And         TheDate Between :fromdate and :todate
Order By    TheDate, TheTime, Line;
echo 'set defaults;';
);
```

ListRestart_Logon_Events

```
REPLACE MACRO SystemFE.ListRestart_Logon_Events
(Fromdate (date,          FORMAT 'YYYY-MM-DD'),
  Todate   (date, DEFAULT DATE, FORMAT 'YYYY-MM-DD')) AS
(
echo 'set defaults;';
echo 'set format on;';
echo 'set width 80;';
echo 'set NULL as "-";';
```

```

echo 'set separator 2;';
echo 'set pagelength 60;';
echo 'set suppress on 3,4;';
echo 'set omit on 1,2;';
echo 'set heading "&DATE||(Crashable/logon-enabled Events)||&TIME//Between &1
and &2";';
Select      :Fromdate      (Format 'YYYY-MM-DD')
            , :ToDate      (Format 'YYYY-MM-DD')
            , TheDate      (Title  'Date')
            , TheTime      (Title  'Time')
            , Text          (Title  'Message Text')
From        DBC.Software_Event_Log
Where       TheDate between :fromdate and :todate
And         (Event_tag = 0331385500
            or Event_tag = 0331389200
            or (Event_tag = 340290000 AND text like '%logons are enabled%'))
Order By    TheDate, TheTime, Line;
);

```

ListSoftware_Event_Log

```

REPLACE MACRO SystemFE.ListSoftware_Event_Log
  (Fromdate (date,                FORMAT 'YYYY-MM-DD'),
   Todate   (date, DEFAULT DATE, FORMAT 'YYYY-MM-DD')) AS
(
echo 'set defaults;';
echo 'set format on;';
echo 'set width 254;';
echo 'set NULL as "-";';
echo 'set separator 2;';
echo 'set pagelength 60;';
echo 'set omit on 1,2;';
echo 'set suppress on 3,4;';
echo 'set heading "&DATE          Software Event Log Listing Between &1 and &2
&TIME|| ||";';
Select      :Fromdate      (Format 'YYYY-MM-DD')
            , :ToDate      (Format 'YYYY-MM-DD')
            , TheDate      (Title  'Date')
            , TheTime      (Title  'Time')
            , Event_Tag
            , Text
From        DBC.Software_Event_Log
Where       TheDate between :fromdate and :todate

```

```
Order By TheDate, TheTime, Line;
echo 'set defaults;';
);
```

MiniCylPacks

```
REPLACE MACRO SystemFE.MiniCylPacks
  (Fromdate (date,                FORMAT 'YYYY-MM-DD'),
   Todate   (date, DEFAULT DATE, FORMAT 'YYYY-MM-DD')) AS
(
  echo 'set defaults;';
  echo 'set format on;';
  echo 'set width 80;';
  echo 'set NULL as "-";';
  echo 'set separator 2;';
  echo 'set pagelength 60;';
  echo 'set suppress on 3,4;';
  echo 'set omit on 1,2;';
  echo 'set heading "&DATE||MiniCylPacks Performed Between &1 and &2|&TIME"
  ';
  Select      :Fromdate      (Format 'YYYY-MM-DD')
             , :Todate        (Format 'YYYY-MM-DD')
             , VProc
             , TheDate        (Title 'Date')
             , TheTime        (Title 'Time')
             , Event_Tag      (Title 'Event')

  From        DBC.Software_Event_Log
  Where        Event_Tag in (0340514100, 0340514200, 0340514300)
  And          Line=1
  And          DBC.Software_Event_Log.TheDate Between :Fromdate and :Todate
  With         Count(*) (Title '***** Number of MiniCylPack log messages On vproc &3
  =', FORMAT 'zzzz9')
  By           3
  Order By     3,4,5;
  echo 'set defaults;';
  );
```

NoFallback_All

```
REPLACE MACRO SystemFE.NoFallback_All AS
(
  echo 'set defaults;';
```

```

echo 'set format on;';
echo 'set width 80;';
echo 'set NULL as "-";';
echo 'set separator 2;';
echo 'set pagelength 60;';
echo 'set suppress on 1;';
echo 'set heading "&DATE||NoFallBack Tables In All Databases||&TIME";';
Select      DBC.AllSpace.DataBaseName
              (Title 'DataBase', Format 'X(15)')
              , DBC.AllSpace.TableName          (Title 'Table')
              , Sum(CurrentPerm) (Title 'Bytes')
From        DBC.AllSpace, DBC.Tables
Where       DBC.Allspace.Databasename = DBC.Tables.Databasename
And         DBC.Allspace.Tablename    = DBC.Tables.Tablename
And         DBC.Tables.ProtectionType = 'N'
And         DBC.Tables.TableKind ^= 'J'
Order By   1,2
Group By   1,2;
echo 'set defaults;';
);

```

NoFallBack_DB

```

REPLACE MACRO SystemFE.NoFallBack_DB (DBName (Char(30))) AS
(
echo 'set defaults;';
echo 'set format on;';
echo 'set width 80;';
echo 'set NULL as "-";';
echo 'set separator 2;';
echo 'set pagelength 60;';
echo 'set suppress on 1;';
echo 'set heading "&DATE||NoFallBack Tables In Database &1||&TIME";';
Select      DBC.AllSpace.DataBaseName
              (Title 'DataBase', Format 'X(30)')
              , DBC.AllSpace.TableName          (Title 'Table')
              , Sum(CurrentPerm) (Title 'Bytes')
From        DBC.AllSpace, DBC.Tables
Where       DBC.AllSpace.Databasename = :DBName
And         DBC.Allspace.Databasename = DBC.Tables.Databasename
And         DBC.Allspace.Tablename    = DBC.Tables.Tablename
And         DBC.Tables.ProtectionType = 'N'
Order By   1,2

```

```

Group By 1,2;
echo 'set defaults;';
);

```

PackDisk

```

REPLACE MACRO SystemFE.PackDisk
  (Fromdate (date,                FORMAT 'YYYY-MM-DD'),
   Todate   (date, DEFAULT DATE, FORMAT 'YYYY-MM-DD')) AS
(
  echo 'set defaults;';
  echo 'set format on;';
  echo 'set width 80;';
  echo 'set NULL as "-";';
  echo 'set separator 2;';
  echo 'set pagelength 60;';
  echo 'set suppress on 3,4;';
  echo 'set omit on 1,2;';
  echo 'set heading "&Date||PackDisks Performed Between &1 and &2||&TIME";';
  Select      :Fromdate      (Format 'YYYY-MM-DD')
             , :Todate        (Format 'YYYY-MM-DD')
             , VProc
             , TheDate         (Title 'Date')
             , TheTime         (Title 'Time')
             , Event_Tag
  From        DBC.Software_Event_Log
  Where       Event_Tag in (0340514400, 0340514500)
  And         Line=1
  And         DBC.Software_Event_Log.TheDate Between :Fromdate and :Todate
  With        (Count(*) +1)/2 (Title '***** Number of PackDisks On vproc &3 ='
                               , FORMAT 'zzzz9')

  By          3
  Order By 3,4,5;
  echo 'set defaults;';
);

```

ReconfigCheck

```

REPLACE MACRO SystemFE.ReconfigCheck (DestAMPs (SmallInt)) AS
(
  Delete From SystemFE.Temp_ReconfigSpace all;
  Insert Into SystemFE.Temp_ReconfigSpace

```

```

select Sum(Currentperm)
From DBC.TableSizeV;
echo 'set defaults;';
echo 'set format on;';
echo 'set width 80;';
echo 'set NULL as "-";';
echo 'set separator 2;';
echo 'set pagelength 60;';
echo 'set omit on 1;';
echo 'set heading "&DATE||Bytes Required For &1 AMP
        System||&TIME";'
;
select :DestAmps,
(Perspace/:DestAmps)
(Title 'Bytes//Required//Per AMP', Format 'zz,zzz,z99,999')
From SystemFE.Temp_ReconfigSpace;
echo 'set defaults;';
);

```

Additional Information

Teradata Links

Link	Description
https://docs.teradata.com/	Search Teradata Documentation, customize content to your needs, and download PDFs. Customers: Log in to access Orange Books.
https://support.teradata.com	One-stop source for Teradata community support, software downloads, and product information. Log in for customer access to: <ul style="list-style-type: none">• Community support• Software updates• Knowledge articles
https://www.teradata.com/University/Overview	Teradata education network
https://support.teradata.com/community	Link to Teradata community