Contents

Preface ............................................................................................................................................................... xiii
   Audience .......................................................................................................................................................... xiii
   Related documents ....................................................................................................................................... xiv
   Conventions .................................................................................................................................................. xiv
   Documentation Accessibility ....................................................................................................................... xv

What's New........................................................................................................................................................ xvi
   New features in release 18.1.2.1.0 ............................................................................................................ xvi
   New features in release 18.1.1.1.0 ............................................................................................................ xvi

1 TimesTen Instance Configuration File
   Configuration attributes ................................................................................................................................. 1-1
      Required attributes ................................................................................................................................ 1-1
      Optional attributes .................................................................................................................................. 1-2

2 Connection Attributes
   Required privileges for attributes ............................................................................................................... 2-1
   List of Attributes ....................................................................................................................................... 2-2
   Data store attributes ................................................................................................................................ 2-10
      Data Source Name.................................................................................................................................. 2-11
      DataStore ........................................................................................................................................... 2-12
      DatabaseCharacterSet ............................................................................................................................... 2-13
      Supported character sets ......................................................................................................................... 2-13
      Description ............................................................................................................................................ 2-16
      Driver ................................................................................................................................................... 2-17
      Durability ............................................................................................................................................. 2-18
      GridLogRecoveryThreshold .................................................................................................................... 2-19
      LogDir .................................................................................................................................................. 2-20
      Preallocate .......................................................................................................................................... 2-21
      ReplicationApplyOrdering ....................................................................................................................... 2-22
      ReplicationParallelism ............................................................................................................................. 2-24
      Temporary ............................................................................................................................................ 2-25
   First connection attributes ......................................................................................................................... 2-26
      AutoCreate .......................................................................................................................................... 2-27
      CkptFrequency ................................................................................................................................... 2-28
CkptLogVolume .......................................................................................................................... 2-30
CkptRate .................................................................................................................................... 2-32
CkptReadThreads ...................................................................................................................... 2-34
Connections ............................................................................................................................... 2-35
EpochInterval ............................................................................................................................ 2-36
ForceConnect ............................................................................................................................ 2-37
ForceDisconnectEnabled ........................................................................................................ 2-38
LogAutoTruncate ...................................................................................................................... 2-39
LogBufMB ................................................................................................................................... 2-40
LogBufParallelism .................................................................................................................... 2-41
LogFileSize ................................................................................................................................ 2-42
LogFlushMethod ......................................................................................................................... 2-43
LogPurge ..................................................................................................................................... 2-44
MemoryLock ............................................................................................................................... 2-45
Overwrite .................................................................................................................................. 2-47
PermSize ..................................................................................................................................... 2-48
RecoveryThreads ....................................................................................................................... 2-49
TempSize ..................................................................................................................................... 2-50

General connection attributes ................................................................................................ 2-51
CommitBufferSizeMax ............................................................................................................... 2-52
ConnectionName ....................................................................................................................... 2-53
CreateEpochAtCommit ............................................................................................................. 2-54
DDLReplicationAction ............................................................................................................. 2-55
DDLReplicationLevel ................................................................................................................ 2-56
Diagnostics ............................................................................................................................... 2-58
DurableCommits ....................................................................................................................... 2-59
IncludeInCore ......................................................................................................................... 2-60
Isolation ..................................................................................................................................... 2-61
LockLevel .................................................................................................................................... 2-63
LockWait .................................................................................................................................... 2-64
OptimizerHint ............................................................................................................................ 2-65
PermWarnThreshold .................................................................................................................. 2-66
PrivateCommands ..................................................................................................................... 2-67
PWDCrypt .................................................................................................................................. 2-68
QueryThreshold ......................................................................................................................... 2-69
ReplicationTrack ....................................................................................................................... 2-70
SQLQueryTimeout ...................................................................................................................... 2-71
TempWarnThreshold .................................................................................................................. 2-72
UID and PWD ............................................................................................................................ 2-73
WaitForConnect .......................................................................................................................... 2-74

NLS general connection attributes ........................................................................................ 2-75
ConnectionCharacterSet ............................................................................................................ 2-76
NLS_LENGTH_SEMANTICS ........................................................................................................ 2-77
NLS_NCHAR_CONV_EXCP ....................................................................................................... 2-78
NLS_SORT .................................................................................................................................... 2-79

Supported linguistic sorts ....................................................................................................... 2-80

PL/SQL first connection attributes ......................................................................................... 2-83
PLSQL_MEMORY_ADDRESS .............................................................. 2-84
PLSQL_MEMORY_SIZE ............................................................... 2-86
PLSQL_general connection attributes ........................................ 2-87
PLSCOPE_SETTINGS ................................................................. 2-88
PLSQL_CCFLAGS ................................................................. 2-89
PLSQL_CONN_MEM_LIMIT ....................................................... 2-90
PLSQL_OPTIMIZE_LEVEL ....................................................... 2-91
PLSQL_SESSION_CACHED_CURSORS ........................................ 2-92
PLSQL_TIMEOUT ................................................................. 2-93
TimesTen Cache first connection attributes ............................ 2-94
CacheAWTMethod ................................................................. 2-95
TimesTen Cache database attributes ........................................ 2-97
CacheAWTParallelism .......................................................... 2-98
TimesTen Cache general connection attributes ......................... 2-99
  DynamicLoadEnable .......................................................... 2-100
  DynamicLoadErrorMode .................................................... 2-101
  OracleNetServiceName .................................................... 2-102
  OraclePWD ................................................................. 2-103
  PassThrough ............................................................... 2-104
  RACCallback .............................................................. 2-107
TimesTen Client connection attributes ..................................... 2-108
  TCP_Port ................................................................. 2-109
  TCP_Port2 ............................................................... 2-110
  TTC_ConnectTimeout .................................................. 2-111
  TTC_FailoverPortRange ................................................. 2-112
  TTC_NoReconnectOnFailover ......................................... 2-113
  TTC_Random_Selection ................................................ 2-114
  TTC_REDIRECT .......................................................... 2-115
  TTC_Redirect_Limit .......................................................... 2-116
  TTC_RollbackRequiredOnFailover ................................... 2-117
  TTC_Server or TTC_Server1 .............................................. 2-118
  TTC_Server2, TTC_ServerN ................................................. 2-119
  TTC_Server_DSN .......................................................... 2-121
  TTC_Server_DSN2 .......................................................... 2-122
  TTC_TCP_KEEPALIVE_INTVL_MS ..................................... 2-123
  TTC_TCP_KEEPALIVE_PROBES ....................................... 2-124
  TTC_TCP_KEEPALIVE_TIME_MS ..................................... 2-125
  TTC_Timeout .............................................................. 2-126
TimesTen Server connection attributes .................................... 2-127
  MaxConnsPerServer ...................................................... 2-128
  ServersPerDSN ............................................................ 2-129
  ServerStackSize .......................................................... 2-130

3 Built-In Procedures

List of built-in procedures .......................................................... 3-2
  ttAgingLRUConfig .......................................................... 3-10
  ttAgingScheduleNow ....................................................... 3-13
ttApplicationContext ................................................................. 3-15
ttBackupStatus ............................................................................. 3-16
ttBlockInfo .................................................................................... 3-18
ttBookmark ................................................................................... 3-19
ttCacheADGStandbyStateGet ......................................................... 3-21
ttCacheADGStandbyStateSet ......................................................... 3-23
ttCacheADGStandbyTimeoutGet ...................................................... 3-25
ttCacheADGStandbyTimeoutSet ...................................................... 3-26
ttCacheAllowFlushAwtSet ......................................................... 3-28
ttCacheAutorefIntervalStatsGet .................................................. 3-30
ttCacheAutorefresh ................................................................. 3-33
ttCacheAutorefreshLogDefrag ....................................................... 3-35
ttCacheAutorefreshStatsGet ....................................................... 3-37
ttCacheAutorefreshSelectLimit .................................................... 3-41
ttCacheAutorefreshXactLimit ....................................................... 3-43
ttCacheAWTMonitorConfig ............................................................ 3-46
ttCacheAWTThresholdGet ............................................................ 3-48
ttCacheAWTThresholdSet ............................................................ 3-49
ttCacheCheck ........................................................................ 3-51
ttCacheConfig ........................................................................... 3-54
ttCacheDbCgStatus ................................................................. 3-58
ttCacheDDLTrackingConfig ........................................................... 3-60
ttCachePolicyGet ................................................................. 3-61
ttCachePolicySet ................................................................. 3-63
ttCachePropagateFlagSet ............................................................ 3-65
ttCacheSqlGet ........................................................................... 3-67
ttCacheStart ........................................................................ 3-69
ttCacheStop ........................................................................... 3-71
ttCacheUidGet ........................................................................... 3-73
ttCacheUidPwdSet ................................................................. 3-74
ttCkpt ..................................................................................... 3-76
ttCkptBlocking ........................................................................... 3-78
ttCkptConfig ........................................................................... 3-80
ttCkptHistory .............................................................................. 3-83
ttCommitBufferStats ............................................................... 3-88
ttCommitBufferStatsReset .......................................................... 3-90
ttCompact ................................................................. 3-91
ttComputeTabSizes ..................................................................... 3-92
ttConfiguration ........................................................................ 3-94
ttContext .................................................................................. 3-97
ttDataStoreStatus ................................................................. 3-98
ttDBCompactConfig ............................................................... 3-100
ttDBConfig ................................................................. 3-103
ttDBConcurrencyModeGet .......................................................... 3-106
ttDBConcurrencyModeSet .......................................................... 3-108
ttDistributeProgress ............................................................... 3-110
ttDurableCommit ................................................................. 3-112
4 TimesTen Scaleout Utilities

Overview ................................................................................................................. 4-1
Required authentication and authorization for utilities ............................................. 4-1
  Required user authentication for utilities ................................................................. 4-1
  Required privileges for executing utilities ............................................................... 4-2
Utilities list ................................................................................................................ 4-2
ttGridAdmin ............................................................................................................... 4-3
  Grid model ............................................................................................................... 4-4
  Grid objects and object naming ............................................................................. 4-5
  Address formats ..................................................................................................... 4-6
  Database management operations ......................................................................... 4-7
  Command timeouts and waits .................................................................................. 4-8
  ttGridAdmin operations ......................................................................................... 4-9
  Help and general options ....................................................................................... 4-10
  Return values ......................................................................................................... 4-11
  Backup and restore operations .............................................................................. 4-12
Back up a database (dbBackup) .......................................................... 4-12
Delete a database backup (dbBackupDelete) ................................ 4-13
Display the status of a database backup (dbBackupStatus) ........ 4-13
Restore a database (dbRestore) .................................................. 4-14
Display the status of a database restore (dbRestoreStatus) .... 4-15

Connectable operations ................................................................. 4-17
Create a connectable (connectableCreate) ......................... 4-17
Delete a connectable (connectableDelete) ......................... 4-18
Export a connectable (connectableExport) .......................... 4-19
List connectables (connectableList) ........................................ 4-19
Modify a connectable (connectableModify) ....................... 4-20

Data space group operations ..................................................... 4-22
Get recommendations for data space group assignments (dataSpaceGroupSuggest) .... 4-22

Database definition operations ............................................... 4-24
Create a database definition (dbdefCreate) ....................... 4-24
Delete a database definition (dbdefDelete) ....................... 4-26
Export a database definition (dbdefExport) ..................... 4-26
List database definitions (dbdefList) .................................. 4-27
Modify a database definition (dbdefModify) ..................... 4-28

Database operations ................................................................. 4-30
Close a database (dbClose) .................................................... 4-30
Create a database (dbCreate) .................................................. 4-31
Destroy a database (dbDestroy) ............................................. 4-31
Force all connections to disconnect (dbDisconnect) ............ 4-35
Check status of forced disconnection (dbDisconnectStatus) .. 4-38
Set or modify the distribution scheme of a database (dbDistribute) .... 4-39
List databases (dbList) .......................................................... 4-43
Load a database into memory (dbLoad) ................................. 4-43
Open a database (dbOpen) ..................................................... 4-44
Monitor the status of a database (dbStatus) ....................... 4-45
Unload a database (dbUnload) .............................................. 4-53

Grid operations .............................................................. 4-55
Export sys.odbc.ini for client/server connections outside grid (gridClientExport) .... 4-55
Create a grid (gridCreate) ....................................................... 4-55
Display information about the grid (gridDisplay) ............... 4-58
Get diagnostic information about the grid (gridDump) ........... 4-59
Collect log information about the grid (gridLogCollect) .... 4-59
Modify grid settings (gridModify) ....................................... 4-60
Configure SSH (gridSshConfig) ............................................. 4-61

Host operations .............................................................. 4-64
Create a host (hostCreate) ....................................................... 4-64
Delete a host (hostDelete) ..................................................... 4-66
Execute a command or script on grid hosts (hostExec) ....... 4-67
List all hosts in the model (hostList) .................................... 4-68
Modify a host (hostModify) .................................................. 4-69

Import and export operations ............................................... 4-71
Export a database (dbExport) .............................................. 4-71
Repository operations.......................................................................................................... 4-108
Management instance operations ........................................................................................... 4-107
Installation operations........................................................................................................... 4-106
Physical group operations ..................................................................................................... 4-105
Model operations .................................................................................................................. 4-105
Membership operations ........................................................................................................ 4-105
Instance operations............................................................................................................... 4-105
Export instance configuration attributes (instanceConfigExport) ........................................ 4-104
Import instance configuration attributes (instanceConfigImport) ........................................ 4-104
Create an instance (instanceCreate) .................................................................................... 4-103
Delete an instance (instanceDelete) ..................................................................................... 4-103
Execute a command or script on grid instances (instanceExec) .......................................... 4-103
List instances (instanceList) ............................................................................................... 4-103
Modify an instance (instanceModify) .................................................................................. 4-103
Display status of instances (instanceStatus) ....................................................................... 4-103
Management instance operations .......................................................................................... 4-103
Start the active management instance (mgmtActiveStart) .................................................. 4-103
Stop the active management instance (mgmtActiveStop) .................................................... 4-103
Switch the active management instance (mgmtActiveSwitch) ............................................ 4-103
Examine management instances (mgmtExamine) .................................................................. 4-103
Start the standby management instance (mgmtStandbyStart) ............................................. 4-103
Stop the standby management instance (mgmtStandbyStop) .............................................. 4-103
Display status of management instances (mgmtStatus) ....................................................... 4-103
Membership operations ........................................................................................................ 4-103
Export the membership configuration file (membershipConfigExport) .............................. 4-102
Import the membership configuration file (membershipConfigImport) ............................. 4-102
Model operations ................................................................................................................ 4-102
Apply the latest version of the model (modelApply) ............................................................. 4-101
Compare models (modelCompare) ....................................................................................... 4-101
Export a version of a model (modelExport) ......................................................................... 4-101
Import a version of the model (modelImport) ....................................................................... 4-101
List model versions (modelList) ......................................................................................... 4-101
Oracle Database operations ................................................................................................ 4-101
Export a sqlnet file (SQLNetExport) ................................................................................... 4-100
Import a sqlnet file (SQLNetImport) .................................................................................... 4-100
Export TNS names (TNSNamesExport) ................................................................................ 4-100
Import TNS names (TNSNamesImport) ................................................................................ 4-100
Physical group operations .................................................................................................... 4-100
Create a physical group (physicalCreate) .......................................................................... 4-100
Delete a physical group (physicalDelete) .......................................................................... 4-100
List physical groups (physicalList) .................................................................................... 4-100
Repository operations........................................................................................................... 4-100
Attach a repository (repositoryAttach) ............................................................................... 4-100
5 Utilities

Overview ........................................................................................................................................... 5-1

Required authentication and authorization for utilities ................................................................. 5-1

Required user authentication for utilities .................................................................................... 5-1

Required privileges for executing utilities .................................................................................... 5-2

Utilities List ................................................................................................................................... 5-2

ttAdmin .......................................................................................................................................... 5-5

Help, version, and query options .................................................................................................. 5-7

Set database loading policies ........................................................................................................ 5-9

Set cache policies .......................................................................................................................... 5-11

Set replication policies .................................................................................................................. 5-14

Force disconnect ............................................................................................................................ 5-16

ttAdoptStores ................................................................................................................................. 5-18

ttBulkCp ......................................................................................................................................... 5-20

ttBackup ........................................................................................................................................ 5-34

ttCapture ........................................................................................................................................ 5-37

ttCheck ......................................................................................................................................... 5-39

ttCWAdmin ..................................................................................................................................... 5-42

ttDaemonAdmin ............................................. 5-48

ttDaemonLog ................................................................................................................................. 5-50

ttDestroy ........................................................................................................................................ 5-54

ttInstallationCheck ...................................................................................................................... 5-56

ttInstallDSN ................................................................................................................................. 5-57

ttInstanceCreate ......................................................................................................................... 5-59

ttDestroy ........................................................................................................................................ 5-61

ttInstanceId .................................................................................................................................. 5-62

ttIsq1 ............................................................................................................................................... 5-64

Commands ....................................................................................................................................... 5-66

Syntax for the IF-THEN-ELSE command construct ................................................................. 5-79

Syntax for the WHENEVER SQLERROR command ................................................................ 5-80

Set/show attributes ....................................................................................................................... 5-82

Comment syntax .......................................................................................................................... 5-88

Command shortcuts ...................................................................................................................... 5-89

Parameters ..................................................................................................................................... 5-90

Examples ........................................................................................................................................ 5-91

ttMigrate ...................................................................................................................................... 5-97

ttRepAdmin ................................................................................................................................... 5-110

Help and version information ....................................................................................................... 5-112

Database information .................................................................................................................... 5-113

Subscriber database operations .................................................................................................. 5-114

Duplicate a database .................................................................................................................... 5-116

Wait for updates to complete .................................................................................................... 5-120
6 System Limits

System limits and defaults ................................................................. 6-1
Limits on number of open files ......................................................... 6-3
Path names .......................................................................................... 6-3

Index
Oracle TimesTen In-Memory Database (TimesTen) is a relational database that is memory-optimized for fast response and throughput. The database resides entirely in memory at runtime and is persisted to disk storage.

- Oracle TimesTen In-Memory Database in classic mode, or TimesTen Classic, refers to single-instance and replicated databases (as in previous releases).
- Oracle TimesTen In-Memory Database in grid mode, or TimesTen Scaleout, refers to multiple-instance distributed databases. TimesTen Scaleout is a grid of interconnected hosts running instances that work together to provide fast access, fault tolerance, and high availability for in-memory data. A grid contains one or more databases and each database is distributed across all instances of the grid.
- TimesTen alone refers to both classic and grid modes (such as in references to TimesTen utilities, releases, distributions, installations, actions taken by the database, and functionality within the database).
- TimesTen Application-Tier Database Cache, or TimesTen Cache, is an Oracle Database Enterprise Edition option. TimesTen Cache is ideal for caching performance-critical subsets of an Oracle database into cache tables within TimesTen databases for improved response time in the application tier. Cache tables can be read-only or updatable. Applications read and update the cache tables using standard Structured Query Language (SQL) while data synchronization between the TimesTen database and the Oracle database is performed automatically. TimesTen Cache offers all of the functionality and performance of TimesTen Classic, plus the additional functionality for caching Oracle Database tables.

TimesTen supports standard application interfaces JDBC, ODBC, and ODP.NET; Oracle interfaces PL/SQL, OCI, and Pro*C/C++; and the TimesTen TTClasses library for C++.

**Audience**

This document provides a reference for TimesTen attributes, built-in procedures, and utilities.

This document is intended for readers with a basic understanding of database systems.
Related documents

TimesTen documentation is available at https://docs.oracle.com/database/timesten-18.1.

Oracle Database documentation is also available on the Oracle documentation website.
This may be especially useful for Oracle Database features that TimesTen supports but
does not attempt to fully document, such as OCI and Pro*C/C++.

Conventions

TimesTen supports multiple platforms. Unless otherwise indicated, the information in
this guide applies to all supported platforms. The term Windows applies to all
supported Windows platforms. The term UNIX applies to all supported UNIX
platforms. The term Linux is used separately. Refer to "Platforms and compilers" in
Oracle TimesTen In-Memory Database Release Notes: Refer to the README.html in your
installation directory for specific platform versions supported by TimesTen.

---

**Note:** In TimesTen documentation, the terms "data store" and
"database" are equivalent. Both terms refer to the TimesTen database.

---

This document uses the following text conventions:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>italic</em></td>
<td>Italic type indicates terms defined in text, book titles, or emphasis.</td>
</tr>
<tr>
<td>monospace</td>
<td>Monospace type indicates code, commands, URLs, function names,</td>
</tr>
<tr>
<td></td>
<td>attribute names, directory names, file names, text that appears on the</td>
</tr>
<tr>
<td></td>
<td>screen, or text that you enter.</td>
</tr>
<tr>
<td><em>italic monospace</em></td>
<td>Italic monospace type indicates a placeholder or a variable in a code</td>
</tr>
<tr>
<td></td>
<td>example for which you specify or use a particular value. For example:</td>
</tr>
<tr>
<td></td>
<td>LIBS = -L{installation_home}/install/lib -ltten</td>
</tr>
<tr>
<td></td>
<td>Replace <em>timesten_home</em> with the path to the TimesTen instance home</td>
</tr>
<tr>
<td></td>
<td>directory.</td>
</tr>
<tr>
<td>[ ]</td>
<td>Square brackets indicate that an item in a command line is optional.</td>
</tr>
<tr>
<td>{ }</td>
<td>Curly braces indicated that you must choose one of the items separated</td>
</tr>
<tr>
<td></td>
<td>by a vertical bar (</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>An ellipsis ( . . . ) after an argument indicates that you may use more</td>
</tr>
<tr>
<td></td>
<td>than one argument on a single command line. An ellipsis in a code</td>
</tr>
<tr>
<td></td>
<td>example indicates that what is shown is only a partial example.</td>
</tr>
<tr>
<td>% or $</td>
<td>The percent sign or dollar sign indicates the UNIX shell prompt,</td>
</tr>
<tr>
<td></td>
<td>depending on the shell that is used.</td>
</tr>
<tr>
<td>#</td>
<td>The number (or pound) sign indicates the UNIX root prompt.</td>
</tr>
</tbody>
</table>

In addition, TimesTen documentation uses the following special conventions:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>installation_dir</em></td>
<td>The path that represents the directory where TimesTen is installed.</td>
</tr>
<tr>
<td><em>timesten_home</em></td>
<td>The path that represents the home directory of a TimesTen instance.</td>
</tr>
</tbody>
</table>
Release or RR

The first two parts in a release number, with or without the dot. The first two parts of a release number represent a major TimesTen release. For example, 181 or 18.1 represents TimesTen Release 18.1.

DSN

TimesTen data source name (for the TimesTen database).

Note: TimesTen release numbers are reflected in items such as TimesTen utility output, file names, and directory names. These are subject to change with every minor or patch release, and the documentation cannot always be up to date. The documentation seeks primarily to show the basic form of output, file names, directory names, and other code that may include release numbers. You can confirm the current release number by looking at the Release Notes or executing the ttVersion utility.

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

## Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.
What's New

This section summarizes the new features of TimesTen In-Memory Database Release 18.1 that are documented in this guide and provides links to more information.

New features in release 18.1.2.1.0

The following features are documented for the new "forced disconnect" feature, which forces all user connections to the specified database to be disconnected. This is useful, for example, prior to maintenance operations. Closing connections is mandatory to ensure a smooth shutdown and no data loss.

- ttGridAdmin dbDisconnect and dbDisconnectStatus commands (TimesTen Scaleout). See "Force all connections to disconnect (dbDisconnect)" on page 4-35.

These additional new features are documented:

- ttGridAdmin dbDistribute -resync option attempts to resynchronize metadata in the user database with metadata in the active management instance in case the state of a dbDistribute -apply command is unknown. "Set or modify the distribution scheme of a database (dbDistribute)" on page 4-39.
- PLSQL_SESSION_CACHED_CURSORS connection attribute specifies the number of session cursors to cache. A user may adjust the setting to free up space not currently needed in the cache. See "PLSQL_SESSION_CACHED_CURSORS" on page 2-92.
- ttDBCConfig built-in procedure supports a new parameter, PLSQL_OPEN_CURSORS, which specifies the maximum number of PL/SQL cursors that can be open in a session at one time. See "ttDBCConfig" on page 3-103.
- ttCkptHistory built-in procedure returns new columns: reason, logsPurged, bookmarkName, and additional_details. See "ttCkptHistory" on page 3-83.
- ttInstallDSN utility, for TimesTen Scaleout, generates a Windows client DSN for each of one or more entries in the provided input file and installs them into the ODBC control panel as system DSNs. See "ttInstallDSN" on page 5-57.

New features in release 18.1.1.1.0

This release of TimesTen contains a configuration file associated with each instance. See Chapter 1, "TimesTen Instance Configuration File" for details about this file.
The book contains descriptions of new and changed utilities:

- `ttGridAdmin utility`
- `ttGridRollout utility`
- `ttInstallationCheck utility`
- `ttInstallDSN utility`
- `ttInstanceCreate utility`
- `ttInstanceDelete utility`
- `ttInstanceModify utility`
- Changes to `ttXactAdmin utility`
- Changes to `ttBulkCp utility`
- Changes to `ttStats utility`

The book contains descriptions of new attributes:

- `TTC_TCP_KEEPALIVE_TIME_MS, TTC_TCP_KEEPALIVE_INTVL_MS, TTC_TCP_KEEPALIVE_PROBES, TTC_Random_Selection, TTC_REDIRECT, TTC_Redirect_Limit` and `TTC_RollbackRequiredOnFailover` connection attributes
- `Durability` connection attribute
- `EpochInterval` and `CreateEpochAtCommit` connection attributes
- `GridLogRecoveryThreshold` connection attribute
- `IncludeInCore` connection attribute
- `OptimizerHint` connection attribute
- Changes to `ttStats utility`

The book contains descriptions of new built-in procedures:

- `ttCacheADGStandbyStateGet`
- `ttCacheADGStandbyStateSet`
- `ttCacheADGStandbyTimeoutGet`
- `ttCacheADGStandbyTimeoutSet`
- `ttEpochCreate`
- `ttEpochSessionGet`
- `ttHeapinfo`
- `ttLatchStatsGet`
TimesTen Instance Configuration File

This chapter describes the TimesTen configuration file attributes. The configuration files contains the attributes of the TimesTen instance and their values.

The `timesten.conf` file is located at `$/TIMESTEN_HOME/conf/timesten.conf`.

Each line of the configuration file consists of one `name=value` pair.

In TimesTen Classic, you can change values through various TimesTen utilities or connection attributes, at instance creation time or through editing this file. For more information about changing values by editing this file, see "Working with the TimesTen Data Manager Daemon" in the Oracle TimesTen In-Memory Database Operations Guide.

In TimesTen Scaleout, you can change the values in this file by using the `ttGridAdmin instanceConfigImport` command. For more details, see "Import instance configuration attributes (instanceConfigImport)" on page 4-80. Do not edit the configuration file by hand.

Configuration attributes

Some attributes in the configuration file are required. Other attributes are optional.

Required attributes

The configuration file contains these required attributes:

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Description</th>
<th>Attribute value and type</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>instance_name</td>
<td>The name of the TimesTen instance. This attribute is optional for client only instances.</td>
<td>An ASCII name from 1 to 255 characters long.</td>
<td>instance1</td>
</tr>
<tr>
<td>daemon_port</td>
<td>The port number on which the TimesTen daemon for this instance listens. TimesTen ignores this attribute for client only instances.</td>
<td>A valid TCP port number.</td>
<td>6624</td>
</tr>
</tbody>
</table>
### Optional attributes

The configuration file can contain these optional attributes:

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Description</th>
<th>Attribute value and type</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>timesten_release</td>
<td>The release that created the instance home and that can use the instance home.</td>
<td>A TimesTen release number.</td>
<td>The installed TimesTen release number.</td>
</tr>
<tr>
<td>client_only</td>
<td>Indicates if the instance only supports client operations.</td>
<td>Yes or No</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>If yes, the instance supports only client operations.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>This entry is added by the ttInstanceCreate utility and must not be changed.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tns_admin</td>
<td>In a TimesTen Cache environment, the directory that contains the TNS_ADMIN settings to allow TimesTen to communicate with the Oracle database.</td>
<td>A directory</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>This attribute is required in a TimesTen Cache environment only.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>This entry is added by the ttInstanceCreate utility and can be changed using the ttInstanceModify utility.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attribute name</td>
<td>Description</td>
<td>Attribute value and type</td>
<td>Default</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>--------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>listen_addr and</td>
<td>listen_addr indicates the IPv4 address where the TimesTen daemon and servers listen.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>listen_6_addr</td>
<td>listen_6_addr indicates the IPv6 address where the TimesTen daemon and servers listen. By default, TimesTen supports IPv6.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>enableipv6</td>
<td>Indicates if TimesTen is configured to support IPv6.</td>
<td>String, 1 or 0</td>
<td>1</td>
</tr>
<tr>
<td>supportlog</td>
<td>The location of the TimesTen daemon support log file.</td>
<td>String, timesten_home/ diag/ttmesg.log</td>
<td></td>
</tr>
<tr>
<td>userlog</td>
<td>The location of the TimesTen daemon user log file. The file location can be modified using the ttDaemonLog utility.</td>
<td>String, timesten_home/ diag/tterrors.log</td>
<td></td>
</tr>
</tbody>
</table>
| max_support_log_files | The TimesTen main daemon automatically rotates the files once they get to a specific size.  
|                   | This attribute specifies the maximum number of support log files to keep. | Integer, 10             |         |
| max_user_log_files | The TimesTen main daemon automatically rotates the files once they get to a specific size.  
|                   | This attribute specifies the maximum number of user log files to keep. | Integer, 10             |         |
| max_support_log_size_mb | The maximum size of a TimesTen daemon support log file.  |
| max_user_log_size_mb | The maximum size of a TimesTen daemon user log file.  | Integer, 10 MB          |         |
| show_date         | Indicates if the date should be prepended on all support and user log entries.  
|                   | 1 indicates that the date should be prepended to every record in the user and support logs                            |
|                   | 0 indicates that the date is not prepended to every record in the user and support logs. | 1 or 0, 1               |         |
## Configuration attributes

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Description</th>
<th>Attribute value and type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>server_port</td>
<td>The port number on which the TimesTen server for this instance listens if you want the instance to be associated with a server.</td>
<td>A valid TCP port number.</td>
<td>6625</td>
</tr>
<tr>
<td>facility</td>
<td>If daemon logs are sent to syslog, the facility to be used. Possible name values are: auth, cron, daemon, local0-local7, lpr, mail, news, user, or uucp. The value is specified by the ttDaemonlog utility</td>
<td>String</td>
<td>None</td>
</tr>
<tr>
<td>server_stack_size</td>
<td>The client server stack size. The value is set by and can be changed by the ServerStackSize connection attribute.</td>
<td>Integer</td>
<td>256KB</td>
</tr>
<tr>
<td>servers_per_dsn</td>
<td>The number of servers per DSN for client server implementations. The value is set by and can be changed by the ServersPerDSN connection attribute</td>
<td>Integer</td>
<td>1</td>
</tr>
<tr>
<td>max_conns_per_server</td>
<td>The maximum number of client server connections for each TimesTen server. The value set by and can be changed by the MaxConnsPerServer connection attribute</td>
<td>Integer</td>
<td>1</td>
</tr>
<tr>
<td>server_pool</td>
<td>The number of processes that the TimesTen server should prespawn and keep in a reserve pool. If not specified, no processes are prespawned. The value is set by the MaxConnsPerServer connection attribute</td>
<td>Integer indicating the number of processes to prespawn</td>
<td>None</td>
</tr>
<tr>
<td>Attribute name</td>
<td>Description</td>
<td>Attribute value and type</td>
<td>Default</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>allow_network_files</td>
<td>Indicates if data access on NFS-mounted systems is allowed.</td>
<td>1 or 0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>By default, TimesTen systems cannot access data across NFS-mounted systems.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>On Linux x86 and Solaris, you can access checkpoint and transaction log files on NFS-mounted systems.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 indicates data access on NFS-mounted systems is allowed.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 indicates that access on NFS-mounted systems is not allowed.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>For more information, see &quot;Using NFS-mounted systems for checkpoint and transaction log files&quot; in the Oracle TimesTen In-Memory Database Operations Guide.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TimesTen ignores this attribute on platforms other than Linux x86 and Solaris.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>server_shmipc</td>
<td>Indicates if the TimesTen Server should use shared memory IPC.</td>
<td>1 or 0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1 indicates that the server uses shared memory IPC.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 indicates that the server does not use shared memory IPC.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>For more information, see &quot;Using shared memory for Client/Server IPC&quot; in the Oracle TimesTen In-Memory Database Operations Guide.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>server_shmsize</td>
<td>The size of the shared memory IPC.</td>
<td>Integer</td>
<td>64 MB</td>
</tr>
<tr>
<td>min_subs</td>
<td>The minimum number of subdaemons TimesTen spawns.</td>
<td>Integer</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>The main TimesTen daemon spawns subdaemons dynamically as they are needed.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>max_subs</td>
<td>The maximum number of subdaemons TimesTen spawns.</td>
<td>Integer</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>The main TimesTen daemon spawns subdaemons dynamically as they are needed.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attribute name</td>
<td>Description</td>
<td>Attribute value and type</td>
<td>Default</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>enable_policy_inactive</td>
<td>1 indicates that if there are too many failures to automatically load a database, TimesTen puts the database into a “policy inactive” mode. This is the default. 0 indicates that the RAM policy is one of Always, Manual or InUse. For more information, see “Specifying a RAM policy” in the Oracle TimesTen In-Memory Database Operations Guide.</td>
<td>Integer</td>
<td>1</td>
</tr>
<tr>
<td>noserverlog</td>
<td>Turns off logging of connects and disconnects from the client applications. 1 indicates logging is on. 0 indicates that logging is off.</td>
<td>Integer</td>
<td>1</td>
</tr>
</tbody>
</table>
The ODBC standard defines four connection attributes:

- DSN
- Driver
- UID
- PWD

For a description of the ODBC definition of these attributes, see the appropriate ODBC manual for your platform:

- Microsoft ODBC 3.0 Programmer’s Reference and SDK Guide
- Microsoft ODBC 2.0 Programmer’s Reference and SDK Guide

This chapter describes all the connection attributes defined by TimesTen. To view the names and values of most attributes specified in the connection string, an application can use the `ttConfiguration` built-in procedure.

---

**Note:** According to the ODBC standard, when an attribute occurs multiple times in a connection string, the first value specified is used, not the last value.

---

On Windows, `False` means the check box is unchecked and `True` means the check box is checked.

On UNIX and Linux systems and Linux systems, `False` means the attribute value is set to 0 and `True` means the attribute value is set to 1.

The following sections provide details on all TimesTen attributes, which are first listed in tables in "List of Attributes" on page 2-2. Following the tables, this chapter describes each attribute in detail.

- Required privileges for attributes
- List of Attributes

### Required privileges for attributes

Only the instance administrator can change a first connection attribute to a value other than the one currently in effect. (No privileges are required to change `AutoCreate` and `ForceConnect`.)
List of Attributes

This section includes the tables:

- Table 2–1, "Data store attributes"
- Table 2–2, "First connection attributes"
- Table 2–3, "General connection attributes"
- Table 2–4, "NLS general connection attributes"
- Table 2–5, "PL/SQL first connection attributes"
- Table 2–6, "PL/SQL general connection attributes"
- Table 2–7, "TimesTen Cache first connection attributes"
- Table 2–8, "TimesTen Cache database attributes"
- Table 2–9, "TimesTen Cache general connection attributes"
- Table 2–10, "TimesTen Client connection attributes"
- Table 2–11, "TimesTen Server connection attributes"

**Table 2–1 Data store attributes**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Source Name</td>
<td>A name that identifies the specific attributes of a connection to the database.</td>
<td>None</td>
</tr>
<tr>
<td>DataStore</td>
<td>Identifies the physical database.</td>
<td>None</td>
</tr>
<tr>
<td>DatabaseCharacterSet</td>
<td>Identifies the character set used by the database. This attribute is required at database creation time.</td>
<td>None</td>
</tr>
<tr>
<td>Description</td>
<td>A statement that identifies the use of the data source name.</td>
<td>None</td>
</tr>
<tr>
<td>Driver</td>
<td>Specifies the TimesTen ODBC Driver Manager.</td>
<td>None</td>
</tr>
<tr>
<td>Durability</td>
<td>Determines the durability of &quot;prepare-to-commit&quot; and commit records.</td>
<td>If K-safety is set to 2, the default value is 0. If K-safety is set to 1, the default value is 1.</td>
</tr>
<tr>
<td>LogDir</td>
<td>The directory where transaction log files are stored.</td>
<td>None</td>
</tr>
<tr>
<td>Preallocate</td>
<td>Specifies that disk space for the database should be preallocated when creating the database.</td>
<td>1 (True)</td>
</tr>
<tr>
<td>ReplicationApplyOrdering</td>
<td>Enables automatic parallel replication.</td>
<td>0 (Starts automatic parallel replication)</td>
</tr>
<tr>
<td>ReplicationParallelism</td>
<td>Specifies the number of tracks available for automatic parallel replication.</td>
<td>4</td>
</tr>
</tbody>
</table>
### Table 2–1  (Cont.) Data store attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Temporary</strong></td>
<td>Specifies that the database is not saved to disk.</td>
<td>0 (False)</td>
</tr>
</tbody>
</table>

### Table 2–2  First connection attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AutoCreate</strong></td>
<td>Specifies that the first connection creates the database if it does not exist.</td>
<td>1 (True)</td>
</tr>
<tr>
<td><strong>CkptFrequency</strong></td>
<td>Controls the frequency in seconds that TimesTen performs a background checkpoint.</td>
<td>0 (No predefined frequency)</td>
</tr>
<tr>
<td><strong>CkptLogVolume</strong></td>
<td>Controls the amount of data in megabytes that collects in the log between background checkpoints.</td>
<td>The value of the LogFileSize attribute</td>
</tr>
<tr>
<td><strong>CkptRate</strong></td>
<td>Controls the maximum rate at which data should be written to disk during a checkpoint operation.</td>
<td>0 (Unlimited rate)</td>
</tr>
<tr>
<td><strong>CkptReadThreads</strong></td>
<td>Controls the number of threads used to read a checkpoint file when loading the database into memory.</td>
<td>1</td>
</tr>
<tr>
<td><strong>Connections</strong></td>
<td>Indicates the upper bound on the number of user-specified concurrent connections to the database.</td>
<td>The lesser of 2000 or the number of semaphores specified in the SEMNS kernel parameter minus 155.</td>
</tr>
<tr>
<td><strong>EpochInterval</strong></td>
<td>Indicates the number of seconds between epochs.</td>
<td>If Durability = 1, the default is 0. (System does not generate periodic epochs) If Durability = 0, the default is 1.</td>
</tr>
<tr>
<td><strong>ForceConnect</strong></td>
<td>Specifies whether a connection is allowed to a failed database if it is not properly restored from the corresponding subscriber database.</td>
<td>0 (Connection disallowed)</td>
</tr>
<tr>
<td><strong>ForceDisconnectEnabled</strong></td>
<td>Enables forced disconnections through <code>ttAdmin -disconnect</code>.</td>
<td>0 (Disabled)</td>
</tr>
<tr>
<td><strong>LogAutoTruncate</strong></td>
<td>Determines whether the first connection to a database should proceed if TimesTen recovery encounters a defective log record.</td>
<td>1 (Continues after log is truncated)</td>
</tr>
<tr>
<td><strong>LogBufMB</strong></td>
<td>The size of the internal log buffer in MB.</td>
<td>64</td>
</tr>
<tr>
<td><strong>LogBufParallelism</strong></td>
<td>The number of log buffer strands.</td>
<td>4</td>
</tr>
</tbody>
</table>
Table 2–2  (Cont.) First connection attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>LogFileSize</td>
<td>The transaction log file size in MB.</td>
<td>The value of the LogBufMB attribute.</td>
</tr>
<tr>
<td>LogFlushMethod</td>
<td>Controls the method used by TimesTen to write and sync log data to transaction log files.</td>
<td>1 (Write data to transaction log files using buffered writes. Use explicit sync operations as needed to sync log data to disk)</td>
</tr>
<tr>
<td>LogPurge</td>
<td>Specifies that unneeded transaction log files are deleted during a checkpoint operation.</td>
<td>1 (True)</td>
</tr>
<tr>
<td>MemoryLock</td>
<td>enables applications that connect to a shared database to specify whether the real memory should be locked during database loading.</td>
<td>0 (Do not acquire a memory lock)</td>
</tr>
<tr>
<td>Overwrite</td>
<td>Specifies that the existing database should be overwritten with a new one when a connection is attempted.</td>
<td>0 (False)</td>
</tr>
<tr>
<td>PermSize</td>
<td>The size in MB for the permanent partition of the database.</td>
<td>32</td>
</tr>
<tr>
<td>RecoveryThreads</td>
<td>The number of threads used to rebuild indexes during recovery.</td>
<td>4</td>
</tr>
<tr>
<td>TempSize</td>
<td>The size in MB for the temporary partition of the database.</td>
<td>The default size as determined from the PermSize value</td>
</tr>
</tbody>
</table>

Table 2–3  General connection attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>CommitBufferSizeMax</td>
<td>Specifies the maximum size of the commit buffer in the transaction control block.</td>
<td>16 KB</td>
</tr>
<tr>
<td>ConnectionName</td>
<td>Specifies whether there is a symbolic name for the data source.</td>
<td>The process name.</td>
</tr>
<tr>
<td>CreateEpochAtCommit</td>
<td>Specifies if each commit generates an epoch.</td>
<td>0 (default) - TimesTen does not write the transaction log to disk on transaction commit.</td>
</tr>
<tr>
<td>DDLReplicationAction</td>
<td>Determines whether a table or sequence is included in an active standby pair replication scheme when it is created, which can only occur if the DDLReplicationLevel connection attribute is set to 2 or 3.</td>
<td>INCLUDE</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Default</td>
</tr>
<tr>
<td>----------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>DDLReplicationLevel</td>
<td>Enables replication of data definition language (DDL) statements in an active standby replication scheme.</td>
<td>2 (Replication of certain objects enabled)</td>
</tr>
<tr>
<td>Diagnostics</td>
<td>Specifies whether diagnostic messages are generated.</td>
<td>1 (Messages are generated)</td>
</tr>
<tr>
<td>DurableCommits</td>
<td>Specifies that commit operations should write log records to disk.</td>
<td>0 (Records not written to disk)</td>
</tr>
<tr>
<td>IncludeInCore</td>
<td>Specifies which parts of shared memory are included in a core dump.</td>
<td>0 (None)</td>
</tr>
<tr>
<td>Isolation</td>
<td>Specifies whether the isolation level is read committed or serializable.</td>
<td>1 (Read committed)</td>
</tr>
<tr>
<td>LockLevel</td>
<td>Specifies whether the connection should use row-level locking (value = 0) or database-level locking (value = 1).</td>
<td>0 (Row-level locking)</td>
</tr>
<tr>
<td>LockWait</td>
<td>Enables an application to configure the lock wait interval for the connection.</td>
<td>10 seconds</td>
</tr>
<tr>
<td>OptimizerHint</td>
<td>Sets optimizer hints at the connection level.</td>
<td>None</td>
</tr>
<tr>
<td>PermWarnThreshold</td>
<td>The threshold at which TimesTen returns a warning when the permanent partition of the database is low in memory.</td>
<td>90%</td>
</tr>
<tr>
<td>PrivateCommands</td>
<td>Determines if commands are shared between connections.</td>
<td>0 (On)</td>
</tr>
<tr>
<td>PWD</td>
<td>Specify the password that corresponds with the specified UID. When caching data from an Oracle database, PWD specifies the TimesTen password. You can specify the Oracle PWD in the connection string, if necessary.</td>
<td>None</td>
</tr>
<tr>
<td>PWDCrypt</td>
<td>The value of the encrypted user password.</td>
<td>None</td>
</tr>
<tr>
<td>QueryThreshold</td>
<td>Determines whether TimesTen returns a warning if a query times out before executing.</td>
<td>0 (No warning is returned)</td>
</tr>
<tr>
<td>ReplicationTrack</td>
<td>Assigns a connection to a replication track.</td>
<td>None</td>
</tr>
<tr>
<td>SQLQueryTimeout</td>
<td>Specifies the time limit in seconds within which the database should execute SQL statements.</td>
<td>0 (No timeout)</td>
</tr>
</tbody>
</table>
## Table 2–3  (Cont.) General connection attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>TempWarnThreshold</td>
<td>The threshold at which TimesTen returns a warning when the temporary partition of the database is low in memory.</td>
<td>90 (percent)</td>
</tr>
<tr>
<td>UID</td>
<td>Specify a user name that is defined on the TimesTen server. When caching data from an Oracle database, the UID must match the UID on the Oracle database that is being cached in TimesTen.</td>
<td>None</td>
</tr>
<tr>
<td>WaitForConnect</td>
<td>Specifies that the connection attempt should wait if an immediate connection is not possible.</td>
<td>1</td>
</tr>
</tbody>
</table>

## Table 2–4  NLS general connection attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConnectionCharacterSet</td>
<td>The character encoding for the connection, which can be different from the database character set.</td>
<td>US7ASCII</td>
</tr>
<tr>
<td>NLS_LENGTH_SEMANTICS</td>
<td>The default length semantics configuration.</td>
<td>BYTE</td>
</tr>
<tr>
<td>NLS_NCHAR_CONV_EXCP</td>
<td>Determines whether an error is reported when there is data loss during an implicit or explicit character type conversion between NCHAR/NVARCHAR data and CHAR/VARCHAR data.</td>
<td>0 (False)</td>
</tr>
<tr>
<td>NLS_SORT</td>
<td>The collating sequence to use for linguistic comparisons.</td>
<td>BINARY</td>
</tr>
</tbody>
</table>

## Table 2–5  PL/SQL first connection attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLSQL_MEMORY_ADDRESS</td>
<td>The virtual address at which the shared memory segment is loaded into each process that uses the TimesTen direct drivers.</td>
<td>Platform specific</td>
</tr>
<tr>
<td>PLSQL_MEMORY_SIZE</td>
<td>The size in megabytes of the shared memory segment used by PL/SQL.</td>
<td>128 MB</td>
</tr>
</tbody>
</table>

## Table 2–6  PL/SQL general connection attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLSCOPE_SETTINGS</td>
<td>Controls whether the PL/SQL compiler generates cross-reference information.</td>
<td>IDENTIFIERS: NONE</td>
</tr>
</tbody>
</table>
### Table 2–6  (Cont.) PL/SQL general connection attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLSQL_CCFLAGS</td>
<td>Controls conditional compilation of PL/SQL units.</td>
<td>NULL</td>
</tr>
<tr>
<td>PLSQL_CONN_MEM_LIMIT</td>
<td>Specifies the maximum amount of process heap memory in MB that PL/SQL can use for this connection.</td>
<td>100</td>
</tr>
<tr>
<td>PLSQL_OPTIMIZE_LEVEL</td>
<td>The optimization level that the PL/SQL compiler uses to compile PL/SQL library units.</td>
<td>2</td>
</tr>
<tr>
<td>PLSQL_SESSION_CACHED_CURSORS</td>
<td>The number of session cursors to cache.</td>
<td>50</td>
</tr>
<tr>
<td>PLSQL_TIMEOUT</td>
<td>The number of seconds a PL/SQL procedure can run before being automatically terminated.</td>
<td>30</td>
</tr>
</tbody>
</table>

### Table 2–7  TimesTen Cache first connection attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>CacheAWTMethod</td>
<td>Enables the AWT propagation method to be used on Oracle database tables.</td>
<td>1 (PL/SQL)</td>
</tr>
</tbody>
</table>

### Table 2–8  TimesTen Cache database attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>CacheAWTParallelism</td>
<td>Indicates the number of threads necessary to apply changes to the Oracle database.</td>
<td>1</td>
</tr>
</tbody>
</table>

### Table 2–9  TimesTen Cache general connection attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>DynamicLoadEnable</td>
<td>Enables or disables transparent load of data from an Oracle database to dynamic cache groups.</td>
<td>1 (Enables Dynamic cache group load)</td>
</tr>
<tr>
<td>DynamicLoadErrorMode</td>
<td>Determines if an error message is returned upon a transparent load failure.</td>
<td>0 (Errors are not returned)</td>
</tr>
<tr>
<td>OracleNetServiceName</td>
<td>The Oracle Service Name of the Oracle database instance from which data is to be loaded into a TimesTen database. This attribute is only used by the cache agent. Set the OracleNetServiceName attribute to the Oracle Service Name.</td>
<td>None</td>
</tr>
<tr>
<td>OraclePWD</td>
<td>Identifies the password for the Oracle database that is being cached in TimesTen.</td>
<td>None</td>
</tr>
<tr>
<td>PassThrough</td>
<td>Specifies which SQL statements are executed locally in TimesTen and which SQL statements are passed through to the Oracle database for execution.</td>
<td>0</td>
</tr>
</tbody>
</table>
### Table 2–9 (Cont.) TimesTen Cache general connection attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>RACCallback</td>
<td>Specifies whether to enable or disable the installation of Application Failover (TAF) and Fast Application Notification (FAN) callbacks.</td>
<td>1 (Install callbacks)</td>
</tr>
</tbody>
</table>

### Table 2–10 TimesTen Client connection attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP_Port</td>
<td>The port number on which the TimesTen server is listening.</td>
<td>None</td>
</tr>
<tr>
<td>TCP_Port2</td>
<td>For TimesTen Classic, the port number on which the TimesTen server should listen if an automatic failover occurs.</td>
<td>None</td>
</tr>
<tr>
<td>TTC_ConnectTimeout</td>
<td>Number of seconds for the client to wait for a connect or disconnect call.</td>
<td>None</td>
</tr>
<tr>
<td>TTC_FailoverPortRange</td>
<td>A range for the failover port numbers.</td>
<td>None</td>
</tr>
<tr>
<td>TTC_NoReconnectOnFailover</td>
<td>Specifies if TimesTen performs an automatic reconnect after all other failover procedures are completed.</td>
<td>1</td>
</tr>
<tr>
<td>TTC_Random_Selection</td>
<td>For TimesTen Scaleout, specifies that the client randomly selects an alternate server from the list provided or selects the server according to TTC_ServerN settings.</td>
<td>0</td>
</tr>
<tr>
<td>TTC_REDIRECT</td>
<td>For TimesTen Scaleout, client/server connections, defines how a client is redirected.</td>
<td>1 (connect to any available server)</td>
</tr>
<tr>
<td>TTC_Redirect_Limit</td>
<td>For TimesTen Scaleout, limits the number of times the client is redirected.</td>
<td>1</td>
</tr>
<tr>
<td>TTC_RollbackRequiredOnFailover</td>
<td>For client/server connections, defines the maximum number of times a client is redirected.</td>
<td>1</td>
</tr>
<tr>
<td>TTC_Server or TTC_Server1</td>
<td>Name of the computer where the TimesTen Server is running or a logical TimesTen server name.</td>
<td>None</td>
</tr>
<tr>
<td>TTC_Server2, TTC_ServerN</td>
<td>If an automatic failover occurs, the name of the system where the TimesTen Server should be running or a logical TimesTen server name. Values of N&gt;2 are supported for TimesTen Scaleout.</td>
<td>None</td>
</tr>
<tr>
<td>TTC_Server_DSN</td>
<td>Server DSN corresponding to the TimesTen database.</td>
<td>None</td>
</tr>
<tr>
<td>TTC_Server_DSN2</td>
<td>For TimesTen Classic, server DSN corresponding to the TimesTen database, if an automatic failover occurs.</td>
<td>None</td>
</tr>
<tr>
<td>TTC_TCP_KEEPALIVE_INTVL_MS</td>
<td>The time interval (in milliseconds) between subsequential probes.</td>
<td>1000</td>
</tr>
<tr>
<td>TTC_TCP_KEEPALIVE_PROBES</td>
<td>The number of unacknowledged probes to send before considering the connection as failed and notifying the client.</td>
<td>2</td>
</tr>
</tbody>
</table>
### Table 2–10  (Cont.) TimesTen Client connection attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTC_TCP_KEEPALIVE_TIME_MS</td>
<td>The duration time (in milliseconds) between the last data packet sent and the first probe.</td>
<td>1000</td>
</tr>
<tr>
<td>TTC_Timeout</td>
<td>Timeout period, in seconds, for completion of a TimesTen client/server operation.</td>
<td>60</td>
</tr>
</tbody>
</table>

### Table 2–11  TimesTen Server connection attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaxConnsPerServer</td>
<td>The maximum number of concurrent connections a child TimesTen server process can handle.</td>
<td>1</td>
</tr>
<tr>
<td>ServersPerDSN</td>
<td>The desired number of TimesTen server processes for the DSN.</td>
<td>1</td>
</tr>
<tr>
<td>ServerStackSize</td>
<td>The size in KB of the thread stack for each connection.</td>
<td>256</td>
</tr>
</tbody>
</table>
Data store attributes

Data store attributes are set at database creation time. The data store attributes are listed in Table 2-1, "Data store attributes" and described in detail in this section. These attributes can be assigned values only during database creation by the instance administrator.
Data Source Name

The data source name (DSN) uniquely identifies the attributes to a connection. It serves two purposes:

- As a unique identifier to the ODBC driver manager (if one is present), allowing it to associate a Data Store Name with a specific ODBC driver.
- As one of potentially many name aliases to a single physical database where the name alias has unique attributes associated with it.

The database attributes can apply to either the data source name (connection to a database) or the Data Store Path Name (database).

On Windows, the data source name and all configuration information associated with the data source (including the database path name) are stored in the system registry. The ODBC driver manager and TimesTen use this information.

Required privilege

Only the instance administrator can change the value of this attribute.

Setting

Set Data Source Name as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems and Linux systems odbc.ini file</td>
<td>DSN</td>
<td>A name that describes the DSN.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>Data Source Name field</td>
<td>A name that describes the DSN.</td>
</tr>
</tbody>
</table>
DataStore

The database path name uniquely identifies the physical database. It is the full path name of the database and the file name prefix, for example: C:\data\AdminData. This name is not a file name. The actual database file names have suffixes, such as .ds0 and .log0, for example C:\data\AdminData.ds0 and C:\data\AdminData.log0.

Note: You are required to specify the database path and name at database creation time. It cannot be altered after the database has been created.

Required privilege

Only the instance administrator can change the value of this attribute.

Setting

Set DataStore as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems and Linux systems odbc.ini file</td>
<td>DataStore</td>
<td>Full path to the physical database that the data source name references.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>Data Store Path + Name field</td>
<td>Full path to the physical database that the data source name references.</td>
</tr>
</tbody>
</table>
DatabaseCharacterSet

The database character set determines the character set in which data is stored.

Note: You are required to specify the database character set at database creation time only. It cannot be altered after the database has been created. If you do not specify a value for this attribute when creating a database, TimesTen returns error message 12701.

Generally, your database character set should be chosen based on the data requirements. For example: Do you have data in Unicode or is your data in Japanese on UNIX and Linux systems (EUC) or Windows (SJIS)?

You should choose a connection character set that matches your terminal settings or data source. See "ConnectionCharacterSet" on page 2-76.

When the database and connection character sets differ, TimesTen performs the data conversion internally based on the connection character set. If the connection and database character sets are the same, TimesTen does not need to convert or interpret the data set. Best performance occurs when connection and database character sets match, since no conversion is required.

To use this attribute you must specify a supported character set. For a list of supported character set names, see "Supported character sets" below.

There are several things to consider when choosing a character set for your database. For a discussion about these considerations, see 'Choosing a database character set' in Oracle TimesTen In-Memory Database Operations Guide.

Required privilege

Only the instance administrator can change the value of this attribute.

Setting

Set DatabaseCharacterSet name as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>DatabaseCharacterSet</td>
<td>Specify the preferred character set.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>Database Character Set list</td>
<td>Select the preferred character set from the list provided in the ODBC Data Source Administrator.</td>
</tr>
</tbody>
</table>

Supported character sets

The tables in this section describe the character sets supported in TimesTen.

Asian character sets

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JA16EUC</td>
<td>EUC 24-bit Japanese</td>
</tr>
</tbody>
</table>
### European character sets

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLT8CP921</td>
<td>Latvian Standard LV58-92 (1) Windows/UNIX/Linux 8-bit Baltic</td>
</tr>
<tr>
<td>BLT8ISO8859P13</td>
<td>ISO 8859-13 Baltic</td>
</tr>
<tr>
<td>BLT8MSWIN1257</td>
<td>Microsoft Windows Code Page 1257 8-bit Baltic</td>
</tr>
<tr>
<td>BLT8PC775</td>
<td>IBM-PC Code Page 775 8-bit Baltic</td>
</tr>
<tr>
<td>CEL8ISO8859P14</td>
<td>ISO 8859-13 Celtic</td>
</tr>
<tr>
<td>CL8ISO8859P5</td>
<td>ISO 8859-5 Latin/Cyrillic</td>
</tr>
<tr>
<td>CL8KO18R</td>
<td>RELCOM Internet Standard 8-bit Latin/Cyrillic</td>
</tr>
<tr>
<td>CL8KO18U</td>
<td>KOI8 Ukrainian Cyrillic</td>
</tr>
<tr>
<td>CL8MSWIN1251</td>
<td>Microsoft Windows Code Page 1251 8-bit Latin/Cyrillic</td>
</tr>
<tr>
<td>EE8ISO8859P2</td>
<td>ISO 8859-2 East European</td>
</tr>
<tr>
<td>EL8ISO8859P7</td>
<td>ISO 8859-7 Latin/Greek</td>
</tr>
<tr>
<td>ET8MSWIN923</td>
<td>Microsoft Windows Code Page 923 8-bit Estonian</td>
</tr>
<tr>
<td>EE8MSWIN1250</td>
<td>Microsoft Windows Code Page 1250 8-bit East European</td>
</tr>
<tr>
<td>EL8MSWIN1253</td>
<td>Microsoft Windows Code Page 1253 8-bit Latin/Greek</td>
</tr>
<tr>
<td>EL8PC737</td>
<td>IBM-PC Code Page 737 8-bit Greek/Latin</td>
</tr>
<tr>
<td>EE8PC852</td>
<td>IBM-PC Code Page 852 8-bit East European</td>
</tr>
<tr>
<td>LT8MSWIN921</td>
<td>Microsoft Windows Code Page 921 8-bit Lithuanian</td>
</tr>
<tr>
<td>NE8ISO8859P10</td>
<td>ISO 8859-10 North European</td>
</tr>
<tr>
<td>NEEE8ISO8859P4</td>
<td>ISO 8859-4 North and North-East European</td>
</tr>
</tbody>
</table>
### Data store attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RU8PC866</td>
<td>IBM-PC Code Page 866 8-bit Latin/Cyrillic</td>
</tr>
<tr>
<td>U8ISO8859P3</td>
<td>ISO 8859-3 South European</td>
</tr>
<tr>
<td>US7ASCII</td>
<td>ASCII 7-bit American</td>
</tr>
<tr>
<td>US8PC437</td>
<td>IBM-PC Code Page 437 8-bit American</td>
</tr>
<tr>
<td>U8ISO8859P1</td>
<td>ISO 8859-1 West European</td>
</tr>
<tr>
<td>U8ISO8859P15</td>
<td>ISO 8859-15 West European</td>
</tr>
<tr>
<td>U8MSWIN1252</td>
<td>Microsoft Windows Code Page 1252 8-bit West European</td>
</tr>
<tr>
<td>U8PC850</td>
<td>IBM-PC Code Page 850 8-bit West European</td>
</tr>
<tr>
<td>U8PC858</td>
<td>IBM-PC Code Page 858 8-bit West European</td>
</tr>
</tbody>
</table>

#### Middle Eastern character sets

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR8ADOS720</td>
<td>Arabic MS-DOS 720 Server 8-bit Latin/Arabic</td>
</tr>
<tr>
<td>AR8ASMO8X</td>
<td>ASMO Extended 708 8-bit Latin/Arabic</td>
</tr>
<tr>
<td>AR8ISO8859P6</td>
<td>ISO 8859-6 Latin/Arabic</td>
</tr>
<tr>
<td>AR8MSWIN1256</td>
<td>Microsoft Windows Code Page 1256 8-bit Latin/Arabic</td>
</tr>
<tr>
<td>AZ8ISO8859P9E</td>
<td>ISO 8859-9 Latin Azerbaijani</td>
</tr>
<tr>
<td>IW8ISO8859P8</td>
<td>ISO 8859-8 Latin/Hebrew</td>
</tr>
<tr>
<td>IW8MSWIN1255</td>
<td>Microsoft Windows Code Page 1255 8-bit Latin/Hebrew</td>
</tr>
<tr>
<td>TR8MSWIN1254</td>
<td>Microsoft Windows Code Page 1254 8-bit Turkish</td>
</tr>
<tr>
<td>TR8PC857</td>
<td>IBM-PC Code Page 857 8-bit Turkish</td>
</tr>
<tr>
<td>WE8ISO8859P9</td>
<td>ISO 8859-9 West European &amp; Turkish</td>
</tr>
</tbody>
</table>

#### Universal character sets

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AL16UTF16</td>
<td>Unicode 4.0 UTF-16 Universal character set. This is the implicit TimesTen national character set.</td>
</tr>
<tr>
<td>AL32UTF8</td>
<td>Unicode 4.0 UTF-8 Universal character set</td>
</tr>
<tr>
<td>UTF8</td>
<td>Unicode 3.0 UTF-8 Universal character set, CESU-8 compliant</td>
</tr>
</tbody>
</table>
Description

Optionally, set this attribute to help you identify the Data Source Name (DSN) and its attributes.

Required privilege

Only the instance administrator can change the value of this attribute.

Setting

Set Description as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>Description</td>
<td>Text description of the Data Source Name. This attribute is optional.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>Description field</td>
<td>Text description of the Data Source Name. This attribute is optional.</td>
</tr>
</tbody>
</table>
**Driver**

The **Driver** attribute specifies the name of the TimesTen ODBC Driver.

For example, on Windows systems the value can be **TimesTen Data Manager 18.1** or **TimesTen Client 18.1**.

On UNIX and Linux systems, the value of the **Driver** attribute is the path name of the TimesTen ODBC Driver shared library file, `timesten_home/install/lib/libtten.so`.

For more information, see "Defining a Data Manager DSN" in *Oracle TimesTen In-Memory Database Operations Guide*.


**Required privilege**

Only the instance administrator can change the value of this attribute.

**Setting**

Set **Driver** as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems <code>odbc.ini</code> file</td>
<td><strong>Driver</strong></td>
<td>Specifies the path name for the TimesTen ODBC Driver shared library file, if using a Driver Manager.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>Select a driver from the <strong>Create New Data Source</strong> dialog.</td>
<td>Specifies the Client or Data Manager driver for TimesTen and the release.</td>
</tr>
</tbody>
</table>
Durability

For TimesTen TimesTen Scaleout, the setting of this attribute determines the durability of "prepare-to-commit" and commit records.

If K-safety is set to 2, the default value is 0.

If K-safety is set to 1, the default value is 1.

Durability=0 is not supported with K-safety set to 1.

For details about setting up K-safety, see "Creating a grid" in the Oracle TimesTen In-Memory Database Scaleout User’s Guide.

Required privilege

No privilege is required to change the value of this attribute.

Setting

Set Durability as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>Durability</td>
<td>0 - TimesTen does not write the transaction prepare-to-commit and commit records to disk on transaction commit. 1 - TimesTen writes the prepare-to-commit record durably and the commit record non-durably.</td>
</tr>
</tbody>
</table>

Windows ODBC Data Source Administrator | Not available. |

See also

CreateEpochAtCommit
GridLogRecoveryThreshold

The `GridLogRecoveryThreshold` attribute determines the number of log files that are prevented from purging before declaring another replica set element is declared unrecoverable. Default value is 0, which means TimesTen saves all the log files. When the value is set to some thing non-zero, log files will keep getting accumulated until the threshold value is reached. Once the threshold is reached, the replica set element for which the log files are being saved is declared as unrecoverable.

**Required privilege**

Only the instance administrator can change the value of this attribute.

**Setting**

Set `GridLogRecoveryThreshold` as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems <code>odbc.ini</code> file</td>
<td><code>GridLogRecoveryThreshold</code></td>
<td>A positive integer that specifies the number of log files to preserve before declaring a replica set element is unrecoverable. The default is 0, meaning all files are saved.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td><strong>Grid Log Recovery Threshold</strong> field</td>
<td>A positive integer that specifies the number of log files to preserve before declaring a replica set element is unrecoverable. The default is 0, meaning all files are saved.</td>
</tr>
</tbody>
</table>
LogDir

The LogDir attribute specifies the directory where database logs reside. Specifying this attribute enables you to place the transaction log files on a different I/O path from the database checkpoint files. Placing the transaction log files and checkpoint files on different disks can improve system performance.

Required privilege

Only the instance administrator can change the value of this attribute.

Setting

Set LogDir as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>LogDir</td>
<td>Specifies the directory where transaction log files reside.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>Transaction Log Directory field</td>
<td>Specifies the directory where transaction log files reside.</td>
</tr>
</tbody>
</table>
Preallocate

The `Preallocate` attribute determines whether TimesTen preallocates file system space for the database when the database is created. Setting this attribute ensures that there is sufficient space for the database when the database is saved to the file system.

Using `Preallocate=1` in combination with `ttRestore` or `ttRepAdmin -duplicate` and a value of `PermSize` that does not match the value of `PermSize` of the original database may result in two checkpoint files with different sizes. This has not been shown to have negative effects. However, the issue can be avoided completely either by using the same `PermSize` as the original database or by setting `Preallocate=0`.

When a duplicate operation is carried out, the duplicated store has behavior consistent with a Preallocate setting of 0, even if it is set to 1 on the original or duplicated database. The behavior is indicated by the size of the checkpoint files, which is the sum of the size of the data and size of the database header.

The checkpoint files are subsequently allowed to grow to the same size as checkpoint files on the master database (`PermSize + database header`), but the space is not preallocated. The checkpoint files increase in size as data is added.

Required privilege

Only the instance administrator can change the value of this attribute.

Setting

Set `Preallocate` as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td><code>Preallocate</code></td>
<td>0 - Does not preallocate file system space for database when creating the database. 1 (Default) - Preallocates file system space for the database.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td><code>Preallocate</code> check box</td>
<td>unchecked - Does not preallocate file system space for database when creating the database. checked (default) - Preallocates file system space for the database.</td>
</tr>
</tbody>
</table>

**Note:** reallocating disk space for a large database is very time consuming.
ReplicationApplyOrdering

Enables parallel replication when used with the ReplicationParallelism attribute. With parallel replication, multiple transmitters on the master send to multiple receivers on the subscriber.

- Automatic parallel replication: Parallel replication over multiple threads that automatically enforces transactional dependencies and all changes applied in commit order. This is the default.

- Automatic parallel replication with disabled commit dependencies: Parallel replication over multiple threads that automatically enforces transactional dependencies, but does not enforce transactions committed in the same order on the subscriber database as on the active database. You can also increase replication throughput by applying transactions to specific tracks.

For more details on configuring parallel replication, see "Configuring parallel replication" in the Oracle TimesTen In-Memory Database Replication Guide.

This attribute also sets parallel propagation for AWT cache groups. By default, this attribute enables parallel propagation of updates to the Oracle database. To learn more about parallel AWT caching, see "Configuring parallel propagation to Oracle Database tables" in the Oracle TimesTen Application-Tier Database Cache User’s Guide.

Required privilege

Only the instance administrator can change the value of this attribute.

Setting

Set ReplicationApplyOrdering as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>ReplicationApplyOrdering</td>
<td>0 - Specifies automatic parallel replication.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Automatic parallel replication is available for both classic and active</td>
</tr>
<tr>
<td></td>
<td></td>
<td>standby pair replication schemes. (default)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 - Specifies automatic parallel replication with disabled commit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>dependencies.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>Replication Apply Ordering</td>
<td>0 - Specifies automatic parallel replication.</td>
</tr>
<tr>
<td></td>
<td>pulldown list</td>
<td>Automatic parallel replication is available for both classic and active</td>
</tr>
<tr>
<td></td>
<td></td>
<td>standby pair replication schemes. (default)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 - Specifies automatic parallel replication with disabled commit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>dependencies.</td>
</tr>
</tbody>
</table>
Restrictions

Restrictions when using automatic parallel replication with disabled commit dependencies:

- The replication scheme must be an active standby pair that uses asynchronous replication. Classic replication schemes are not supported.
- The replication scheme cannot contain cache groups.
- This is only supported for TimesTen Release 11.2.2.8 and greater for both the active and standby masters. Both the active and standby masters must have commit dependencies disabled.
- XLA is not supported.

All data stores in the replication scheme must use the same setting.
ReplicationParallelism

This attribute specifies the number of tracks, or the number of transmitter/receiver pairs, used for automatic parallel replication.

If the value is greater than 1, the LogBufParallelism first connection attribute must be an integral multiple of ReplicationParallelism.

To configure automatic parallel replication, set this attribute to a value from 2 to 32, indicating the number of transmitter/receiver pairs. The default value is 4.

To learn more about automatic parallel replication, see "Configuring parallel replication" in the Oracle TimesTen In-Memory Database Replication Guide.

Required privilege

Only the instance administrator can change the value of this attribute.

Setting

Set ReplicationParallelism as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>ReplicationParallelism</td>
<td>( n ) - A value between 1 and 32, indicating the number of tracks to replicate in parallel. The default is 4.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>Replication Parallelism field</td>
<td>( n ) - A value between 1 and 32, indicating the number of tracks to replicate in parallel. The default is 4.</td>
</tr>
</tbody>
</table>

Restrictions

Restrictions and things to consider when specifying parallel replication include:

- When parallel replication is enabled, the Description column of the ttLogHolds built-in procedure displays one row per track per subscriber node.
- We recommend setting the value of this attribute to a value no greater than half the value of the LogBufParallelism attribute. If you specify more replication tracks than log buffer threads, some replication tracks can remain empty.
Temporary

Set this attribute to create a temporary database. Temporary databases are not saved to the file system. They may, however, be shared and therefore require a data store path name. A temporary database is deleted when the last connection is closed. You cannot assign the Temporary data store attribute to an existing permanent database.

**Note:** You cannot back up or replicate a temporary database.

**Required privilege**

Only the instance administrator can change the value of this attribute.

**Setting**

Set Temporary as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>Temporary</td>
<td>0 (default) - Creates a permanent database. 1 - Creates a temporary database.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>Temporary check box</td>
<td>unchecked (default) - Creates a permanent database. checked - Creates temporary a database.</td>
</tr>
</tbody>
</table>
First connection attributes

TimesTen sets first connection attributes when a database created by the instance administrator is loaded into memory and persist for the first connection and all subsequent connections until the last connection to this database is closed.

You can modify first connection attributes only when the TimesTen database is unloaded. Then the instance administrator reconnects with new values for the first connection attributes.

First connection attributes are listed in Table 2–2, "First connection attributes" and described in detail in this section.

If you try to connect to the database using attributes that are different from the first connection attribute settings, the new connection can be rejected or the attribute value can be ignored. However, for example, if existing connections have a LogFileSize of one size and a new connection specifies a LogFileSize of another size, TimesTen ignores the new value and returns a warning.

---

**Note:** Only the instance administrator can change a first connection attribute to a value other than the one currently in effect. To change the value of a first connection attribute, you must first shut down the database and then connect with ADMIN privileges. (No privileges are required to change AutoCreate and ForceConnect.)
AutoCreate

If you connect to a database that has the `AutoCreate` attribute set and the database does not exist yet, the database is created automatically if you supplied a valid existing path. With `AutoCreate` set, TimesTen creates the database, but not the path to the database. If you attempt to connect to a database that does not exist and the `AutoCreate` attribute is not set, the connection fails.

Also see “Overwrite” on page 2-47.

Required privilege

No privilege is required to change the value of this attribute.

Setting

Set `AutoCreate` as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
</table>
| C or Java programs or UNIX and Linux systems `odbc.ini` file | `AutoCreate` | 0 - Does not create new database if database does not exist.  
1 (default) - Creates a new database if the specified database does not exist. |
| Windows ODBC Data Source Administrator | `AutoCreate` check box | `unchecked` - Does not create new database if database does not exist.  
`checked` (default) - Creates a new database if database does not exist. |
**CkptFrequency**

Controls the frequency in seconds that TimesTen performs a background checkpoint. The counter used for the checkpoint condition is reset at the beginning of each checkpoint.

If both `CkptFrequency` and `CkptLogVolume` attributes have a value greater than 0, a checkpoint is performed when either of the two conditions becomes true. The values set by the `ttCkptConfig` built-in procedure replace the values set by these attributes.

In the case that your application attempts to perform a checkpoint operation while a background checkpoint is in process, TimesTen waits until the background checkpoint finishes and then executes the application’s checkpoint. To turn off background checkpointing, set `CkptFrequency`=0 and `CkptLogVolume`=0.

When using TimesTen Scaleout, if `Durability` = 0, set `EpochInterval` to a value less than the value of the `CkptFrequency` when `Durability`=0 to guarantee at least 1 epoch per interval.

The value of this attribute is "sticky" as it persists across database loads and unloads unless it is explicitly changed. The default value is only used during database creation. Subsequent first connections default to using the existing value stored in the database.

When the value of this attribute is more than 0, if a checkpoint fails, TimesTen attempts a checkpoint once every 30 seconds. If a checkpoint failure occurs due to a lack of file system space, we recommend that you attempt a manual checkpoint as soon as space is available. Once any successful checkpoint occurs, background checkpointing reverts to the configured schedule.

**Required privilege**

Only the instance administrator can change the value of this attribute.

**Setting**

Set `CkptFrequency` as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td><code>CkptFrequency</code></td>
<td>Enter a value in seconds for the frequency at which TimesTen should perform a background checkpoint. The default is 0. If you do not specify this attribute with a value, TimesTen does not perform background checkpoints. For an existing database, TimesTen uses the stored value.</td>
</tr>
<tr>
<td>Where to set the attribute</td>
<td>How the attribute is represented</td>
<td>Setting</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>----------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>Ckpt Frequency (secs) field</td>
<td>Enter a value in seconds for the frequency at which TimesTen should perform a background checkpoint. The default is 0. If you do not specify this attribute with a value, TimesTen does not perform background checkpoints. For an existing database, TimesTen uses the stored value.</td>
</tr>
</tbody>
</table>
CkptLogVolume

Controls the amount of data in megabytes that collects in the log between background checkpoints. The counter used for the checkpoint condition is reset at the beginning of each checkpoint.

If both CkptFrequency and CkptLogVolume attributes have a value greater than 0, a checkpoint is performed when either of the two conditions becomes true. The values set by the ttCkptConfig built-in procedure replace the values set by these attributes.

In the case that your application attempts to perform a checkpoint operation while a background checkpoint is in process, TimesTen waits until the background checkpoint finishes and then executes the application’s checkpoint. To turn off background checkpointing, set CkptFrequency=0 and CkptLogVolume=0.

The value of this attribute is "sticky" as it persists across database loads and unloads unless it is explicitly changed. The default value is only used during database creation. Subsequent first connections default to using the existing value stored in the database.

When the value of this attribute is more than 0 and CkptFrequency=0, if a checkpoint fails, TimesTen attempts a checkpoint every 30 seconds. If a checkpoint failure occurs due to a lack of file system space, we recommend that you attempt a manual checkpoint as soon as space is available. Once any successful checkpoint occurs, background checkpointing reverts to the configured schedule.

Required privilege

Only the instance administrator can change the value of this attribute.

Setting

Set CkptLogVolume as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>CkptLogVolume</td>
<td>Specify the amount of data in megabytes that can accumulate in the transaction log file between background checkpoints. The default is the value supplied for the LogFileSize attribute. For an existing database, TimesTen uses the stored value. If the attribute is specified, but you do not supply a value, TimesTen uses the value supplied for the LogFileSize attribute.</td>
</tr>
</tbody>
</table>
### First Connection Attributes

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td><strong>Ckpt LogVolume</strong> field</td>
<td>Specify the amount of data in MBs that can accumulate in the transaction log file between background checkpoints. The default is 0. If you do not specify this attribute, TimesTen uses the default value (0) for database creation. For an existing database, TimesTen will use the stored value. If the attribute is specified, but you do not supply a value, TimesTen uses the default value (0).</td>
</tr>
</tbody>
</table>


CkptRate

Controls the maximum rate at which data should be written to disk during a checkpoint operation. This can be useful when the writing of checkpoints to disk interferes with other applications.

All background checkpoints and by checkpoints initiated by the `ttCkpt` and `ttCkptBlocking` built-in procedures use the rate specified by this connection attribute. Foreground checkpoints (checkpoints taken during first connect and last disconnect) do not use this rate. The rate is specified in MB per second.

A value of 0 disables rate limitation. This is the default. The value can also be specified using the `ttCkptConfig` built-in procedure. The value set by the `ttCkptConfig` built-in procedure replaces the value set by this attribute.

The value of this attribute is "sticky" as it persists across database loads and unloads unless it is explicitly changed. The default value is only used during database creation. Subsequent first connections default to using the existing value stored in the database. If left unspecified (or empty in the Windows ODBC Data Source Administrator), TimesTen uses the stored setting. To turn the attribute off, you must explicitly specify a value of 0. For existing databases that are migrated to this release, the value is initialized to 0. To use the current or default value, the attribute value should be left unspecified.

For more details about the benefits of and issues when using CkptRate, see "Setting the checkpoint rate" in Oracle TimesTen In-Memory Database Operations Guide.

Required privilege

Only the instance administrator can change the value of this attribute.

Setting

Set CkptRate as follows:
<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>CkptRate</td>
<td>Specify the maximum rate in MB per second at which a checkpoint should be written to disk. A value of 0 indicates that the rate should not be limited. This is the default. If you do not specify this attribute, TimesTen uses the default value (0) for database creation. TimesTen uses the stored value for existing databases. If the attribute is specified, but you do not supply a value, the value of 0 is used. Specifying a value of -1 is equivalent to omitting this attribute. If you specify a value of -1, the default value (0) is used for database creation, otherwise the stored value is used.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>CkptRate field</td>
<td>Specify the maximum rate in MB per second at which a checkpoint should be written to disk. A value of 0 indicates that the rate should not be limited. This is the default. If you do not specify this attribute, TimesTen uses the default value (0) for database creation. TimesTen uses the stored value for existing databases. If the attribute is specified, but you do not supply a value, TimesTen uses the default value (0).</td>
</tr>
</tbody>
</table>
CkptReadThreads

Determines the number of threads used to read the checkpoint file when loading the database into memory, such as in first connection or recovery operations.

When the CkptReadThreads attribute is set to 1, TimesTen reads checkpoint files serially. When the CkptReadThreads attribute is set to a value greater than 1, TimesTen uses the specified number of threads to read checkpoint files concurrently (in parallel). When the CkptReadThreads attribute is set to 0 or unspecified, the previously specified value is used.

Required privilege

Only the instance administrator can change the value of this attribute.

Setting

Set CkptReadThreads as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>CkptReadThreads</td>
<td>( n ) - The number of threads to use when reading the checkpoint files during the loading of the database into memory. Takes an integer value of 0 or greater (maximum = ( 2^{31} - 1 )). Default is 1.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>Checkpoint Read Threads field</td>
<td>( n ) - The number of threads to use when reading the checkpoint files during the loading of the database into memory. Takes an integer value of 0 or greater (maximum = ( 2^{31} - 1 )). Default is 1.</td>
</tr>
</tbody>
</table>

Notes

For a progress report on a recovery process, see the rebuild messages in the support log.

Set the number of threads low enough to leave sufficient resources on the TimesTen server for other services/processes.
Connections

Indicates the upper bound on the number of user-specified concurrent connections to the database. TimesTen allocates one semaphore for each expected connection. If the number of connections exceeds the value of this attribute, TimesTen returns an error.

The number of current connections to a database can be determined by viewing the output from the ttStatus utility.

As a guideline, set this value to the maximum number of expected application connections plus ten percent.

If you receive an error indicating that the number of connections exceeds the value of this attribute, increase the value until you no longer receive this error.

There is both a fixed and per connection overhead allocated from the PL/SQL segment, even if you do not use PL/SQL. For details, see "PLSQL_MEMORY_SIZE" on page 2-86.

Note: The kernel must be configured with enough semaphores to handle all active databases. For details on setting semaphores for your system, see "Operating system prerequisites" in Oracle TimesTen In-Memory Database Installation, Migration, and Upgrade Guide or "Operating system prerequisites" in Oracle TimesTen In-Memory Database Scaleout User's Guide.

Required privilege

Only the instance administrator can change the value of this attribute.

Setting

Set Connections as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>Connections</td>
<td>The default value is the lesser of 2000 or the number of semaphores specified in the SEMMS kernel parameter minus 155. A legal value is an integer from 1 through 32000. The value represents the maximum number of connections.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>Connections field</td>
<td>The default value is 2000. A legal value is an integer from 1 through 32000. The value represents the maximum number of connections.</td>
</tr>
</tbody>
</table>
EpochInterval

For TimesTen TimesTen Scaleout, use this attribute to set the number of seconds between epochs.

If **Durability** = 1, the default for this attribute is 0. (System does not generate periodic epochs)

If **Durability** = 0, the default for this attribute is 1. You must set this attribute to a value less than the value of the `CkptFrequency` when `Durability=0` to guarantee at least 1 epoch per interval.

If this attribute is set to 0, the system does not generate periodic epochs. An application can generate epochs at custom intervals by calling the `ttEpochCreate` built-in procedure each time the application wants to create an epoch.

If an element is down, an epoch interval can be skipped.

As long as one element in each replica set is up, the system never skips more than `K*EpochInterval` seconds between epochs.

**Required privilege**

No privilege is required to change the value of this attribute.

**Setting**

Set **EpochInterval** as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
</table>
| C or Java programs or UNIX and Linux systems `odbc.ini` file | `EpochInterval` | 0 - TimesTen does not generate periodic epochs.  
`n > 0` - An integer that indicates the number of seconds between epoch writes. |
| Windows ODBC Data Source Administrator | Not available. | |

**See also**

`CreateEpochAtCommit`
**ForceConnect**

When return receipt replication is used with the NONDURABLE TRANSMIT option, a failed master database is allowed to recover only by restoring its state from a subscriber database using the -duplicate option of the ttRepAdmin utility. In other words, the failed database cannot just come up and have replication bring it up to date because it may lose some transactions that were transmitted to the subscriber but not durably committed locally. The ForceConnect connection attribute overrides this restriction.

The ttConfiguration built-in procedure does not return the value of the ForceConnect attribute.

**Required privilege**

No privilege is required to change the value of this attribute.

**Setting**

Set ForceConnect as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>ForceConnect</td>
<td>0 (default) - Do not allow connection to failed database if it is not properly restored from the corresponding subscriber database. 1 - Allow connection to a failed database even if it is not properly restored from the corresponding subscriber database.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>ForceConnect check box</td>
<td>unchecked (default) - Do not allow connection to failed database if it is not properly restored from the corresponding subscriber database. checked - Allow connection to a failed database even if it is not properly restored from the corresponding subscriber database.</td>
</tr>
</tbody>
</table>
**ForceDisconnectEnabled**

Allows you to force disconnection from the database using `ttAdmin -disconnect`. See "Force disconnect" on page 5-16.

The `ttConfiguration` built-in procedure does not return the value of the `ForceDisconnectEnabled` attribute.

---

**Notes:**

- By default, the forced disconnect feature is disabled. Existing direct-connect applications may find it undesirable for TimesTen to spawn the thread that is required to implement this functionality.
- Users should not specify different values of this attribute for different database connections. If the force disconnect feature is desired, add `ForceDisconnectEnabled=1` to the DSN definition in the `sys.odbc.ini` file.

---

**Required privilege**

No privilege is required to change the value of this attribute.

**Setting**

Set `ForceDisconnectEnabled` as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td><code>ForceDisconnectEnabled</code></td>
<td>0 (default) - Do not allow forced disconnections. 1 - Allow forced disconnections.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>Not supported.</td>
<td></td>
</tr>
</tbody>
</table>
LogAutoTruncate

Determines whether the first connection to the database should proceed if TimesTen recovery encounters a defective log record.

Required privilege

Only the instance administrator can change the value of this attribute.

Setting

Set LogAutoTruncate as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>LogAutoTruncate</td>
<td>0 - If a defective log record is encountered, terminate recovery and return an error to the connecting application. Checkpoint and transaction log files remain unmodified. 1 (default) - If a defective log record is encountered, truncate the log at the defective record’s location and continue with recovery. The original transaction log files are moved to a directory called savedLogFiles, which is created as a subdirectory of the log directory. The transaction log files are saved for diagnostic purposes.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>LogAutoTruncate box</td>
<td>unchecked - If a defective log record is encountered, terminate recovery and return an error to the connecting application. Checkpoint and transaction log files remain unmodified. checked (default) - If a defective log record is encountered, truncate the log at the defective record’s location and continue with recovery. The original transaction log files are moved to a directory called savedLogFiles, which is created as a subdirectory of the log directory. The transaction log files are saved for diagnostic purposes.</td>
</tr>
</tbody>
</table>
**LogBufMB**

For TimesTen Classic, the LogBufMB attribute specifies the size of the internal transaction log buffer in megabytes, for the database.

For TimesTen Scaleout, the LogBufMB attribute specifies the size of the internal transaction log buffer in megabytes, for the element.

If you change the value of LogBufMB, you also may need to change the value of LogBufParallelism to satisfy the constraint that LogBufMB / LogBufParallelism >= 8.

If you increase the value of LogBufMB, ensure the value of LogFileSize is greater than or equal to the value of LogBufMB (LogFileSize >= LogBufMB).

**Required privilege**

Only the instance administrator can change the value of this attribute.

**Setting**

Set LogBufMB as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>LogBufMB</td>
<td>$n$ - Size of log buffer in megabytes.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td><strong>Log Buffer Size (MB) field</strong></td>
<td>Size of log buffer, in megabytes.</td>
</tr>
</tbody>
</table>

If not set and the database or element exists, TimesTen uses the value stored in the database.

If not set and the database or element is being created, TimesTen uses the default value of 64 MB.

The maximum value is 65,536 MB (64 GB).
LogBufParallelism

The LogBufParallelism attribute specifies the number of transaction log buffer strands to which TimesTen writes log files before the log is written to disk, allowing for improved log performance. Each buffer has its own insertion latch. Records are inserted in any of the strands. The log flusher gathers records from all strands and writes them to the log files.

If you change the value of LogBufParallelism, you also may need to change the value of LogBufMB to satisfy the constraint that LogBufMB/LogBufParallelism >= 8.

Required privilege

Only the instance administrator can change the value of this attribute.

Setting

Set LogBufParallelism as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>LogBufParallelism</td>
<td>An integer value between 1 and 64. Default is 4.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>Log Buffer Parallelism field</td>
<td>An integer value between 1 and 64. Default is 4.</td>
</tr>
</tbody>
</table>
LogFileSize

The LogFileSize attribute specifies the maximum size of transaction log files in megabytes. The minimum value is 8 MB. The default value is 64 MB. If you specify a size smaller than 8 MB, TimesTen returns an error message. Before TimesTen release 11.2.1.4, the minimum size was 1 MB. If you created your database in a previous release of TimesTen and specified a log file size of less than 8 MB, you must increase the value assigned to this attribute to avoid an error.

Actual transaction log file sizes can be slightly smaller or larger than LogFileSize because log records cannot span transaction log files.

If you specify a value of zero, TimesTen uses the default transaction log file size if the database does not exist. If the database exists, TimesTen uses the current specified transaction log file size.

Set the value of LogFileSize to be larger than or equal to the value of LogBufMB (LogFileSize \( \geq \) LogBufMB).

**Required privilege**

Only the instance administrator can change the value of this attribute.

**Setting**

Set LogFileSize as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>LogFileSize</td>
<td>( n ) - Size of transaction log file in megabytes. Default is the value of LogBufMB when the database is created and 0 (current size in effect) on subsequent connections. The minimum size is 8 MB. The maximum value is 65,536 MB (64 GB).</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>Log files Size (MB) field</td>
<td>Size of transaction log file in megabytes. Default is the value of LogBufMB when the database is created and 0 (current size in effect) on subsequent connections. The minimum size is 8 MB. The maximum value is 65,536 MB (64 GB).</td>
</tr>
</tbody>
</table>
**LogFlushMethod**

Controls the method used by TimesTen to write and sync log data to transaction log files. The overall throughput of a system can be significantly affected by the value of this attribute, especially if the application chooses to commit most transactions durably.

As a general rule, use the value 2 if most of your transactions commit durably and use the value 1 otherwise.

For best results, however, experiment with both values using a typical workload for your application and platform. Although application performance can be affected by this attribute, transaction durability is not affected. Changing the value of this attribute does not affect transaction durability in any way.

**Required privilege**

Only the instance administrator can change the value of this attribute.

**Setting**

Set LogFlushMethod as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>LogFlushMethod</td>
<td>0 - Write data to the transaction log files using the previously used value.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 (default) - Write data to transaction log files using buffered writes and use explicit sync operations as needed to sync log data to disk (for example with durable commits).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 - Write data to transaction log files using synchronous writes such that explicit sync operations are not needed.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Windows ODBC Data Source Administrator</th>
<th>Log Flush Method dropdown list</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 - Write data to the transaction log files using the previously used value.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 (default) - Write data to transaction log files using buffered writes and use explicit sync operations as needed to sync log data to disk (for example with durable commits).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 - Write data to transaction log files using synchronous writes such that explicit sync operations are not needed.</td>
<td></td>
</tr>
</tbody>
</table>

**See also**

DurableCommits
LogPurge

If the LogPurge attribute is set, TimesTen automatically removes transaction log files when they have been written to both checkpoint files and there are no transactions that still need the transaction log files’ contents. The first time checkpoint is called, TimesTen writes the contents of the transaction log files to one of the checkpoint files. When checkpoint is called the second time, TimesTen writes the contents of the transaction log files to the other checkpoint file.

TimesTen purges the transaction log files if all these conditions are met:
- The contents of the transaction log files have been written to both checkpoint files.
- The transaction log files are not pending incremental backup.
- If replication is being used, the transaction log files have been replicated to all subscribers.
- If XLA is being used, all XLA bookmarks have advanced beyond the transaction log files.
- The transaction log files are not being used by any distributed transactions using the XA interface.

If this attribute is set to 0 or unchecked, unneeded transaction log files are appended with the .arch suffix. Applications can then delete the files.

Required privilege

Only the instance administrator can change the value of this attribute.

Setting

Set LogPurge as follows:

<table>
<thead>
<tr>
<th>Where to set the attributes</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
</table>
| C or Java programs or UNIX and Linux systems odbc.ini file | LogPurge | 0 - Does not remove old transaction log files at connect and checkpoint.  
1 (default) - Removes old transaction log files at connect and checkpoint. |
| Windows ODBC Data Source Administrator | LogPurge check box | unchecked - Does not remove old transaction log files at connect and checkpoint.  
checked (default) - Removes old transaction log files at connect and checkpoint. |
MemoryLock

On Windows systems, TimesTen enables applications that connect to a shared database to specify whether the real memory should be locked while the database is being loaded into memory or while the store is in memory. If the physical memory used for the database is locked, the operating system's virtual memory subsystem cannot borrow that memory for other uses. No part of the database is ever paged out but this could lead to memory shortages in a system that is under configured with RAM. While memory locking can improve database load performance, it may impede other applications on the same computer.

On Linux, MemoryLock=4 will try to obtain a MemoryLock based on what the operating system allows. On Linux, locking all of the virtual memory size (physical + swap) can occur. TimesTen performs best if it does not use swap. Since the operating system allows locking more memory than is actually available, it is important to carefully configure the operating system memory management parameters to limit the amount of lockable memory. You can configure these parameters in the /etc/security/limits.conf file.

On AIX the MemoryLock attribute is not implemented.

The PL/SQL shared memory segment is not subject to MemoryLock.

Required privilege

Only the instance administrator can change the value of this attribute.

On Linux systems, set the groupname in the MemLock setting to be the same as the instance administrator in the /etc/security/limits.conf file. Set the value of MemLock to be at least as large as the TimesTen database shared memory segment.

To view privileges, use:

% ppriv $$

To add the privilege for user ID timesten, a root user uses:

# usermod -K defaultpriv=basic,proc_lock_memory timesten

After adding the privilege, the timesten user must log in to a new shell, unload all TimesTen databases from memory and restart the TimesTen daemons.

To restart the TimesTen daemons, in the new login shell, use:

% ttDaemonadmin -restart

Setting

Set MemoryLock as follows.
<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>MemoryLock</td>
<td>0 (default) - Does not lock memory.</td>
</tr>
</tbody>
</table>

1 - Tries to obtain a memory lock. If unable to lock, the connection succeeds. If a lock is obtained, it is released after the database is loaded into memory (recommended).

2 - A memory lock is required. If unable to lock, the connection fails. If a lock is obtained, the connection succeeds and the lock is released after the database is loaded into memory.

3 - Tries to obtain and keep a memory lock. If unable to lock, the connection succeeds. If a memory lock is obtained, the connection succeeds and the memory lock is held until the database is unloaded from memory.

4 - A memory lock is required and is held until the database is unloaded from memory. If unable to lock, the connection fails. If a lock is obtained, the connection succeeds and the memory lock is held until the database is unloaded from memory.

| Windows ODBC Data Source Administrator Memory Lock field | 0 (default) - Does not lock memory. |

1 - Tries to obtain a memory lock. If unable to lock, the connection succeeds. If a lock is obtained, it is released after the database is loaded into memory (recommended).

2 - A memory lock is required. If unable to lock, the connection fails. If a lock is obtained, the connection succeeds and the lock is released after the database is loaded into memory.

3 - Tries to obtain and keep a memory lock. If unable to lock, the connection succeeds. If a memory lock is obtained, the connection succeeds and the memory lock is held until the database is unloaded from memory.

4 - A memory lock is required and is held until the database is unloaded from memory. If unable to lock, the connection fails. If a lock is obtained, the connection succeeds and the memory lock is held until the database is unloaded from memory.
Overwrite

If the Overwrite attribute is set and there is an existing database with the same database path name as the new database, TimesTen destroys the existing database and creates a new empty database if the existing database is not in use. If the Overwrite attribute is set and there is not a database with the specified database path name, TimesTen only creates a new database if the AutoCreate attribute is also set (see "AutoCreate" on page 2-27). TimesTen ignores the Overwrite attribute if AutoCreate is set to 0. Applications should use caution when specifying the Overwrite = 1 attribute.

Required privilege

Only the instance administrator can change the value of this attribute. If a user other than an instance administrator attempts to connect to a database with Overwrite=1, TimesTen returns an error.

Setting

Set Overwrite as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>Overwrite</td>
<td>0 (default) - TimesTen does not overwrite an existing database with the same path name. 1 - TimesTen overwrites an existing database with the same path name.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>Not available</td>
<td>Not available.</td>
</tr>
</tbody>
</table>
PermSize

For TimesTen Classic, indicates the size in MB of the permanent memory region for the database.

For TimesTen Scaleout, indicates the size in MB of the permanent memory region for the element.

You may increase PermSize at first connect but not decrease it. TimesTen returns a warning if you attempt to decrease the permanent memory region size. If the database does not exist, a PermSize value of 0 or no value indicates to use the default size. Default size is 32 MB. For an existing database, a value of 0 or no value indicates that the existing size should not be changed.

Once you have created a database, you can make the permanent partition larger, but not smaller. See "Specifying the size of a database" in Oracle TimesTen In-Memory Database Operations Guide.

Also see information about the TempSize connection attribute.

The ttMigrate and ttDestroy utilities can also be used to change the Permanent Data Size, when appropriate.

Required privilege

Only the instance administrator can change the value of this attribute.

Setting

Set PermSize as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>PermSize</td>
<td>( n ) - Size of permanent partition in megabytes; default is 32 MB. Minimum size is 32 MB.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>Permanent Data Size field</td>
<td>( n ) - Size of permanent partition, in megabytes; default is 32 MB. Minimum size is 32 MB.</td>
</tr>
</tbody>
</table>
RecoveryThreads

The `RecoveryThreads` attribute determines the number of threads used to rebuild indexes during recovery.

If `RecoveryThreads=1`, during recovery, indexes that must be rebuilt are done serially. If you have enough processors available to work on index rebuilds on your computer, setting this attribute to a number greater than 1 can improve recovery performance. The performance improvement occurs only if different processors can work on different indexes. There is no parallelism in index rebuild within the same index.

The value of `RecoveryThreads` can be any value up to the number of CPUs available on your system.

The default is 1 when the database is created. Upon subsequent connections, if the database must be recovered and `RecoveryThreads` is unspecified or has a value of 0, then TimesTen uses the previous setting for this attribute.

Required privilege

Only the instance administrator can change the value of this attribute.

Setting

Set `RecoveryThreads` as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td><code>RecoveryThreads</code></td>
<td>( n ) - The number of threads to use when rebuilding indexes during recovery. Default is 4 when the database is created and 0 on subsequent connections.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator field</td>
<td><code>RecoveryThreads</code></td>
<td>( n ) - The number of threads to use when rebuilding indexes during recovery. Default is 4 when the database is created and 0 on subsequent connections.</td>
</tr>
</tbody>
</table>

Notes

For a progress report on the recovery process, see the rebuild messages in the support log.

Set the number of threads low enough to leave sufficient resources on the TimesTen server for other services/processes.
TempSize

For TimesTen Classic, TempSize indicates the total amount of memory in MB allocated to the temporary region for the database.

For TimesTen Scaleout, indicates the total amount of memory in MB allocated to the temporary region for an element.

TempSize has no predefined value. If left unspecified, its value is determined from PermSize as follows:

- If PermSize is less than 64 MB, TempSize = 32 MB + ceiling(PermSize / 4 MB).
- Otherwise, TempSize = 40 MB + ceiling(PermSize / 8 MB).

TimesTen rounds the value up to the nearest MB.

If specified, TimesTen always honors the TempSize value. Since the temporary data partition is recreated each time a database is loaded, the TempSize attribute can be increased or decreased each time a database is loaded. For an existing database, a value of 0 or no value indicates that the existing size should not be changed. The minimum TempSize is 32 MB.

Required privilege

Only the instance administrator can change the value of this attribute.

Setting

Set TempSize as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>TempSize</td>
<td>n - Size of the temporary partition, in MB. Minimum size is 32 MB on all platforms.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>Temporary Data Size field</td>
<td>n - Size of the temporary partition, in MB. Minimum size is 32 MB on all platforms.</td>
</tr>
</tbody>
</table>
General connection attributes

General connection attributes are set by each connection and persist for the duration of the connection. General connection attributes are listed in Table 2-3, "General connection attributes" and described in detail in this section.
CommitBufferSizeMax

CommitBufferSizeMax indicates the total amount of memory in MB allocated to the transaction commit buffer. Set this attribute to handle the size of reclaim records.

You can use the ALTER SESSION SQL statement, described in Oracle TimesTen In-Memory Database SQL Reference, to assign or change the maximum size of the commit buffer within a session. The new value takes effect when a new transaction starts.

```
ALTER SESSION SET COMMIT_BUFFER_SIZE_MAX = n;
```

You can see the configured maximum for the commit buffer by calling the `ttConfiguration` built-in procedure.

For more information on reclaim operations, including details about setting the commit buffer size, see "Transaction reclaim operations" in the Oracle TimesTen In-Memory Database Operations Guide. Also see information about the `ttCommitBufferStats` and the `ttCommitBufferStatsReset` built-in procedures.

**Required privilege**

No privilege is required to change the value of this attribute.

**Setting**

Set CommitBufferSize as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
</table>
| C or Java programs or UNIX and Linux systems `odbc.ini` file                               | CommitBufferSizeMax              | 0 - Commit buffer is configured to the default size. (10MB).  
|                                                                                            |                                  | n - Integer value. Minimum user configured size is 1 (MB). Configure this value to a value much smaller than `TempSize`.         |
| Windows ODBC Data Source Administrator                                                    | Maximum Commit Buffer Size field  | n - Integer value. The minimum user configured size is 1 (MB). Configure this value to a value much smaller than `TempSize`.        |

**Notes**

When you call the built-in procedure `ttCommitBufferStats`, the commit buffer statistics are expressed in bytes. However, the `ttConfiguration` built-in procedure output and the value set by the connection attribute `CommitBufferSizeMax` are expressed in MB.
**ConnectionName**

This attribute is also available as a Client connection attribute.

This attribute enables you to attach a symbolic name to any database connection. Connection names are unique within a process.

TimesTen uses the symbolic name to help identify the connection in various administrative utilities, such as `ttIsql`, `ttXactAdmin` and `ttStatus`. This can be particularly useful with processes that make multiple connections to the database, as is typical with multithreaded applications or in the identification of remote clients.

The value of this attribute is intended to be dynamically defined at connection time using the connection string. The default value is the connecting executable file name. It can also be defined statically in the DSN definition. Values used for `ConnectionName` should follow SQL identifier syntax rules.

**Required privilege**

No privilege is required to change the value of this attribute.

**Setting**

Set `ConnectionName` as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td><code>ConnectionName</code></td>
<td>Enter a string up to 30 characters that represents the name of the connection. If the specified or default connection name is in use, TimesTen assigns the name <code>conn</code>, where $n$ is an integer greater than 0 to make the name unique. If not specified, the connecting process name.</td>
</tr>
</tbody>
</table>

| Windows ODBC Data Source Administrator | `Connection` field | Enter a string up to 30 characters that represents the name of the connection. If the specified or default connection name is in use, TimesTen assigns the name `conn`, where $n$ is an integer greater than 0 to make the name unique. If not specified, the connecting process name. |
CreateEpochAtCommit

For TimesTen Scaleout, specifies if each commit generates an epoch.
Set this general connection attribute to '1' to make every commit from this connection an epoch. Default value is '0' which will make it so commits are not epochs unless the transaction included a call to the `ttEpochCreate` built-in procedure.'
TimesTen Classic ignores this attribute.

Required privilege
No privilege is required to change the value of this attribute.

Setting
Set `CreateEpochAtCommit` as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
</table>
| C or Java programs or UNIX and Linux systems odbc.ini file | CreateEpochAtCommit | 0 (default) - TimesTen does not write the transaction log to disk on transaction commit.  
1 - TimesTen writes log to disk on transaction commit. |
| Windows ODBC Data Source Administrator | Not available. | |

See also
LogFlushMethod
**DDLReplicationAction**

Determines whether a table or a sequence is included in an active standby pair replication scheme when created. The table can be included if the DDLReplicationLevel connection attribute is set to 2 or 3. The sequence can be included if the DDLReplicationLevel connection attribute is set to 3.

Replication of DDL operations is enabled (with restrictions) by the set value of the DDLReplicationLevel connection attribute. For more details, see "DDLReplicationLevel" on page 2-56.

The value can be modified by an ALTER SESSION SQL statement, described in Oracle TimesTen In-Memory Database SQL Reference. For example:

```sql
ALTER SESSION SET DDL_REPLICATION_ACTION='EXCLUDE';
```

Values set by ALTER SESSION override the value set by this attribute.

For examples of altering an active standby pair, see "Altering an Active Standby Pair" in the Oracle TimesTen In-Memory Database Replication Guide.

DDL operations are automatically committed. When RETURN TWOSAFE has been specified, errors and timeouts may occur as described in "RETURN TWOSAFE" in the Oracle TimesTen In-Memory Database Replication Guide. If a RETURN TWOSAFE timeout occurs, the DDL transaction is committed locally regardless of the LOCAL COMMIT ACTION that has been specified.

To learn more about replicating DDL, see "Making DDL changes in an active standby pair" in the Oracle TimesTen In-Memory Database Replication Guide.

**Required privilege**

ADMIN privilege is required if the value of this attribute is INCLUDE.

**Setting**

Set DDLReplicationAction as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>DDLReplicationAction</td>
<td>INCLUDE (default) - When a table or sequence is created, it is automatically added to the active standby pair scheme when the appropriate DDLReplicationLevel value is configured. EXCLUDE - When a table or sequence is created, it is not automatically included in the active standby pair.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator field</td>
<td>DDLReplicationAction</td>
<td>INCLUDE (default) - When a table or sequence is created, it is automatically added to the active standby pair scheme when the appropriate DDLReplicationLevel value is configured. EXCLUDE - When a table or sequence is created, it is not automatically included in the active standby pair.</td>
</tr>
</tbody>
</table>

Connection Attributes 2-55
DDLReplicationLevel

Enables replication of a subset of data definition language (DDL) statements (with restrictions) in an active standby replication scheme.

When the value of the DDLReplicationLevel connection attribute is set to 1, CREATE or DROP statements for tables, indexes, or synonyms are not replicated to the standby database. However, you can add or drop columns with the ALTER TABLE ADD or DROP COLUMN to or from a replicated table, and those actions are replicated to the standby database.

When the value of the DDLReplicationLevel connection attribute is set to 2 (the default), the following DDL statements (described in Oracle TimesTen In-Memory Database SQL Reference) are replicated to the standby and any subscribers:

- CREATE INDEX or DROP INDEX
- CREATE SYNONYM or DROP SYNONYM
- CREATE TABLE or DROP TABLE (including global temporary tables but not CREATE TABLE AS SELECT)

When the value of the DDLReplicationLevel connection attribute is set to 3, the following DDL statements (described in Oracle TimesTen In-Memory Database SQL Reference) and those replicated when the value is set to 2 are replicated to the standby and any subscribers:

- CREATE VIEW or DROP VIEW
- CREATE SEQUENCE or DROP SEQUENCE
- Replication of the results to the standby master when setting the cache administration user name and password with the UidPwdSet built-in procedure. You do not need to stop and restart the cache agent or replication agent to execute the UidPwdSet built-in procedure. For more information, see "Changing cache user names and passwords" in the Oracle TimesTen Application-Tier Database Cache User’s Guide or "ttCacheUidPwdSet" on page 3-74.

The value of this attribute can be modified by an ALTER SESSION statement, described in Oracle TimesTen In-Memory Database SQL Reference. For example:

```
ALTER SESSION SET DDL_REPLICATION_LEVEL=3;
```

Values set by ALTER SESSION override the value set by this attribute.

For examples of altering an active standby pair, see "Altering an Active Standby Pair" in the Oracle TimesTen In-Memory Database Replication Guide.

To learn more about replicating DDL, see "Making DDL changes in an active standby pair" in the Oracle TimesTen In-Memory Database Replication Guide.

**Required privilege**

No privilege is required to change the value of this attribute.

**Setting**

Set DDLReplicationLevel as follows:
Replication of DDL operations has these restrictions:

- `CREATE TABLE AS SELECT` statements are not replicated.
- The `CREATE INDEX` statement is replicated only when the index is created on an empty table.
- To control whether a table or sequence is included in an active standby pair replication scheme at the time of creation, use the `DDLReplicationAction` connection attribute.
- Sequences with the `CYCLE` attribute cannot be replicated.
- Objects are replicated only when the receiving database is of a TimesTen release that supports that level of replication, and is configured for an active standby pair replication scheme. For example, replication of sequences (requiring `DDL_REPLICATION_LEVEL=3`) to a database release prior to 11.2.2.7.0 is not supported. When `DDLReplicationLevel` value is set to 3, both the active and standby master databases need to be TimesTen Release 11.2.2.7 or later. When `DDL_REPLICATION_LEVEL=2`, the receiving database must be at least release 11.2.1.8.0 for replication of objects to be supported.
- All restrictions for the `UidPwdSet` built-in procedure apply.
- When `DDLReplicationLevel=1` or `2`, you cannot alter a table to add a `NOT NULL` column to a table that is part of a replication scheme with the `ALTER TABLE ... ADD COLUMN NOT NULL DEFAULT` statement. You must remove the table from the replication scheme first before you can add a `NOT NULL` column to it. However, if `DDLReplicationLevel=3`, then you can alter a table to add a `NOT NULL` column to a table that is part of a replication scheme.
Diagnostics

Enables an application to configure the level of diagnostics information generated by TimesTen for the connection. TimesTen diagnostics messages are warnings whose numbers lie within the range 20000 through 29999. Diagnostics connection attribute values are integers.

Required privilege

No privilege is required to change the value of this attribute.

Setting

Set Diagnostics as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>Diagnostics</td>
<td>0 - TimesTen does not generate diagnostic messages.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 (default) - TimesTen generates base-level diagnostics messages.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>Diagnostics field</td>
<td>0 - TimesTen does not generate diagnostic messages.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 (default) - TimesTen generates base-level diagnostics messages.</td>
</tr>
</tbody>
</table>
**DurableCommits**

By default, `DurableCommits` is set to 0. With this setting, TimesTen writes a log record to the file system when a transaction is committed, but the log record is not immediately written to disk. This reduces transaction execution time at the risk of losing some committed transactions if a failure occurs. When `DurableCommits` is set to 1, TimesTen writes a log record to disk when the transaction is committed.

A connection can also call the `ttDurableCommit` built-in procedure to do durable commits explicitly on selected transactions. A call to `ttDurableCommit` flushes the log buffer to disk. The log buffer is shared among all connections and contains log records from transactions of all connections.

Log records are continually copied from the file system to disk. You can use `LogFlushMethod` to control when the file system is synchronized with the disk.

**Required privilege**

No privilege is required to change the value of this attribute.

**Setting**

Set `DurableCommits` as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td><code>DurableCommits</code></td>
<td>0 (default) - TimesTen does not write the transaction log to disk on transaction commit. 1 - TimesTen writes log to disk on transaction commit.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td><code>Durable Commits</code> check box</td>
<td>unchecked (default) - TimesTen does not write the transaction log to disk on transaction commit. checked - TimesTen writes log to disk on transaction commit.</td>
</tr>
</tbody>
</table>

**See also**

`LogFlushMethod`
IncludeInCore

This attribute allows the application to control whether TimesTen shared memory should be included in application core dumps, and which portions of that memory should be included.

If multiple connections exist from a single application process to a single TimesTen database, the IncludeInCore value of the most recent connection of the process determines the parts of the core file to dump.

For Client/Server connections, the setting is passed to TimesTen Server, which passes it on to the direct driver.

TimesTen daemons always dump everything.

The settings noted below are additive. For example, set IncludeInCore to 3 (1+2) for DB header and other fixed allocations plus perm space. Set it to 15 (1+2+4+8) for DB header and other fixed allocations plus perm space, temp space, and log buffer.

Required privilege

ADMIN privilege is required to include the DB header and other fixed allocations.

Setting

Set IncludeInCore as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>IncludeInCore</td>
<td>0 - Exclude the shared memory associated with this database connection from any core files. 1 - Include the DB header and other fixed allocations from this database in core files. This is necessary to make sense of other information you request. 2 - Include perm space from this database in core files. 4 - Include temp space from this database in core files. 8 - Include the log buffer from this database in core files. 16 - Include the PL/SQL shared memory from this database in core files. The default value is 255.</td>
</tr>
</tbody>
</table>

Windows ODBC Data Source Administrator | Not supported. |
**Isolation**

By default, TimesTen uses read committed isolation. The Isolation attribute specifies the initial transaction isolation level for the connection. For a description of the isolation levels, see "Concurrency control through isolation and locking" in *Oracle TimesTen In-Memory Database Operations Guide*.

The value can be modified by an ALTER SESSION statement, described in *Oracle TimesTen In-Memory Database SQL Reference*. For example:

```
ALTER SESSION SET ISOLATION_LEVEL=serializable;
```

CREATE CACHE GROUP, ALTER CACHE GROUP and DROP CACHE GROUP statements are not supported in serializable isolation mode.

If the passthrough or the propagate TimesTen Cache feature is used, the TimesTen isolation level setting is inherited by the Oracle session. TimesTen serializable mode is mapped to Oracle’s serializable mode. TimesTen read committed mode is mapped to Oracle’s read committed mode. For more details on the passthrough attribute, see "PassThrough" on page 2-104.

With PassThrough set to 3, you must use an ALTER SESSION statement to permanently modify the isolation level on the Oracle database connection. For example on a connection to the DSN repdb1_181:

1. **Call ttIsql and connect to the DSN with PassThrough level 3:**
   ```
   % ttIsql;
   Command> connect "dsn=repdb1_181;passthrough=3";
   Connection successful: . . . PassThrough=3;
   <default setting Autocommit=1>
   ```

2. **Turn off AutoCommit:**
   ```
   Command> autocommit=0;
   ```

3. **Temporarily change the PassThrough level to 0:**
   ```
   Command> passthrough=0;
   ```

4. **Alter the isolation level to serializable:**
   ```
   Command> prepare 1 ALTER SESSION SET ISOLATION_LEVEL=serializable;
   commit;
   exec=1;
   ```

**Required privilege**

No privilege is required to change the value of this attribute.

**Setting**

Set Isolation as follows:
<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
</table>
| C or Java programs or UNIX and Linux systems odbc.ini file | Isolation | 0 - Connects to database in serializable isolation mode.  
1 (default) - Connects to database in read committed isolation mode. |

| Windows ODBC Data Source Administrator | Isolation dropdown list | 0 - Connects to database in serializable isolation mode.  
1 (default) - Connects to database in read committed isolation mode. |
LockLevel

By default, TimesTen enables row-level locking for maximum concurrency. With row-level locking, transactions usually obtain locks on the individual rows that they access, although a transaction may obtain a lock on an entire table if TimesTen determines that doing so would result in better performance. Row-level locking is the best choice for most applications, as it provides the finest granularity of concurrency control. To use row-level locking, applications must set the LockLevel connection attribute to 0 (the default value). To cache Oracle database tables, you must set row-level locking. To CREATE, DROP, or ALTER a user, you can only use row-level locking and thus, the lock level must be set to 0 before you can perform any of these operations.

To give every transaction in this connection exclusive access to the database, you can enable database-level locking by setting the LockLevel attribute to 1. Doing so may improve performance for some applications.

A connection can change the desired lock level at any time by calling the \texttt{ttLockLevel} built-in procedure. Connections can also wait for unavailable locks by calling the \texttt{ttLockWait} built-in procedure. Different connections can coexist with different levels of locking, but the presence of even one connection doing database-level locking leads to loss of concurrency. To display a list of all locks on a particular database you can use the \texttt{ttXactAdmin} utility.

When using PL/SQL in your applications, set \texttt{LockLevel} = 0 and selectively change to database level locking for specific transactions that require that level of locking by using the \texttt{ttLockLevel} built-in procedure.

\textbf{Required privilege}

\texttt{ADMIN} privilege is required if the value of this attribute is 1.

\textbf{Setting}

Set LockLevel as follows:

\begin{tabular}{|l|l|l|}
\hline
\textbf{Where to set the attribute} & \textbf{How the attribute is represented} & \textbf{Setting} \\
\hline
C or Java programs \hspace{1cm} or UNIX and Linux \hspace{1cm} systems odbc.ini file & LockLevel & \texttt{0} (default) - Transactions access the database using row-level locking. \\
& & \texttt{1} - Transactions access the database by acquiring an exclusive lock on the entire database. \\
\hline
Windows ODBC Data Source Administrator & DS-Level Locking check box & \texttt{unchecked} (default) - Transactions access the database using row-level locking. \\
& & \texttt{checked} - Transactions access the database by acquiring an exclusive lock on the entire database. \\
\hline
\end{tabular}
LockWait

Enables an application to configure the lock wait interval for the connection. The lock wait interval is the number of seconds to wait for a lock when there is contention on it. Sub-second LockWait values significant to tenths of a second can be specified using decimal format for the number of seconds. For example:

\[
\text{LockWait} = 0.1
\]

results in a lock wait of one tenth of a second.

LockWait can be set to any value between 0 and 1,000,000 inclusive to a precision of tenths of a second. The default is 10 seconds:

\[
\text{LockWait} = 10.0
\]

Actual lock wait response time is imprecise and can be exceeded by up to one tenth of a second, due to the scheduling of the agent that detects timeouts. This imprecision does not apply to zero second timeouts, which are always reported immediately.

The number of connections to a database can impact the time needed to resolve lock contentions. If you anticipate having many connections to the database, increase the lock wait interval.

A connection can change the lock wait interval at any time by calling the built-in procedure.

To display a list of all locks on a particular database you can use the TimesTen utility ttXactAdmin.

Required privilege

No privilege is required to change the value of this attribute.

Setting

Set LockWait as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>LockWait</td>
<td>( s ) - Indicates the number of seconds to wait for locking conflict resolution before timing out. The default is 10 seconds.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>LockWait field</td>
<td>( s ) - Indicates the number of seconds to wait for locking conflict resolution before timing out. The default is 10 seconds.</td>
</tr>
</tbody>
</table>
OptimizerHint

The OptimizerHint connection attribute establishes the connection’s optimizer hint defaults. They can be different than the system defaults. The optimizer hints set with this connection attribute are set for every SQL statement in the user application.

The value of this attribute is a string of the same format as the statement level optimizer hints, but without the delimiters ++, */ and -. The string can only contain the optimizer hint names. It cannot be mixed with other hint strings.

The order of precedence for optimizer hints is statement level hints, transaction level hints and lastly hints set by this connection attribute.

For client/server applications, the attribute set by the client connection takes precedence over server DSN settings of this attribute.

Some symbols, such as semi-colons (;) are not accepted in attribute values. For hints where the parameter might contain a semi-colon, multiple hints of the same name are combined into one hint. For example, to express:

\[\text{TT\_INDEX (t1,i1, 0; t2, i2,0)}\]

use

\[\text{TT\_INDEX (t1,i1,0) TT\_INDEX (t2, i2,0 )}\]

To combine multiple hints at the connection level, you must enter them in the same line.

For a list of optimizer hints supported as values to this attribute, see "Statement level optimizer hints" in the Oracle TimesTen In-Memory Database SQL Reference.

**Required privilege**

No privilege is required to change the value of this attribute.

**Setting**

Set OptimizerHint as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>OptimizerHint</td>
<td>A string specifying optimizer hints and their values. The maximum length of the string is 512.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>Optimizer Hint field</td>
<td>A string specifying optimizer hints and their values. The maximum length of the string is 512.</td>
</tr>
</tbody>
</table>
PermWarnThreshold

Indicates the threshold percentage at which TimesTen issues out-of-memory warnings for the permanent partition of the database's memory. The database is considered no longer out of permanent memory if it falls 10% below this threshold. An application must call the built-in procedure `ttWarnOnLowMemory` to receive out-of-memory warnings.

**Required privilege**

No privilege is required to change the value of this attribute.

**Setting**

Set `PermWarnThreshold` as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td><code>PermWarnThreshold</code></td>
<td>$p$ - Percentage at which TimesTen should issue out-of-memory warnings. Default is 90.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td><code>Low Memory Warning Thresholds for Permanent Data field</code></td>
<td>$p$ - Percentage at which TimesTen should issue out-of-memory warnings. Default is 90.</td>
</tr>
</tbody>
</table>
PrivateCommands

When multiple connections execute the same command, they access common command structures controlled by a single command lock. To avoid sharing their commands and possibly placing contention on the lock, you can use PrivateCommands. This gives you better scaling at the cost of increased temporary space usage.

By default, the PrivateCommands is turned off and commands are shared.

Required privilege

No privilege is required to change the value of this attribute.

Setting

Set PrivateCommands as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX</td>
<td>PrivateCommands</td>
<td>0 (default) - Commands are shared with other connections.</td>
</tr>
<tr>
<td>and Linux systems odbc.ini file</td>
<td></td>
<td>1 - Commands are not shared with any other connection.</td>
</tr>
<tr>
<td>Windows ODBC Data Source</td>
<td>Private Commands field</td>
<td>0 (default) - Commands are shared with other connections.</td>
</tr>
<tr>
<td>Administrator</td>
<td></td>
<td>1 - Commands are not shared with any other connection.</td>
</tr>
</tbody>
</table>

Notes

If there are many copies of the same command, all of them are invalidated by a DDL or statistics change. Reprepare of these multiple copies takes longer when PrivateCommands = 1. With more commands DDL execution can take slightly longer.

When using the PrivateCommands attribute, memory consumption can increase considerably if the attribute is not used cautiously. For example, if PrivateCommands=1 for an application that has 100 connections with 100 commands, there are 10,000 commands in the system: one private command for each connection.
**PWDCrypt**

The **PWDCrypt** contains an encrypted version of the corresponding **PWD** value. The value for **PWD** is stored in clear text, which does not allow special characters, in the `.odbc.ini` file on UNIX and Linux systems and in the Windows Registry on Windows. Any users who have access to the `.odbc.ini` file or Windows Registry can view the value for this attribute. The **PWDCrypt** attribute enables special characters, is case sensitive and contains the value of the encrypted password.

For security reasons, the **PWDCrypt** attribute should only be placed in User DSNs or user private `ODBCINI` files. The presence of the **PWDCrypt** in System DSNs enables any user to use the **PWDCrypt** value to connect to TimesTen, even though they have no knowledge of the cleartext password.

To generate the value for this attribute, run the `ttUser` utility.

**Required privilege**

No privilege is required to change the value of this attribute.

**Notes**

If **PWD** and **PWDCrypt** are both supplied, TimesTen uses the value of the **PWD** attribute. See "UID and PWD" on page 2-73.

TimesTen does not store the value of the **PWD** attribute anywhere in the TimesTen system.

See "Required user authentication for utilities" in the description of "UID and PWD" for details about the treatment of passwords when using utilities that require specific privileges.

**Setting**

Set **PWDCrypt** as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems <code>.odbc.ini</code> file</td>
<td><strong>PWDCrypt</strong></td>
<td>Enter the value generated by the <code>ttUser</code> utility.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td><strong>PWDCrypt</strong> field</td>
<td>Enter the value generated by the <code>ttUser</code> utility.</td>
</tr>
</tbody>
</table>
QueryThreshold

Use this attribute to write a warning to the support log when the execution time of a SQL statement exceeds the specified value. You cannot set a query threshold for a SQL statement that is executed by the cache agent. The value of QueryThreshold applies to all connections. It applies to all SQL statements except those executed by the replication agent or the cache agent.

The value of this attribute can be any integer equal to or greater than 0. The default value is 0. A value of 0 indicates that no warning is issued. The unit is seconds.

Required privilege

No privilege is required to change the value of this attribute.

Setting

Set QueryThreshold as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>QueryThreshold</td>
<td>A nonnegative integer. Default is 0 and indicates that TimesTen does not return a warning.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>QueryThreshold (secs) field</td>
<td>A nonnegative integer. Default is 0 and indicates that TimesTen does not return a warning.</td>
</tr>
</tbody>
</table>
ReplicationTrack

When managing track-based parallel replication, you can assign a connection to a replication track. All transactions issued by the connection are assigned to this track, unless the track is altered.

To start track-based parallel replication you must set a value for the ReplicationParallelism attribute, specifying the number of replication tracks to be applied in parallel. You must also set ReplicationApplyOrdering to 2.

The Track_ID column of the TTREP.REPPEERS system table (described in Oracle TimesTen In-Memory Database System Tables and Views Reference) shows the track associated with the connection.

You can use the ALTER SESSION SQL statement, described in Oracle TimesTen In-Memory Database SQL Reference, to assign or change the value of this attribute within a session. For example:

```
ALTER SESSION SET REPLICATION_TRACK=4;
```

**Required privilege**

No privilege is required to change the value of this attribute.

**Setting**

Set ReplicationTrack as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>ReplicationTrack</td>
<td>( n ) - An integer between 1 and 64 that specifies the replication track to be used by transactions issued by the connection.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>Replication Track field</td>
<td>( n ) - An integer between 1 and 64 that specifies the replication track to be used by transactions issued by the connection.</td>
</tr>
</tbody>
</table>
SQLQueryTimeout

Use this attribute to specify the time limit in seconds within which the database should execute SQL statements.

The value of SQLQueryTimeout can be any integer equal to or greater than 0. The default value is 0. A value of 0 indicates that the query does not time out.

This attribute does not stop TimesTen Cache operations that are being processed on an Oracle database. This includes passthrough statements, flushing, manual loading, manual refreshing, synchronous writethrough, and propagating.

Required privilege

No privilege is required to change the value of this attribute.

Setting

Set SQLQueryTimeout as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems <code>odbc.ini</code> file</td>
<td>SQLQueryTimeout</td>
<td>( n ) - Time limit in seconds for which the database should execute SQL queries.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>QueryTimeout (secs) field</td>
<td>( n ) - Time limit in seconds for which the database should execute SQL queries.</td>
</tr>
</tbody>
</table>
**TempWarnThreshold**

Indicates the threshold percentage at which TimesTen issues out-of-memory warnings for the temporary partition of the database's memory. The database is considered no longer out of temporary memory if it falls 10% below this threshold. An application must call the built-in procedure `ttWarnOnLowMemory` to receive out-of-memory warnings. See "ttWarnOnLowMemory" on page 3-255.

**Required privilege**

No privilege is required to change the value of this attribute.

**Setting**

Set `TempWarnThreshold` as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td><code>TempWarnThreshold</code></td>
<td><code>p</code> - Percentage at which warning should be issued. Default is 90.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td><strong>Low Memory Warning Thresholds for Temporary Data field</strong></td>
<td><code>p</code> - Percentage at which warning should be issued. Default is 90.</td>
</tr>
</tbody>
</table>
UID and PWD

A user ID and password must be provided by a user who is identified internally to TimesTen. Alternatively, an encrypted password can be supplied using the PwDCrypt attribute. Some TimesTen operations prompt for the UID and PWD of the user performing the operation.

For TimesTen client/server applications, specify UID and PWD either in the Client DSN configuration or in the connection string. The UID and PWD values specified in a connection string take precedence over the values specified in the Client DSN configuration.

When caching Oracle database tables, PWD specifies the TimesTen password while OraclePWD specifies the Oracle password.

Required user authentication for utilities

All utilities that require a password prompt for one.

If a UID connection attribute is given but no PWD attribute is given, either through a connection string or in the ODBCINI file for the specified DSN, TimesTen prompts for a password. When explicitly prompted, input is not displayed on the command line.

A password given on the command line, before TimesTen prompts for the password, is visible to the ps command, so use of the PWD connection attribute is not recommended in the first call to the utility. For example, the following usage is not recommended:

```bash
% ttIsqI -connStr 'DSN=mydsn;UID=terry;PWD=secret';
```

Generally, when no UID connection attribute is given, the UID is assumed to be the user name identified by the operating system, and TimesTen does not prompt for a password.

When a utility accepts a DSN, connection string or database path as a parameter, specify the value at the end of the command line.

Required privilege

No privilege is required to change the values of these attributes.

Setting

Set UID and PWD as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>UID</td>
<td>Character string specifying the user ID.</td>
</tr>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>PWD</td>
<td>Character string specifying the password that corresponds to the user ID.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>User ID field</td>
<td>Character string specifying the user ID.</td>
</tr>
</tbody>
</table>
**WaitForConnect**

When an application requests a connection to a TimesTen database and the connection is not possible (perhaps during concurrent loading/recovery of a database), TimesTen normally waits for completion of the conflicting connection. In some cases, it can take some time for an application to connect to a database. If the `WaitForConnect` attribute is off and the database is not immediately accessible, TimesTen returns immediately an error. For a description of the error, look for the error message number in "Warnings and Errors" in *Oracle TimesTen In-Memory Database Error Messages*.

**Required privilege**

No privilege is required to change the value of this attribute.

**Setting**

Set `WaitForConnect` as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
</table>
| C or Java programs or UNIX and Linux systems odbc.ini file | `WaitForConnect` | 0 - Does not wait if connection to database fails.  
1 (default) - Waits until connection to database is possible. |
| Windows ODBC Data Source Administrator | **Wait For Connect** check box | **unchecked** - Does not wait if connection to database fails.  
**checked** (default) - Waits until connection to database is possible. |
NLS general connection attributes

NLS connection attributes are set by each connection and persist for the duration of the connection. These attributes control the globalization behaviors of the database. NLS general connection attributes are listed Table 2–4, "NLS general connection attributes" and described in detail in this section.

You can use the ALTER SESSION statement, described in Oracle TimesTen In-Memory Database SQL Reference, to change NLS parameters to override the values that are assigned to these attributes at connection time.
ConnectionCharacterSet

ConnectionCharacterSet is also available as a Client connection attribute.

This attribute specifies the character encoding for the connection, which can be different from the database character set. This can be useful when you have multiple connections to a database and one or more of those connections requires a character set that differs from that specified in the database.

The connection character set determines the character set in which data is displayed or presented.

Generally, you should choose a connection character set that matches your terminal settings or data source. Your database character set should be chosen based on the data requirements. For example: Do you have data in Unicode or is your data in Japanese on UNIX or Linux (EUC) or Windows (SJIS)?

When the database and connection character sets differ, TimesTen performs data conversion internally based on the connection character set. If the connection and database character sets are the same, TimesTen does not need to convert or interpret the data set. Best performance occurs when connection and database character sets match, since no conversion is required.

Parameters and SQL query text sent to the connect should be in the connection character set. Results and error messages returned by the connection are returned in the connection character set.

This attribute accepts the same values used for the DatabaseCharacterSet. For a list of supported character set names, see “Supported character sets” on page 2-13.

Required privilege

No privilege is required to change the value of this attribute.

Setting

Set ConnectionCharacterSet as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>ConnectionCharacterSet</td>
<td>The default value for ConnectionCharacterSet is US7ASCII.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>Connection CharacterSet list</td>
<td>The default value for ConnectionCharacterSet is US7ASCII.</td>
</tr>
</tbody>
</table>
NLS_LENGTH_SEMANTICS

TimesTen uses the NLS_LENGTH_SEMANTICS attribute to set the default length semantics configuration. Length semantics determines how the length of a character string is determined. The length can be treated as a sequence of characters or a sequence of bytes.

NLS_LENGTH_SEMANTICS can be modified by an ALTER SESSION SQL statement, described in Oracle TimesTen In-Memory Database SQL Reference.

Required privilege

No privilege is required to change the value of this attribute.

Setting

Set NLS_LENGTH_SEMANTICS as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>NLS_LENGTH_SEMANTICS</td>
<td>Specify either BYTE (default) or CHAR.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>NLS_LENGTH_SEMANTICS list</td>
<td>Select either BYTE (default) or CHAR.</td>
</tr>
</tbody>
</table>
NLS_NCHAR_CONV_EXCP

The `NLS_NCHAR_CONV_EXCP` attribute determines whether an error is reported when there is data loss during an implicit or explicit character type conversion between `NCHAR/NVARCHAR2` data and `CHAR/VARCHAR2` data. A replacement character is substituted for characters that cannot be converted.

Implicit and explicit conversions between `CHAR` and `NCHAR` are supported.

`NLS_NCHAR_CONV_EXCP` can be modified by an `ALTER SESSION` SQL statement, described in Oracle TimesTen In-Memory Database SQL Reference.

**Required privilege**

No privilege is required to change the value of this attribute.

**Setting**

Set `NLS_NCHAR_CONV_EXCP` as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td><code>NLS_NCHAR_CONV_EXCP</code></td>
<td>0 (default) - Errors are not reported when there is a data loss during character type conversion. 1 - Errors are reported when there is a data loss during character type conversion.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td><code>NLS_NCHAR_CONV_EXCP</code> check box</td>
<td>unchecked (default) - Error messages are not reported when there is a data loss during character type conversion. checked - Error messages are reported when there is a data loss during character type conversion.</td>
</tr>
</tbody>
</table>
NLS_SORT

The NLS_SORT attribute indicates which collating sequence to use for linguistic comparisons. It accepts the values listed in "Supported Linguistic Sorts." All these values can be modified to do case-insensitive sorts by appending _CI to the value. To perform accent-insensitive and case-insensitive sorts, append _AI to the value.

For materialized views and cache groups, TimesTen recommends that you explicitly specify the collating sequence using the NLSSORT SQL function rather than using this attribute in the connection string or DSN definition.

Operations involving character comparisons support linguistic case-sensitive collating sequences. Case-insensitive sorts may affect DISTINCT value interpretation.

NLS_SORT may affect many operations. The supported operations that are sensitive to collating sequence are:

- MIN, MAX
- BETWEEN
- =, <>, >, >=, <, <=
- DISTINCT
- CASE
- GROUP BY
- HAVING
- ORDER BY
- IN
- LIKE

NLS_SORT settings other than BINARY may have significant performance impact on character operations.

NLS_SORT can be modified by an ALTER SESSION SQL statement, described in Oracle TimesTen In-Memory Database SQL Reference.

---

**Note:** Primary key indexes are always based on the BINARY collating sequence. Use of non-BINARY NLS_SORT equality searches cannot use the primary key index.

---

**Required privilege**

No privilege is required to change the value of this attribute.

**Setting**

Set NLS_SORT as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>NLS_SORT</td>
<td>Specify the linguistic sort sequence or BINARY (default).</td>
</tr>
</tbody>
</table>
Supported linguistic sorts

The tables in this section list the supported values for the `NLS_SORT` general connection attribute and the `NLS_SORT` SQL function.

### Monolingual linguistic sorts

<table>
<thead>
<tr>
<th>Basic name</th>
<th>Extended name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARABIC</td>
<td>-</td>
</tr>
<tr>
<td>ARABIC_MATCH</td>
<td>-</td>
</tr>
<tr>
<td>ARABIC_ABJ_SORT</td>
<td>-</td>
</tr>
<tr>
<td>ARABIC_ABJ_MATCH</td>
<td>-</td>
</tr>
<tr>
<td>ASCII7</td>
<td>-</td>
</tr>
<tr>
<td>AZERBAIJANI</td>
<td>XAZERBAIJANI</td>
</tr>
<tr>
<td>BENGALI</td>
<td>-</td>
</tr>
<tr>
<td>BIG5</td>
<td>-</td>
</tr>
<tr>
<td>BINARY</td>
<td>-</td>
</tr>
<tr>
<td>BULGARIAN</td>
<td>-</td>
</tr>
<tr>
<td>CANADIAN FRENCH</td>
<td>-</td>
</tr>
<tr>
<td>CATALAN</td>
<td>XCATALAN</td>
</tr>
<tr>
<td>CROATIAN</td>
<td>XCROATIAN</td>
</tr>
<tr>
<td>CZECH</td>
<td>XCZECH</td>
</tr>
<tr>
<td>CZECH_PUNCTUATION</td>
<td>XCZECH_PUNCTUATION</td>
</tr>
<tr>
<td>DANISH</td>
<td>XDANISH</td>
</tr>
<tr>
<td>DUTCH</td>
<td>XDUTCH</td>
</tr>
<tr>
<td>EBCDIC</td>
<td>-</td>
</tr>
<tr>
<td>EBC_EURO</td>
<td>-</td>
</tr>
<tr>
<td>EBC_EUROPA3</td>
<td>-</td>
</tr>
<tr>
<td>ESTONIAN</td>
<td>-</td>
</tr>
<tr>
<td>FINNISH</td>
<td>-</td>
</tr>
<tr>
<td>FRENCH</td>
<td>XFRENCH</td>
</tr>
<tr>
<td>GERMAN</td>
<td>XGERMAN</td>
</tr>
<tr>
<td>GERMAN_DIN</td>
<td>XGERMAN_DIN</td>
</tr>
<tr>
<td>GBK</td>
<td>-</td>
</tr>
<tr>
<td>GREEK</td>
<td>-</td>
</tr>
<tr>
<td>HEBREW</td>
<td>-</td>
</tr>
<tr>
<td>HKSCS</td>
<td>-</td>
</tr>
</tbody>
</table>
### Multilingual linguistic sorts

<table>
<thead>
<tr>
<th>Sort name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CANADIAN_M</td>
<td>Canadian French sort supports reverse secondary, special expanding characters.</td>
</tr>
<tr>
<td>DANISH_M</td>
<td>Danish sort supports sorting uppercase characters before lowercase characters.</td>
</tr>
<tr>
<td>FRENCH_M</td>
<td>French sort supports reverse sort for secondary.</td>
</tr>
<tr>
<td>GENERIC_M</td>
<td>Generic sorting order which is based on ISO14651 and Unicode canonical equivalence rules but excluding compatible equivalence rules.</td>
</tr>
<tr>
<td>JAPANESE_M</td>
<td>Japanese sort supports SJIS character set order and EUC characters which are not included in SJIS.</td>
</tr>
<tr>
<td>KOREAN_M</td>
<td>Korean sort Hangul characters are based on Unicode binary order. Hanja characters based on pronunciation order. All Hangul characters are before Hanja characters.</td>
</tr>
</tbody>
</table>

### Sort group name

<table>
<thead>
<tr>
<th>Basic name</th>
<th>Extended name</th>
</tr>
</thead>
<tbody>
<tr>
<td>HUNGARIAN</td>
<td>XHUNGARIAN</td>
</tr>
<tr>
<td>ICELANDIC</td>
<td>-</td>
</tr>
<tr>
<td>INDONESIAN</td>
<td>-</td>
</tr>
<tr>
<td>ITALIAN</td>
<td>-</td>
</tr>
<tr>
<td>LATIN</td>
<td>-</td>
</tr>
<tr>
<td>LATVIAN</td>
<td>-</td>
</tr>
<tr>
<td>LITHUANIAN</td>
<td>-</td>
</tr>
<tr>
<td>MALAY</td>
<td>-</td>
</tr>
<tr>
<td>NORWEGIAN</td>
<td>-</td>
</tr>
<tr>
<td>POLISH</td>
<td>-</td>
</tr>
<tr>
<td>PUNCTUATION</td>
<td>XPUNCTUATION</td>
</tr>
<tr>
<td>ROMANIAN</td>
<td>-</td>
</tr>
<tr>
<td>RUSSIAN</td>
<td>-</td>
</tr>
<tr>
<td>SLOVAK</td>
<td>XSLOVAK</td>
</tr>
<tr>
<td>SLOVENIAN</td>
<td>XSLOVENIAN</td>
</tr>
<tr>
<td>SPANISH</td>
<td>XSPANISH</td>
</tr>
<tr>
<td>SWEDISH</td>
<td>-</td>
</tr>
<tr>
<td>SWISS</td>
<td>XSWISS</td>
</tr>
<tr>
<td>THAI_DICTIONARY</td>
<td>-</td>
</tr>
<tr>
<td>TURKISH</td>
<td>XTURKISH</td>
</tr>
<tr>
<td>UKRAINIAN</td>
<td>-</td>
</tr>
<tr>
<td>UNICODE_BINARY</td>
<td>-</td>
</tr>
<tr>
<td>VIETNAMESE</td>
<td>-</td>
</tr>
<tr>
<td>WEST_EUROPEAN</td>
<td>XWEST_EUROPEAN</td>
</tr>
<tr>
<td>Sort name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SPANISH_M</td>
<td>Traditional Spanish sort supports special contracting characters.</td>
</tr>
<tr>
<td>THAI_M</td>
<td>Thai sort supports swap characters for some vowels and consonants.</td>
</tr>
<tr>
<td>SCHINESE_RADICAL_M</td>
<td>Simplified Chinese sort is based on radical as primary order and number of strokes order as secondary order.</td>
</tr>
<tr>
<td>SCHINESEROKE_M</td>
<td>Simplified Chinese sort uses number of strokes as primary order and radical as secondary order.</td>
</tr>
<tr>
<td>SCHINESE_PINYIN_M</td>
<td>Simplified Chinese Pinyin sorting order.</td>
</tr>
<tr>
<td>TCHINESE_RADICAL_M</td>
<td>Traditional Chinese sort based on radical as primary order and number of strokes order as secondary order.</td>
</tr>
<tr>
<td>TCHINESEROKE_M</td>
<td>Traditional Chinese sort uses number of strokes as primary order and radical as secondary order. It supports supplementary characters.</td>
</tr>
</tbody>
</table>
PL/SQL first connection attributes

PL/SQL connection attributes are set by each connection and persist for the duration of the connection. These attributes control the behaviors of the database. PL/SQL first connection attributes are listed Table 2–5, “PL/SQL first connection attributes” and described in detail in this section.
Use of PL/SQL requires a shared memory segment. This shared memory contains recently-executed PL/SQL code, shared package state, and metadata associated with the operation of PL/SQL. This shared memory segment is separate from the one containing the TimesTen database.

This attribute determines the virtual address at which this shared memory segment is loaded into each process that uses the TimesTen "direct" drivers. This memory address must be identical in each process using TimesTen. You must specify the value as a hexadecimal address.

If you do not specify a value for PLSQL_MEMORY_ADDRESS, TimesTen uses a platform-dependent default value.

The default values for each platform are designed to:

1. Maximize the amount of virtual space for your TimesTen database and for your applications.
2. Minimize the fragmentation of the virtual address space.
3. Avoid conflicts with other uses of virtual address space.

The platform specific default memory addresses are:

<table>
<thead>
<tr>
<th>Operating system</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux x86</td>
<td>0000007fa0000000</td>
</tr>
<tr>
<td>AIX</td>
<td>06ffffff00000000</td>
</tr>
<tr>
<td>Windows</td>
<td>000000005b8c0000</td>
</tr>
<tr>
<td>HP-UX</td>
<td>0</td>
</tr>
</tbody>
</table>

Some things to consider when setting this attribute are:

- If applications simultaneously connect to multiple TimesTen databases in direct mode, then each database must use a different value for PLSQL_MEMORY_ADDRESS.
- The value of this attribute is stored persistently by TimesTen. The persistent attribute value is specified in situations when the database is loaded automatically by TimesTen. For example, the database is automatically loaded if RamPolicy for the database is set to 1.
- If the PL/SQL shared memory cannot be mapped at the appropriate address, TimesTen returns an error and the connection to the database fails.
- The memory segment size is determined by the value of PLSQL_MEMORY_SIZE.

**Required privilege**

Only the instance administrator can change the value of this attribute.

**Setting**

Set PLSQL_MEMORY_ADDRESS as follows:
<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>PLSQL_MEMORY_ADDRESS</td>
<td>A hexadecimal value that indicates the memory address for PL/SQL process.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>PL/SQL Memory Address field</td>
<td>A hexadecimal value that indicates the memory address for PL/SQL process.</td>
</tr>
</tbody>
</table>
PLSQL_MEMORY_SIZE

Use of PL/SQL requires a shared memory segment. This attribute determines the size in megabytes of the shared memory segment used by PL/SQL. All connections share this memory segment.

This shared memory contains recently-executed PL/SQL code, the shared package state, and metadata associated with the operation of PL/SQL. This shared memory segment is separate from the one containing the TimesTen database.

Some things to consider when setting this attribute are:

- The value of this attribute is stored persistently by TimesTen. The persistent attribute value is specified in situations when the database is loaded automatically by TimesTen. For example, the database is automatically loaded if RamPolicy for the database is set to 1.

- For most PL/SQL users, the default memory size should be an adequate amount of memory. For databases that make extensive use of PL/SQL, specify a larger memory size. If the memory space is exhausted, ORA-4031 errors may occur during PL/SQL execution.

- The address of the memory segment is determined by the value of PLSQL_MEMORY_ADDRESS.

- There is both a fixed and per connection overhead allocated from the PL/SQL segment, even if you do not use PL/SQL. The minimum fixed memory allocated is approximately 1500 KB. Additionally, approximately 40 KB of memory is allocated per connection. Thus, you can compute an estimated minimum memory setting needed as 1500 KB plus (number_of_connections * 40). If the application uses PL/SQL, we recommend that you allocate twice the estimated minimum required memory for this segment. If the application does not use PL/SQL, you can allocate less than twice the estimated minimum required memory.

Required privilege

Only the instance administrator can change the value of this attribute.

Setting

Set PLSQL_MEMORY_SIZE as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>PLSQL_MEMORY_SIZE</td>
<td>Specify a positive integer greater than 2 representing the size in MB of the shared memory segment in megabytes. The default size is 128 MB.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>PL/SQL Memory Size field</td>
<td>Specify a positive integer greater than 2 representing the size in MB of the shared memory segment in megabytes. The default size is 128 MB.</td>
</tr>
</tbody>
</table>
PL/SQL general connection attributes

PL/SQL general connection attributes are set by each connection and persist for the duration of the connection. These attributes control the behaviors of the database. PL/SQL general connection attributes are listed in Table 2–6, "PL/SQL general connection attributes" and described in detail in this section.

You can use the ALTER SESSION statement, described in Oracle TimesTen In-Memory Database SQL Reference, to change PL/SQL parameters to override the values that are assigned to the PL/SQL general connection attributes at connection time.
PLSCOPE_SETTINGS

PLSCOPE_SETTINGS controls whether the PL/SQL compiler generates cross-reference information. Either all or no cross-references are generated. Possible values are IDENTIFIERS:NONE or IDENTIFIERS:ALL.

The PLSCOPE_SETTINGS connection attribute determines the initial value of this attribute within a session. The value can be modified by an ALTER SESSION statement, described in Oracle TimesTen In-Memory Database SQL Reference. For example:

```
ALTER SESSION SET PLSCOPE_SETTINGS = 'IDENTIFIERS:ALL' ;
```

Required privilege

No privilege is required to change the value of this attribute.

Setting

Set PLSCOPE_SETTINGS as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>PLSCOPE_SETTINGS</td>
<td>IDENTIFIERS:NONE (default)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDENTIFIERS:ALL</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>PLScope settings pulldown list</td>
<td>IDENTIFIERS:NONE (default)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDENTIFIERS:ALL</td>
</tr>
</tbody>
</table>
PLSQL_CCFLAGS

This attribute sets directives to control conditional compilation of PL/SQL units, which enables you to customize the functionality of a PL/SQL program depending on conditions that are checked. This is especially useful when applications are deployed to multiple database environments. Possible uses include activating debugging or tracing features, or basing functionality on the version of the database.

Use this format:

PLSQL_CCFLAGS = 'v1:c1,v2:c2,...,vn:cn'

v1 has the form of an unquoted PL/SQL identifier. It is unrestricted and can be a reserved word or a keyword. The text is insensitive to case. Each one is known as a flag or flag name. Each v1 can occur multiple times in the string, each occurrence can have a different flag value, and the flag values can be of different kinds.

c1 is one of the following: a PL/SQL boolean literal, a PLS_INTEGER literal, or the literal NULL. The text is insensitive to case. Each one is known as a flag value and corresponds to a flag name.

You can use the ALTER SESSION SQL statement, described in Oracle TimesTen In-Memory Database SQL Reference, to change this attribute within a session:

ALTER SESSION SET PLSQL_CCFLAGS = 'v1:c1,v2:c2,...,vn:cn';

Required privilege

No privilege is required to change the value of this attribute.

Setting

Set PLSQL_CCFLAGS as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>PLSQL_CCFLAGS</td>
<td>'A string literal with this format: 'v1:c1,v2:c2,...,vn:cn' Default: null</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>PL/SQL CCFlags field</td>
<td>'A string literal with this format: 'v1:c1,v2:c2,...,vn:cn' Default: null</td>
</tr>
</tbody>
</table>
PLSQL_CONN_MEM_LIMIT

This attribute specifies the *maximum* amount of process heap memory in megabytes that PL/SQL can use for the connection in which it is set.

Some things to consider when setting this attribute are:

- PL/SQL does not allocate this memory until or unless it is needed. Many PL/SQL programs require only a small amount of memory. How you write your application can determine memory requirements. For example, using large VARRAYs in PL/SQL code can require a lot of memory.
- If you attempt to allocate more memory than allowed, TimesTen returns an error.
- The value can be modified with the `ALTER SESSION` statement, described in *Oracle TimesTen In-Memory Database SQL Reference*. For example:

  ```sql
  ALTER SESSION SET PLSQL_CONN_MEM_LIMIT = 100;
  ```

See "PL/SQL connection attributes" in *Oracle TimesTen In-Memory Database PL/SQL Developer’s Guide* for more information.

**Required privilege**

No privilege is required to change the value of this attribute.

**Setting**

Set `PLSQL_CONN_MEM_LIMIT` as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td><strong>PLSQL_CONN_MEM_LIMIT</strong></td>
<td>An integer value in MB. Default value is 100.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td><strong>PL/SQL Connection Memory Limit field</strong></td>
<td>An integer value in MB. Default value is 100.</td>
</tr>
</tbody>
</table>
PL/SQL general connection attributes

PLSQL_OPTIMIZE_LEVEL

This attribute specifies the optimization level to be used to compile PL/SQL library units. The higher the setting of this parameter, the more effort the compiler makes to optimize PL/SQL library units. Possible values are 0, 1, 2, or 3.

The PLSQL_OPTIMIZE_LEVEL connection attribute determines the initial value of this attribute within a session. The value can be modified by an ALTER SESSION statement, described in Oracle TimesTen In-Memory Database SQL Reference. For example:

```
ALTER SESSION SET PLSQL_OPTIMIZE_LEVEL = 2;
```

Required privilege

No privilege is required to change the value of this attribute.

Setting

Set PLSQL_OPTIMIZE_LEVEL as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>PLSQL_OPTIMIZE_LEVEL</td>
<td>The default value is 2.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>PL/SQL Optimization Level pulldown list</td>
<td>The default value is 2.</td>
</tr>
</tbody>
</table>
PLSQL_SESSION_CACHED_CURSORS

This attribute specifies the number of session cursors to cache. A user may adjust the setting to free up space not currently needed in the cache.

PLSQL_SESSION_CACHED_CURSORS can be modified by an ALTER SESSION SQL statement, described in Oracle TimesTen In-Memory Database SQL Reference.

ALTER SESSION SET PLSQL_SESSION_CACHED_CURSORS=25;

Note: This attribute has the same functionality as SESSION_CACHED_CURSORS in Oracle Database.

Required privilege

No privilege is required to change the value of this attribute.

Setting

Set PLSQL_SESSION_CACHED_CURSORS as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>PLSQL_SESSION_CACHED_CURSORS</td>
<td>A positive integer from 0 to 65535 representing the number of cursors to cache. The default value is 50.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>Not supported.</td>
<td></td>
</tr>
</tbody>
</table>

2-92  Oracle TimesTen In-Memory Database Reference
PLSQL_TIMEOUT

This attribute controls how long (in seconds) PL/SQL program units, including PL/SQL procedures, anonymous blocks and functions, are allowed to run before being automatically terminated.

This value can be modified with an ALTER SESSION statement, described in Oracle TimesTen In-Memory Database SQL Reference. If this value is modified through ALTER SESSION, the new value impacts any PL/SQL program units that are currently running. For example:

```
ALTER SESSION SET PLSQL_TIMEOUT = 10;
```

---

Notes:

- See "Choose SQL and PL/SQL timeout values" in Oracle TimesTen In-Memory Database Operations Guide for information about the relationship between TTC_Timeout, SQLQueryTimeout, and PLSQL_TIMEOUT.

- The frequency with which PL/SQL programs check execution time against this timeout value is variable. It is possible for programs to run significantly longer than the timeout value before being terminated.

---

Required privilege

No privilege is required to change the value of this attribute.

Setting

Set PLSQL_TIMEOUT as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>PLSQL_TIMEOUT</td>
<td>A positive integer representing the number of seconds for the timeout value. A value of 0 means that there is no timeout limit. The default value is 30.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>PL/SQL Timeout field</td>
<td>A positive integer representing the number of seconds for the timeout value. A value of 0 means that there is no timeout limit. The default value is 30.</td>
</tr>
</tbody>
</table>

---

See also

TTC_TCP_KEEPALIVE_TIME_MS
TimesTen Cache first connection attributes

TimesTen Cache first connection attributes are used only when you are using the TimesTen Cache product. TimesTen Cache first connection attributes are listed in Table 2–7, "TimesTen Cache first connection attributes" and described in detail in this section.
CacheAWTMethod

Determines whether asynchronous writethrough propagation uses the PL/SQL execution method or SQL array execution method to apply changes to the Oracle database server.

By default, asynchronous writethrough (AWT) uses PL/SQL execution method, CacheAWTMethod=1. AWT bundles all pending operations into a single PL/SQL collection that is sent to the Oracle database server to be executed. This method can improve AWT throughput when there are mixed transactions and network latency between TimesTen and the Oracle database server.

The SQL array execution to apply changes within TimesTen to the Oracle database works well when the same type of operation is repeated. For example, array execution is very efficient when a user does an update that affects several rows of the table. Updates are grouped together and sent to the Oracle database server in one batch.

PL/SQL execution method transparently falls back to array execution mode temporarily when it encounters one of the following:

- A statement that is over 32761 bytes in length.
- A statement that references a column of type BINARY FLOAT, BINARY DOUBLE and VARCHAR of length greater than 4000 bytes.

Specify the SQL execution method, CacheAWTMethod=0, if any AWT cache group contains a VARBINARY column.

The SYSTEMSTATS table contains information about the number of times the execution method temporarily falls back to SQL array execution.

---

Notes:

- This attribute can also be set through the ttDBConfig built-in procedure, which overrides the connection attribute setting. See “ttDBConfig” on page 3-103.
- Use the same AWT execution method on all TimesTen nodes in any active standby pair replication scheme.

---

**Required privilege**

Only the instance administrator can change the value of this attribute.

**Setting**

Set CacheAWTMethod as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>CacheAWTMethod</td>
<td>0 - Use SQL array execution method. 1 (default) - Use PL/SQL collections and anonymous blocks (PL/SQL execution method).</td>
</tr>
<tr>
<td>Where to set the attribute</td>
<td>How the attribute is represented</td>
<td>Setting</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>----------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>Cache AWT Method field</td>
<td>0 - Use SQL array execution method.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 (default) - Use PL/SQL collections and anonymous blocks (PL/SQL execution method).</td>
</tr>
</tbody>
</table>
TimesTen Cache database attributes

TimesTen Cache connection attributes are used only when you are using the TimesTen Cache product. TimesTen Cache data store attributes are listed and described in detail in this section.
CacheAWTParallelism

CacheAWTParallelism indicates the number of threads that apply changes to the Oracle database. This attribute has a relationship to ReplicationParallelism and ReplicationApplyOrdering.

If you do not set this attribute or if you set it to the default value of 1, the number of threads that apply changes to the Oracle database is twice the setting for ReplicationParallelism to the maximum value of 31.

If both ReplicationParallelism and CacheAWTParallelism attributes are set, the value set in CacheAWTParallelism configures the number of threads used for parallel propagation. The setting for CacheAWTParallelism determines the number of apply threads for parallel propagation and the setting for ReplicationParallelism determines the number of threads for parallel replication.

CacheAWTParallelism only has an affect when there are AWT cache groups.

To learn more about parallel AWT caching, see "Configuring parallel propagation to Oracle Database tables" in the Oracle TimesTen Application-Tier Database Cache User’s Guide.

Required privilege

Only the instance administrator can change the value of this attribute.

Setting

Set CacheAWTParallelism as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>CacheAWTParallelism</td>
<td>( n ) - An integer between 1 and 31 that indicates the number of threads that apply changes to the Oracle database. The default is 1.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>Cache AWT Parallelism field</td>
<td>( n ) - An integer between 1 and 31 that indicates the number of threads that apply changes to the Oracle database. The default is 1.</td>
</tr>
</tbody>
</table>
TimesTen Cache general connection attributes

TimesTen Cache general connection attributes are used only when you are using the TimesTen Cache product. TimesTen Cache general connection attributes are listed in Table 2–9, "TimesTen Cache general connection attributes" and described in detail in this section.
DynamicLoadEnable

This attribute enables or disables dynamic load of data from an Oracle database to a TimesTen dynamic cache group. By default, dynamic load of data from an Oracle database is enabled.

To enable or disable dynamic load at the statement level and temporarily override the setting of this attribute, set the `DynamicLoadEnable` optimizer flag with the `ttOptSetFlag` built-in procedure or using the statement level optimizer hint `TT_DynamicLoadEnable` in a SQL statement.

**Note:** The value of this attribute overrides the dynamic load behavior of all dynamic cache groups for the current connection to the database.

**Required privilege**

No privilege is required to change the value of this attribute.

**Setting**

Set `DynamicLoadEnable` as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
</table>
| C or Java programs or UNIX and Linux systems odbc.ini file | `DynamicLoadEnable` | **0** - Disables dynamic load of data from an Oracle database to TimesTen dynamic cache groups for the current connection.  
**1** (default) - Enables dynamic load of data from an Oracle database to TimesTen dynamic cache groups for the current connection. |
| Windows ODBC Data Source Administrator | **Dynamic Load Enable** field | **0** - Disables dynamic load of data from an Oracle database to TimesTen dynamic cache groups for the current connection.  
**1** (default) - Enables dynamic load of data from an Oracle database to TimesTen dynamic cache groups for the current connection. |
DynamicLoadErrorMode

This attribute controls what happens when an application executes a SQL operation against a dynamic cache group and the SQL operation cannot use dynamic load.

With a value of 0, the SQL operation executes against whatever data is in the TimesTen cache tables and returns a result based on that data with no error indicated.

With a value of 1, any statement that cannot use dynamic load (even if it does not need dynamic load) fails with an error indicating that it is not dynamic load-compliant.

For more information on caching data from an Oracle database in a TimesTen cache group, see Oracle TimesTen Application-Tier Database Cache User’s Guide.

---

**Note:** To override the value of this attribute at the statement level, set the DynamicLoadErrorMode optimizer flag with the ttOptSetFlag built-in procedure or using the statement level optimizer hint TT_DynamicLoadErrorMode in a SQL statement.

For details, see "Statement level optimizer hints" in the Oracle TimesTen In-Memory Database SQL Reference.

---

**Required privilege**

No privilege is required to change the value of this attribute.

**Setting**

Set DynamicLoadErrorMode as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>DynamicLoadErrorMode</td>
<td>0 (default) - Statements execute against the cached data with no error. 1 - Statements use dynamic load or fail with an error.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator field</td>
<td>DynamicLoadErrorMode</td>
<td>0 (default) - Statements execute against the cached data with no error. 1 - Statements use dynamic load or fail with an error.</td>
</tr>
</tbody>
</table>
OracleNetServiceName

The TimesTen Cache uses the OracleNetServiceName attribute. This attribute identifies the Service Name for the Oracle instance. To cache Oracle database tables and enable communication with the Oracle database, you must specify an Oracle Service Name.

Required privilege

No privilege is required to change the value of this attribute.

Setting

Set OracleNetServiceName as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>OracleNetServiceName</td>
<td>Character string specifying the Oracle Service Name that is to be used as the Oracle ID.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>OracleNetServiceName field</td>
<td>Character string specifying the Oracle Service Name that is to be used as the Oracle ID.</td>
</tr>
</tbody>
</table>
OraclePWD

The TimesTen Cache uses the OraclePWD attribute.
The value of this attribute is the password for the user specified by UID to connect to the Oracle database to perform cache operations.

Required privilege

No privilege is required to set the value of this attribute.

Setting

This attribute must be set in the connection string. On Linux, suppose you have defined the following odbc.ini file:

```
[myDSN]
Datastore=/data/myDSN
PermSize=128
DatabaseCharacterSet=AL32UTF8
ConnectionCharacterSet=AL32UTF8
```

Set OraclePWD for user ttuser by connecting to myDSN as follows:

```
% ttisql
```

Copyright (c) 1996-2011, Oracle. All rights reserved.
Type ? or ‘help’ for help, type ‘exit’ to quit ttisql.

```
Command> connect "dsn=myDSN;OraclePWD=mypwd";
Connection successful:
DSN=beta4;UID=ttuser;DataStore=/data/myDSN;DatabaseCharacterSet=AL32UTF8;
ConnectionCharacterSet=AL32UTF8;PermSize=128;
(Default setting AutoCommit=1)
```

On Windows, set OraclePWD in the connection string in the same way that it is set on Linux.

See also

UID and PWD
PassThrough

The TimesTen Cache uses the PassThrough attribute.

It specifies which SQL statements are executed only in the cache database and which SQL statements are passed through to the Oracle database. For more information about the TimesTen Cache, see Oracle TimesTen Application-Tier Database Cache User’s Guide and “CREATE CACHE GROUP” in Oracle TimesTen In-Memory Database SQL Reference.

The execution of a prepared PassThrough command assumes that the schema of dependent objects in the Oracle database has not changed since the prepare. If the schema has changed the PassThrough command may cause unexpected results from the Oracle database.

When passing SQL statements through to the Oracle database, use only TimesTen supported data types in column definitions. If the specified data type is not supported in TimesTen, the passthrough statement fails.

For information on changing the isolation level on the Oracle database connection, when using this attribute, see "Isolation" on page 2-61.

Required privilege

No privilege is required to change the value of this attribute.

Setting

Set PassThrough as follows.

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc . ini file</td>
<td>PassThrough</td>
<td>0 (default) - SQL statements are executed only on TimesTen. 1 - INSERT, UPDATE and DELETE statements are executed on TimesTen unless they reference one or more tables that are not in TimesTen. If they reference one or more tables not in TimesTen, they are passed through to the Oracle database. DDL statements are executed on TimesTen. Other statements are passed through to the Oracle database if they generate a syntax error in TimesTen or if one or more tables referenced within the statement are not in TimesTen. 2 - INSERT, UPDATE and DELETE statements performed on tables in read-only cache groups or user managed cache groups with the READONLY cache table attribute are passed through to the Oracle database. Passthrough behavior for other cache group types is the same as PassThrough=1. 3 - All statements are passed through to the Oracle database for execution.</td>
</tr>
</tbody>
</table>
Restrictions

Certain restrictions must be considered when using the passthrough feature. They include:

- If the PassThrough attribute is set so that a query must be executed in the Oracle database, the query is sent to the Oracle database without any changes. If the query uses a synonym for a table in a cache group, then a synonym with the same name must be defined for the corresponding Oracle database table for the query to be successful.

- In the case that a SQL statement that uses TimesTen only syntax is passed through to the Oracle database, TimesTen returns an error message that indicates the syntax is not supported in the Oracle database.

- Execution of a prepared passthrough command assumes that the schema of dependent objects in the Oracle database have not changed after the prepare. If the schema has changed, unexpected results can occur.

- TimesTen does not include a cache invalidation feature. TimesTen does not verify that the cached tables are up to date. When a query is syntactically correct in TimesTen and the cache contains all the tables referenced in the query, the query is executed in TimesTen regardless of whether the cache is up to date.

- The passthrough of Oracle INSERT, UPDATE, or DELETE operations depends on the setting of the PassThrough attribute as described in the table above. TimesTen Cache cannot detect INSERT, UPDATE and DELETE operations that are hidden in a trigger or stored procedure. Therefore, TimesTen cannot enforce the passthrough rule on hidden operations.

- You cannot pass PL/SQL blocks through to the Oracle database.
- The effects of a passthrough `INSERT`, `UPDATE`, or `DELETE` operation on a read-only cache group are only seen after the transaction is committed and after the next autorefresh operation is completed.

- There is no mechanism to detect or block updates on an Oracle database table that is cached in a TimesTen synchronous writethrough cache group. Whether the updates are made by statements passed through the cache or from other Oracle database applications, the changes are never reflected in TimesTen Cache.

- Oracle Call Interface (OCI) does not support a mechanism to describe the binding type of the input parameters. Ensure that your application supplies the correct SQL types for passthrough statements. The ODBC driver converts the C and SQL types and presents the converted data and the SQL type code to TimesTen. TimesTen presents the information to OCI. The length of the input binding values is restricted to 4000 for `LONG` and `LONG RAW` types.

- At all passthrough levels, passthrough execution of DDL statements does not result in commits on the TimesTen side.

- A transaction that contains operations that are replicated with `RETURN TWOSAFE` cannot have a `PassThrough` setting greater than 0. If `PassThrough` is greater than 0, an error is returned and the transaction must be rolled back.

- When `PassThrough` is set to 0, 1, or 2, the following behavior occurs when a dynamic load condition exists:
  - A dynamic load can occur for a `SELECT` operation on cache tables in any dynamic cache group type.
  - A dynamic load for an `INSERT`, `UPDATE`, or `DELETE` operation can only occur on cached tables with dynamic asynchronous or synchronous writethrough cache groups.

Refer to "SQL Statements" in Oracle TimesTen In-Memory Database SQL Reference for details about the `INSERT`, `UPDATE`, `DELETE`, and `SELECT` statements.
RACCallback

This attribute enables you to enable or disable the installation of Transparent Application Failover (TAF) and Fast Application Notification (FAN) callbacks when using Oracle Real Application Clusters (Oracle RAC) with TimesTen Cache.

For more information about TimesTen Cache, see Oracle TimesTen Application-Tier Database Cache User’s Guide and "CREATE CACHE GROUP" in Oracle TimesTen In-Memory Database SQL Reference.

Required privilege

No privilege is required to change the value of this attribute.

Setting

Set RACCallback as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
</table>
| C or Java programs or UNIX and Linux systems odbc.ini file | RACCallback | 0 - Do not install TAF and FAN callbacks.  
1 (default) - Install the TAF and FAN callbacks. |
| Windows ODBC Data Source Administrator | RACCallback check box | unchecked - Do not install TAF and FAN callbacks.  
checked (default) - Install the TAF and FAN callbacks. |
TimesTen Client connection attributes

TimesTen Client connection attributes are used only when you are connecting to a TimesTen server from a TimesTen client application. TimesTen Client connection attributes are listed in Table 2–10, "TimesTen Client connection attributes" and described in detail in this section.

In addition to the attributes listed in this section, some database attributes and general connection attributes are also available for client connections or impact the behavior of the connection. These attributes are:

- ConnectionCharacterSet
- ConnectionName
- UID and PWD

To view the value of a client attribute:

- In ODBC 3.5, use the ODBC function SQLGetConnectAttr. To learn more about this function, see "Attribute support for ODBC 3.5 SQLSetConnectAttr and SQLGetConnectAttr" in Oracle TimesTen In-Memory Database C Developer’s Guide
- In ODBC 2.5, use the ODBC function SQLGetConnectOption. To learn more about this function, see "Option support for ODBC 2.5 SQLSetConnectOption and SQLGetConnectOption" section of the Oracle TimesTen In-Memory Database C Developer’s Guide.
TCP_Port

When connecting to a TimesTen database using the TimesTen Client and Server, the TimesTen Client requires the network address and the TCP port number of the computer running the TimesTen Server. As a convenience, TimesTen enables you to define a logical server name that contains the network address and port number pair.

If you specify anything other than a logical server name for the `TTC_Server` or `TTC_Server1` attribute in the Client DSN definition, TimesTen Client assumes that the Server is running on the default TCP/IP port number. In such cases, if your Server is running on a port other than the default port, you must specify the port number in the ODBC connection string. For example:

```
*TTC_SERVER=server_host_name;
TTC_SERVER_DSN=Server_DSN;TCP_PORT=server_port"
```

or

```
*DSN=Client_DSN;TCP_Port=server_port"
```

**Required privilege**

No privilege is required to change the value of this attribute.

**Setting**

Set `TCP_Port` as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs</td>
<td><code>TCP_Port</code></td>
<td>Specify the port number where the Server is listening.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator and UNIX and Linux systems <code>ttconnect.ini</code> file</td>
<td><code>TCP_Port</code></td>
<td>Specify the port number where the Server is listening.</td>
</tr>
</tbody>
</table>
TCP_Port2

For TimesTen Classic, TimesTen uses this attribute to specify the port number to use if an automatic failover occurs. (This is unnecessary for TimesTen Scaleout.) See the description of TCP_Port for details on setting the value of this attribute and associated attributes.

See "Using automatic client failover" in Oracle TimesTen In-Memory Database Operations Guide for more information on automatic client failover.

Required privilege

No privilege is required to change the value of this attribute.

Setting

Set TCP_Port2 as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs</td>
<td>TCP_Port2</td>
<td>Specify the failover port number where the Server should listen.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator and UNIX and Linux systems ttconnect.ini file.</td>
<td>TCP_Port2.</td>
<td>Specify the failover port number where the Server should listen.</td>
</tr>
</tbody>
</table>
**TTC_ConnectTimeout**

For client/server, this specifies the integer number of seconds the client will wait for a `SQLDriverConnect` or `SQLDisconnect` request. It overrides the value of `TTC_Timeout` for those requests.

A value of 0 means there is no timeout. A negative value defers to the `TTC_Timeout` setting. As with `TTC_Timeout`, if the timeout is reached, the connection and the associated socket are closed without a call to `SQLDisconnect`.

`TTC_ConnectTimeout` can be set in either the client connection string or the client DSN.

Also see "TTC_Timeout" on page 2-126.

**Required privilege**

No privilege is required to change the value of this attribute.

**Setting**

Set `TTC_ConnectTimeout` as follows.

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems ODBC.INI file</td>
<td><code>TTC_ConnectTimeout</code></td>
<td>Seconds to wait for a client connect or disconnect request.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

Connection Attributes 2-111
**TTC_FailoverPortRange**

Specifies a port range for the port that the automatic client failover thread listens on for failover notifications in an active/standby replication configuration. The failover configuration enables a client application to connect to a new active node automatically if there is a failure on the current node.

Specifying a port range helps accommodate firewalls between the client and server systems. By default, TimesTen uses a port chosen by the operating system.

See "Using automatic client failover" in *Oracle TimesTen In-Memory Database Operations Guide* for more information on automatic client failover.

**Required privilege**

No privilege is required to change the value of this attribute.

**Setting**

Set TTC_FailoverPortRange as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>TTC_FailoverPortRange</td>
<td>Specify a lower value and an upper value for the port numbers in the format lowervalue—uppervalue.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>Failover Port Range field</td>
<td>Specify a lower value and an upper value for the port numbers in the format lowervalue—uppervalue.</td>
</tr>
</tbody>
</table>
TTC_NoReconnectOnFailover

Specifies whether the TimesTen client should not automatically reconnect to the server after a failover. If this is set to 1 (enabled), TimesTen is instructed to do all the usual client failover processing except for the reconnect. (For example, statement and connection handles are marked as invalid.) This is useful if the application does its own connection pooling or manages its own reconnection to the database after failover. The default value is 0 (reconnect).

**Note:** Client failover is supported only when the client is part of an active/standby pair replication configurations.


**Required privilege**

No privilege is required to change the value of this attribute.

**Setting**

Set TTC_NoReconnectOnFailover as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>TTC_NoReconnectOnFailover</td>
<td>0 = Client reconnects to server after failover (default).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = Client does all the failover processing, but does not reconnect after failover.</td>
</tr>
</tbody>
</table>

| Windows ODBC Data Source Administrator     | No reconnect on failover field    | 0 = Client reconnects to server after failover (default).              |
|                                            |                                  | 1 = Client does all the failover processing, but does not reconnect after failover. |
**TTC_Random_Selection**

For TimesTen Scaleout, specifies that the TimesTen client, if necessary, randomly selects an alternative server from the list provided in TTC_ServerN attribute settings.

If the client cannot connect to the selected server, it keeps redirecting until it successfully connects to one of the listed servers. If the client cannot connect to any of the selected servers, TimesTen returns an error.

See “Using automatic client failover” in Oracle TimesTen In-Memory Database Operations Guide for more information on automatic client failover.

**Required privilege**

No privilege is required to change the value of this attribute.

**Setting**

Set TTC_Random_Selection as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>TTC_Random_Selection</td>
<td>0 = Client selects the first server specified by the TTC_ServerN attributes. 1 (default) = Client selects the server randomly from the list of servers provided by TTC_ServerN attributes.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>Random selection field</td>
<td>0 = Client selects the first server specified by the TTC_ServerN attributes. 1 (default) = Client selects the server randomly from the list of servers provided by TTC_ServerN attributes.</td>
</tr>
</tbody>
</table>
**TTC_REDIRECT**

For TimesTen Scaleout, TTC_REDIRECT defines how a client is redirected. If this is set to 0 and the initial connection attempt to the desired data instance fails, then an error is returned and there are no further connection attempts. This does not affect subsequent failovers on that connection.

Automatic redirection: By default, this connection attribute is set to 1 so that a client connection is automatically redirected to any available data instance within the grid if the current host is busy or unavailable. The connection is redirected to the host with the fewest number of client connections.

Elements within a single replica set: If you want the client to connect to elements within a single replica set (because the data you are interested in is contained within this replica set), then set the TTC_REDIRECT connection attribute to 0. Then, the client connects only to the host indicated by the DSN or to the host with the element in the same replica set. If the connection is rejected, then a connection error is returned.

The TTC_Redirect_Limit attribute limits how many times the client is redirected. The number of hosts in your grid can be of a size that you want to limit the number of redirected client connection attempts for performance reasons. You can set the TTC_Redirect_Limit connection attribute to the number of connection redirection attempts. For example, setting TTC_Redirect_Limit=10 limits the number of client connection redirection attempts to other hosts to 10 attempts. If the client does not connect within this number of attempts, a connection error is returned.

**Required privilege**

No privilege is required to change the value of this attribute.

**Setting**

Set TTC_REDIRECT as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>TTC_REDIRECT</td>
<td>1 (default) - redirect to any available data instance 0 - error if redirection to specified data instance fails</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>N/A</td>
<td>.</td>
</tr>
</tbody>
</table>
**TTC_Redirect_Limit**

For TimesTen Scaleout, **TTC_Redirect_Limit** limits how many times the client is redirected.

This is useful if the number of hosts in your grid is such that you want to limit the number of redirected client connection attempts for performance reasons.

For example, setting **TTC_Redirect_Limit=10** limits the number of client connection redirection attempts to other hosts to 10 attempts. If the client does not connect within this number of attempts, a connection error is returned.

---

**Note:** There is no setting for no limit, but you can set it to a very large integer.

---

**Required privilege**

No privilege is required to change the value of this attribute.

---

**Setting**

Set **TTC_Redirect_Limit** as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td><strong>TTC_Redirect_Limit</strong></td>
<td>Integer to limit how many times the client is redirected. The default is 1.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>N/A</td>
<td>.</td>
</tr>
</tbody>
</table>
TTC_RollbackRequiredOnFailover

Indicates if the application must roll back if a client failover occurs. The default setting of 1 (enabled) results in a "transaction must roll back" error on the connection handle and all associated statement handles if failover occurs in the middle of a transaction. This explicitly indicates that the SQL operation may have failed. The application must roll back the transaction before proceeding. The setting of 0 results in no indication of possible failure. Use the setting of 0 only if necessary for backward compatibility.

Required privilege

No privilege is required to change the value of this attribute.

Setting

Set TTC_RollbackRequiredOnFailover as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>TTC_RollbackRequiredOnFailover</td>
<td>1 to require rollback</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>Rollback required on failover field</td>
<td>1 to require rollback.</td>
</tr>
</tbody>
</table>
### TTC_Server or TTC_Server1

When connecting to a TimesTen database using the TimesTen Client and Server, the TimesTen Client requires the specification of the network address and TCP port number of the computer running the TimesTen Server. As a convenience, TimesTen enables you to define a logical server name that contains the network address and port number pair. If you specify anything other than a logical server name for this attribute, TimesTen Client assumes that the Server is running on the default TCP/IP port number. In such cases, if your Server is running on a port other than the default port, you must specify the port number in the ODBC connection string. For example:

```
*TTC_SERVER=server_host_name;
TTC_SERVER_DSN=Server_DSN;TCP_PORT=server_port
```

Or:

```
*DSN=Client_DSN;TCP_Port=server_port
```

Once the logical server name is defined, you can use that name as the value for the `TTC_Server` attribute in a Client DSN. Multiple Client DSNs referencing the same computer that is running the TimesTen Server can use the same logical server name for the value of the `TTC_Server` attribute instead of having to specify repeatedly the same network address and port number within each of the Client DSNs.

---

**Notes:**

- You can use `TTC_Server1` for TimesTen Scaleout. It is interpreted the same as `TTC_Server`.
- TimesTen recommends that you specify a logical server name for the `TTC_Server` attribute. However, you can also specify a domain name server (DNS), host name or IP address for the `TTC_Server` attribute. If you do not use a logical server name and the TimesTen Server is listening on a nondefault port number, you must provide the port number in the ODBC connection string. For example:

```
*TTC_SERVER=server_host_name;TTC_SERVER_DSN=Server_DSN;
TCP_PORT=server_port
```

Or:

```
*DSN=Client_DSN;TCP_Port=server_port
```

---

**Required privilege**

No privilege is required to change the value of this attribute.

**Setting**

Set `TTC_Server` or `TTC_Server1` as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>TTC_Server, TTC_Server1</td>
<td>Character string specifying the logical server.</td>
</tr>
<tr>
<td>Where to set the attribute</td>
<td>How the attribute is represented</td>
<td>Setting</td>
</tr>
<tr>
<td>----------------------------</td>
<td>----------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td><strong>Server Name or Network Address field</strong></td>
<td>Character string specifying the logical server.</td>
</tr>
</tbody>
</table>
**TTC_Server2, TTC_ServerN**

This attribute specifies the logical server name to use if an automatic failover occurs. See the description of `TTC_Server` or `TTC_Server1` for details on setting the value of this attribute and associated attributes. Additional attributes where \( \text{N} > 2 \) are supported for TimesTen Scaleout, so that you can specify a list of servers.

The value of this attribute can be the same as the value specified for `TTC_Server` or `TTC_Server1` if it is a virtual IP address.

For TimesTen Classic, if the client has already failed over and has connected to `TTC_Server2` and the connection fails, it connects to `TTC_Server`. It alternately attempts to connect to `TTC_Server` and `TTC_Server2` until the `TTC_TIMEOUT` attribute expires.

For TimesTen Scaleout, TimesTen can go down a list of servers, as necessary, that you provide with `TTC_Server2`, `TTC_Server3`, `TTC_Server4`, and so on.

See "Using automatic client failover" in *Oracle TimesTen In-Memory Database Operations Guide* for more information on automatic client failover.

**Required privilege**

No privilege is required to change the value of this attribute.

**Setting**

Set `TTC_Server2` or `TTC_ServerN` as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and</td>
<td><code>TTC_Server2, TTC_ServerN</code></td>
<td>Character string specifying the logical server to be used if an automatic failover occurs. The maximum number of servers that the client can specify is 999.</td>
</tr>
<tr>
<td>Linux systems odbc.ini file</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows ODBC Data Source</td>
<td><code>Server Name or Network Address</code></td>
<td>Character string specifying the logical server to be used if an automatic failover occurs. The maximum number of servers that the client can specify is 999.</td>
</tr>
<tr>
<td>Administrator</td>
<td>2 field</td>
<td></td>
</tr>
</tbody>
</table>
**TTC_Server_DSN**

The **TTC_Server_DSN** attribute specifies a Server DSN on the computer running the TimesTen Server.

On Windows, Server DSNs are the set of TimesTen System DSNs that use the TimesTen Data Manager driver. Use the ODBC Data Source Administrator to define Server DSNs.

On UNIX and Linux systems, Server DSNs are defined in the `/var/TimesTen/instance/sys.odbc.ini` file. More details on this topic can be found in the platform-specific sections.

**Required privilege**

No privilege is required to change the value of this attribute.

**Setting**

Set **TTC_Server_DSN** as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>TTC_Server_DSN</td>
<td>Character string specifying the DSN that resides on the Server.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>Server DSN field</td>
<td>Character string specifying the DSN that resides on the Server.</td>
</tr>
</tbody>
</table>
TTC_Server_DSN2

For TimesTen Classic, this attribute specifies the Server DSN on the computer running the TimesTen Server. (This is unnecessary for TimesTen Scaleout.) This is the Server DSN to be used if an automatic failover occurs. See the description of TTC_Server_DSN for details on setting the value of this attribute and associated attributes.

If a failover occurs, if the client cannot connect to TTC_Server_DSN or loses the connection to the DSN, it attempts to connect to TTC_Server_DSN2.

See "Using automatic client failover" in Oracle TimesTen In-Memory Database Operations Guide for more information on automatic client failover.

Required privilege

No privilege is required to change the value of this attribute.

Setting

Set TTC_Server_DSN2 as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems ODBC.INI file</td>
<td>TTC_Server_DSN2</td>
<td>Character string specifying the DSN that resides on the Server to be used if an automatic failover occurs.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>Server DSN2 field</td>
<td>Character string specifying the DSN that resides on the Server to be used if an automatic failover occurs.</td>
</tr>
</tbody>
</table>
TTC_TCP_KEEPALIVE_INTVL_MS

The TTC_TCP_KEEPALIVE_INTVL_MS attribute sets the time interval (in milliseconds) between subsequential probes.

By default, if the connection fails, TimesTen Scaleout sends the client connection to another active server. Part of the method to see if the connection is up or if it has failed is to check the TCP socket. When a TCP connection is started, a set of timers are associated with the connection. These timers indicate when TimesTen Scaleout checks the TCP socket to determine whether the connection is up or if it has failed.

Required privilege

No privilege is required to change the value of this attribute.

Setting

Set TTC_TCP_KEEPALIVE_INTVL_MS as follows.

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems</td>
<td>TTC_TCP_KEEPALIVE_INTVL_MS</td>
<td>A positive integer value. Default is 1000.</td>
</tr>
<tr>
<td>ODBC.INI file</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>KeepAlive Interval field</td>
<td>A positive integer value. Default is 1000.</td>
</tr>
</tbody>
</table>
TTC_TCP_KEEPALIVE_PROBES

The TTC_TCP_KEEPALIVE_PROBES attribute sets the number of unacknowledged probes to send before considering the connection as failed and notifying the client.

By default, if the connection fails, TimesTen Scaleout sends the client connection to another active server. Part of the method to see if the connection is up or if it has failed is to check the TCP socket. When a TCP connection is started, a set of timers are associated with the connection. These timers indicate when TimesTen Scaleout checks the TCP socket to determine whether the connection is up or if it has failed.

Required privilege

No privilege is required to change the value of this attribute.

Setting

Set TTC_TCP_KEEPALIVE_PROBES as follows.

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems ODBC.INI file</td>
<td>TTC_TCP_KEEPALIVE_PROBES</td>
<td>A positive integer value. Default is 2.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator KeepAlive Probes field</td>
<td></td>
<td>A positive integer value. Default is 2.</td>
</tr>
</tbody>
</table>
TTC_TCP_KEEPALIVE_TIME_MS

The TTC_TCP_KEEPALIVE_TIME_MS attribute sets the duration time (in milliseconds) between the last data packet sent and the first probe.

By default, if the connection fails, TimesTen Scaleout sends the client connection to another active server. Part of the method to see if the connection is up or if it has failed is to check the TCP socket. When a TCP connection is started, a set of timers are associated with the connection. These timers indicate when TimesTen Scaleout checks the TCP socket to determine whether the connection is up or if it has failed.

Required privilege

No privilege is required to change the value of this attribute.

Setting

Set TTC_TCP_KEEPALIVE_TIME_MS as follows.

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems ODBC.INI file</td>
<td>TTC_TCP_KEEPALIVE_TIME_MS</td>
<td>A positive integer value. Default is 1000.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>KeepAlive Time field</td>
<td>A positive integer value. Default is 1000.</td>
</tr>
</tbody>
</table>
TTC_Timeout

The **TTC_Timeout** attribute sets a maximum time limit, in seconds, for a network operation that is completed by using the TimesTen Client and Server. The **TTC_Timeout** attribute also determines the maximum number of seconds a TimesTen Client application waits for the result from the corresponding TimesTen Server process before timing out. For example, if the Client application is running long queries, you may want to increase the timeout interval.

The operating system `select()` call on the client side of a client/server connection uses the value of **TTC_Timeout**. The `SQLExecute()` and `OCIStmExecute()` functions do not.

A value of 0 indicates that client/server operations should not timeout. If this attribute is not set, the default timeout period is 60 seconds. The maximum timeout period is 99,999 seconds. Upon timeout, the operation is interrupted, the Client application receives a timeout error and the connection is terminated and socket closed (without a call to `SQLDisconnect`).

For active standby pair failover scenarios, the minimum value is 60 seconds.

The timeout value can be set after establishing a connection by calling the `ttIsql` `clienttimeout` command. When the query timeout is set after establishing a connection to the database, the client driver returns an error if the network timeout value is greater than 0, and the query timeout value greater than or equal to the network timeout value. The SQLState is set to S1000.

This attribute is not supported (the setting ignored) when shared memory is used for Client/Server inter-process communication.

See "Choose SQL and PL/SQL timeout values" in *Oracle TimesTen In-Memory Database Operations Guide* for information about the relationship between **TTC_Timeout**, `SQLQueryTimeout`, and `PLSQL_TIMEOUT`.

**TTC_Timeout** can be overridden for connect and disconnect requests by **TTC_ConnectTimeout**. See "**TTC_ConnectTimeout**" on page 2-111.

**Required privilege**

No privilege is required to change the value of this attribute.

**Setting**

Set **TTC_Timeout** as follows.

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems ODBC.INI file</td>
<td><strong>TTC_Timeout</strong></td>
<td>A value between 0 and 99999 that represents the number of seconds that the TimesTen Client waits for an operation to complete before timing out. (The default value is 60.) In an active standby pair failover scenario, the minimum value is 60.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td><strong>Timeout Interval</strong> field</td>
<td>A value between 0 and 99999 that represents the number of seconds that the TimesTen Client waits for an operation to complete before timing out. (The default value is 60.) In an active standby pair failover scenario, the minimum value is 60.</td>
</tr>
</tbody>
</table>
Server connection attributes are specified in the Server DSN only and are read at first connection. See “Defining server DSNs on a TimesTen Server system” in Oracle TimesTen In-Memory Database Operations Guide. Use these attributes to set the number of connections to a TimesTen server, the number of servers for each DSN and the size of each connection to the server. These attributes allow you to specify multiple client connections to a single Server. By default, TimesTen creates only one connection to a Server per child process.

Note: These attributes must be specified in the DSN. If these attributes are specified in a connection string, TimesTen ignores them and their values.

There are also TimesTen main daemon options that can specify multiple Server connections. In the case that both the daemon options and these attributes have been specified, the value of the attributes takes precedence.

Server connection attributes are listed in Table 2–11, "TimesTen Server connection attributes" and described in detail in this section.
MaxConnsPerServer

The MaxConnsPerServer attribute sets the maximum number of concurrent connections to the server which the DSN references.

If you want to support many connections to the Server, you must ensure that the per-process file descriptor limit for the UID that TimesTen is being run as is set to a value somewhat more than the number of concurrent child servers that are active. This is the number of anticipated concurrent client connections divided by MaxConnsPerServer. For full details on MaxConnsPerServer, see "Server DSN connection attributes defined in odbc.ini file" in the Oracle TimesTen In-Memory Database Operations Guide.

The value of this attribute takes precedence over the setting of the value of the max_conns_per_server attribute in the timesten.conf file. For details, see "Specifying multiple connections to the TimesTen Server" in Oracle TimesTen In-Memory Database Operations Guide.

For limits on the maximum number of connections to a TimesTen database, see Chapter 6, "System Limits".

Changes to TimesTen Server settings do not occur until the TimesTen server is restarted. To restart the Server, use the command ttDaemonAdmin -restartserver.

Required privilege

Only a user with operating system privileges on the system DSN in which this attribute is defined can change the value of this attribute to a value other than the one currently in effect.

Setting

Set MaxConnsPerServer as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>MaxConnsPerServer</td>
<td>A value between 1 and 2047. The default is 1, which indicates that each connection has its own separate process, not just a separate thread within a process.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>Maximum Connections Per Server Process field</td>
<td>A value between 1 and 2047. The default is 1, which indicates that each connection has its own separate process, not just a separate thread within a process.</td>
</tr>
</tbody>
</table>

For limits on the maximum number of connections to a TimesTen database, see Chapter 6, "System Limits".
ServersPerDSN

The ServersPerDSN attribute specifies the number of child server processes for a particular server DSN that will use round-robin connection distribution.

This attribute only has any effect if the TimesTen server is configured to operate in multithreaded mode (\(\text{MaxConnsPerServer} > 1\)). If ServersPerDSN is set to 1 then the first \(\text{MaxConnsPerServer}\) client connections to the server DSN will be assigned to one child server process, the next \(\text{MaxConnsPerServer}\) connections to a second child server process and so on. See "Server DSN connection attributes defined in odbc.ini file" in the Oracle TimesTen In-Memory Database Operations Guide for more details.

The value of this attribute takes precedence over the setting of the value of the servers_per_DSN attribute in the timesten.conf file. For details, see "Specifying multiple connections to the TimesTen Server" in the Oracle TimesTen In-Memory Database Operations Guide.

Changes to TimesTen Server settings do not occur until the TimesTen server is restarted. To restart the Server, use the command \texttt{ttDaemonAdmin -restartserver}.

Required privilege

Only a user with operating system privileges on the system DSN in which this attribute is defined can change the value of this attribute to a value other than the one currently in effect.

Setting

Set ServersPerDSN as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td>ServersPerDSN</td>
<td>A value between 1 and 2047. The default is 1.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td>Server Processes Per DSN field</td>
<td>A value between 1 and 2047. The default is 1.</td>
</tr>
</tbody>
</table>
ServerStackSize

The `ServerStackSize` attribute value determines the size of the stack on the Server for each connection. The value of this attribute is only meaningful if the value of `MaxConnsPerServer` is greater than one. If there is only one connection per Server, the child server uses the process’ main stack. It is also platform-dependent, as defined in the setting below.

You generally should not need to set the `ServerStackSize` attribute. However, if the `ttcserver` process is getting repeatable Access Violations (Windows) or core dumps (Linux and Unix), you may consider increasing the `ServerStackSize` attribute to 1024 KB or greater.

This value of this attribute takes precedence over the setting of the `server_stack_size` attribute in the `timesten.conf` file. For details, see "Specifying multiple connections to the TimesTen Server" in Oracle TimesTen In-Memory Database Operations Guide.

Changes to TimesTen Server settings do not occur until the TimesTen server is restarted. To restart the Server, use the command `ttDaemonAdmin -restartserver`.

Required privilege

Only a user with operating system privileges on the system DSN in which this attribute is defined can change the value of this attribute to a value other than the one currently in effect.

Setting

Set `ServerStackSize` as follows:

<table>
<thead>
<tr>
<th>Where to set the attribute</th>
<th>How the attribute is represented</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>C or Java programs or UNIX and Linux systems odbc.ini file</td>
<td><code>ServerStackSize</code></td>
<td>Valid values depend on the platform. The default is 768 KB. If the <code>sysconf</code> call is available, the minimum is: <code>sysconf(_SC_THREAD_STACK_MIN) / 1024</code> else the minimum is 0.</td>
</tr>
<tr>
<td>Windows ODBC Data Source Administrator</td>
<td><code>Server Stack Size</code> field</td>
<td>Valid values depend on the platform. The default is 768 KB.</td>
</tr>
</tbody>
</table>
Built-In Procedures

TimesTen built-in procedures extend standard ODBC and JDBC functionality. You can invoke these procedures using the ODBC or JDBC procedure call interface. The procedure takes the position of the SQL statement, as illustrated in the following examples.

The following ODBC SQLExecDirect call invokes the `ttOpsSetFlag` built-in procedure to tell the optimizer that it should not generate temporary hash indexes when preparing commands:

```sql
SQLExecDirect (hstmt, (SQLCHAR*)
   "(CALL ttOptSetFlag ('TmpHash', 0))", SQL_NTS);
```

This is the equivalent JDBC call:

```java
CallableStatement cstmt = con.prepareCall
               {"CALL ttOptSetFlag ('TmpHash', 0)"};
            cstmt.execute();
```

TimesTen built-in procedures can also be called from PL/SQL using the `EXECUTE IMMEDIATE` statement with `CALL`, as illustrated in the following example. See "Dynamic SQL in PL/SQL (EXECUTE IMMEDIATE statement)" in the Oracle TimesTen In-Memory Database PL/SQL Developer’s Guide for more details on this statement.

TimesTen built-in procedures that return result sets are not supported directly through OCI. You can use PL/SQL for this purpose. For an example, see "Use of PL/SQL in OCI to call a TimesTen built-in procedure" in the Oracle TimesTen In-Memory Database C Developer’s Guide.

For example, to call the built-in procedure `ttConfiguration`, create a PL/SQL record type and then `SELECT INTO` that record type. Because `ttConfiguration` returns multiple rows, use `BULK COLLECT`.

```plsql
DECLARE
   TYPE ttConfig_record IS RECORD
      (name varchar2(255), value varchar2 (255));
   TYPE ttConfig_table IS TABLE OF ttConfig_record;
   v_ttConfigs ttConfig_table;
BEGIN
   EXECUTE IMMEDIATE  'CALL ttConfiguration'
      BULK COLLECT into v_ttConfigs;
   DBMS_OUTPUT.PUT_LINE ('Name: ' || v_ttConfigs(1).name
                        || ' Value: ' || v_ttConfigs(1).value);
END;
/
```

PL/SQL procedure successfully completed.
You can also call built-in procedures from the `ttIsq1` command line:

```
Command> call ttDBCompactConfig(2000,5,2000);
< 2000, 5, 2000 >
1 row found.
```

**Note:** String parameter values for built-in procedures must be single-quoted as indicated in these examples, unless the value is `NULL`.

---

### List of built-in procedures

This section lists the built-in procedures available in TimesTen.

<table>
<thead>
<tr>
<th>Built-in Name</th>
<th>Description</th>
<th>Classic Support</th>
<th>Grid Support</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ttAgingLRUConfig</code></td>
<td>Sets the Least Recently Used (LRU) aging attributes on all regular tables that have been defined with an LRU aging policy.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><code>ttAgingScheduleNow</code></td>
<td>Starts the aging process</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><code>ttApplicationContext</code></td>
<td>Sets application-defined context for the next update record to pass application specific data to XLA readers.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><code>ttBackupStatus</code></td>
<td>Returns information about the current or last backup of the database.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><code>ttBlockInfo</code></td>
<td>Provides information about perm blocks and the amount of block-level fragmentation in a database.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><code>ttBookmark</code></td>
<td>Returns information about the TimesTen transaction log.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><code>ttCacheADGStandbyStateGet</code></td>
<td>Returns the state for the standby Oracle database in an Active Data Guard configuration.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><code>ttCacheADGStandbyStateSet</code></td>
<td>Sets the state of the standby Oracle database in an Active Data Guard configuration.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><code>ttCacheADGStandbyTimeoutGet</code></td>
<td>Retrieves the timeout value of the Oracle database in an Active Data Guard configuration.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><code>ttCacheADGStandbyTimeoutSet</code></td>
<td>Sets the timeout value of the standby Oracle database in an Active Data Guard configuration.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><code>ttCacheAllowFlushAwtSet</code></td>
<td>Enables you to execute a <code>FLUSH CACHE GROUP</code> statement on an AWT cache group.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><code>ttCacheAutorefIntervalStatsGet</code></td>
<td>Returns statistical information about the last 10 autorefresh cycles for a specified autorefresh interval.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Built-in Name</td>
<td>Description</td>
<td>Classic Support</td>
<td>Grid Support</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>ttCacheAutorefresh</td>
<td>Starts an immediate autorefresh on a set of cache groups.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttCacheAutorefreshLogDefrag</td>
<td>Compacts the trigger log space for a cache autorefresh table.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttCacheAutorefreshStatsGet</td>
<td>Returns information about the last 10 autorefresh transactions on the specified cache group.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttCacheAutorefreshSelectLimit</td>
<td>Configures the incremental autorefresh on a specific number of rows.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttCacheAutorefreshXactLimit</td>
<td>Starts an immediate autorefresh on single table cache groups within a specified autorefresh interval and commits after the specified number of operations.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttCacheAWTMonitorConfig</td>
<td>Sets AWT cache group monitoring.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttCacheAWTThresholdGet</td>
<td>Returns the current transaction log file threshold for databases that include AWT cache groups.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttCacheAWTThresholdSet</td>
<td>Sets the threshold for the number of transaction log files that can accumulate before AWT is considered terminated or too far behind to catch up.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttCacheCheck</td>
<td>Checks for missing constraints for cached tables on the Oracle database</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttCacheConfig</td>
<td>Configures timeout value and recovery policies for cache groups.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttCacheDbCgStatus</td>
<td>Returns the automatic refresh status of the database and the specified cache group.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttCacheDDLTrackingConfig</td>
<td>Configures tracking of DDL statements issued on cached Oracle database tables.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttCachePolicyGet</td>
<td>Returns the current policy used to determine when the TimesTen cache agent for the connected database should run.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttCachePolicySet</td>
<td>Sets the policy used to determine when the TimesTen cache agent for the connected database should run.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttCachePropagateFlagSet</td>
<td>Configures propagation of committed updates to a cache group within the current transaction to the Oracle database.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttCacheSqlGet</td>
<td>Generates the Oracle SQL statements to install or uninstall Oracle database objects for certain types of cache groups.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttCacheStart</td>
<td>Starts the TimesTen cache agent.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Built-in Name</td>
<td>Description</td>
<td>Classic Support</td>
<td>Grid Support</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------------------------------------------------------------------------------------------------</td>
<td>-----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>ttCacheStop</td>
<td>Stops the TimesTen cache agent.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttCacheUidGet</td>
<td>Returns the cache administration user ID.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttCacheUidPwdSet</td>
<td>Sets the cache administration user ID and password.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttCkpt</td>
<td>Performs a non-blocking checkpoint operation.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttCkptBlocking</td>
<td>Performs a blocking checkpoint operation.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttCkptConfig</td>
<td>Reconfigures the background checkpoint manager dynamically or returns the currently active settings of the configuration parameters.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttCkptHistory</td>
<td>Returns information about the last eight checkpoints.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttCommitBufferStats</td>
<td>Returns the number of commit buffer overflows and the high watermark for memory used by transaction reclaim records during transaction commit process.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttCommitBufferStatsReset</td>
<td>Resets transaction commit buffer statistics to 0.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttCompact</td>
<td>Compacts both the permanent and temporary data partitions of the database.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttComputeTabSizes</td>
<td>Refreshes table size statistics stored in TimesTen system tables.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttConfiguration</td>
<td>Returns the values for most, but not all, connection attributes for the current database connection.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttContext</td>
<td>Returns the context value of the current connection.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttDataStoreStatus</td>
<td>Returns the list of processes connected to a database.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttDBCompactConfig</td>
<td>Sets or returns the value of a TimesTen database system parameter.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttDBConfig</td>
<td>Sets or returns the value of a TimesTen database system parameter.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttDBWriteConcurrencyModeGet</td>
<td>Returns information about the write concurrency mode of the database and the status of write concurrency mode operations and transitions.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttDBWriteConcurrencyModeSet</td>
<td>Controls read optimization during periods of concurrent write operations.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttDistributeProgress</td>
<td>Returns a progress report of an ongoing redistribution process.</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
### Table 3-1 (Cont.) TimesTen built-in procedures

<table>
<thead>
<tr>
<th>Built-in Name</th>
<th>Description</th>
<th>Classic Support</th>
<th>Grid Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>ttDurableCommit</td>
<td>Sets transaction durability.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttEpochCreate</td>
<td>Causes the next committed transaction in a grid to commit as an epoch transaction.</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>ttEpochSessionGet</td>
<td>Returns the epoch identifier of the last epoch created by the current connection.</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>ttHeapInfo</td>
<td>Returns information about the size and usage of heap memory.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttHostNameGet</td>
<td>Returns the name of the current local host.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttHostNameSet</td>
<td>Specifies the name of the default local host</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttIndexAdviceCaptureDrop</td>
<td>Drops existing capture data for either the current connection</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttIndexAdviceCaptureEnd</td>
<td>Ends either an active connection level capture from the current connection or an active database level capture</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttIndexAdviceCaptureInfoGet</td>
<td>Returns information for each active capture.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttIndexAdviceCaptureOutput</td>
<td>Returns index recommendations from the last recorded capture at the specified level.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttIndexAdviceCaptureStart</td>
<td>Enables index advice capture.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>tlLatchStatsGet</td>
<td>Displays latch statistics.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>tlLoadFromOracle</td>
<td>Executes a query on the Oracle database and loads the result into a TimesTen table.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>tlLockLevel</td>
<td>Changes the lock level between row-level and database-level locking on the next transaction and for all subsequent transactions for the connection.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>tlLockWait</td>
<td>Changes the lock timeout interval of the current connection.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>tlLogHolds</td>
<td>Returns information about transaction log holds</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>tlMonitorHighWaterReset</td>
<td>Changes the value of the PERM_IN_USE_HIGH_WATER column in the MONITOR system table to the value of the PERM_IN_USE_SIZE and sets the value of the TEMP_IN_USE_HIGH_WATER column to the current value of TEMP_IN_USE_SIZE column.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttOptClearStats</td>
<td>Clears the statistics for the specified table.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Built-in Name</td>
<td>Description</td>
<td>Classic Support</td>
<td>Grid Support</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>ttOptCmdCacheInvalidate</td>
<td>Forces a recompilation should a dependent command be invoked again, or removes the command from the cache. It must be re-prepared by the user.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttOptEstimateStats</td>
<td>Updates the statistics for the specified table.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttOptGetColStats</td>
<td>Returns statistics information in text format.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttOptGetFlag</td>
<td>Returns the optimizer flag settings for the current transaction.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttOptGetMaxCmdFreeListCnt</td>
<td>Returns the size of the free list of SQL compiled command cache.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttOptGetOrder</td>
<td>Returns a single-row result set containing the join order for the current transaction.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttOptSetColIntvlStats</td>
<td>Modifies the statistics for the specified columns with interval information.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttOptSetColStats</td>
<td>Modifies the statistics for the specified columns.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttOptSetFlag</td>
<td>Sets flags to alter the generation of execution plans by the TimesTen query optimizer.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttOptSetMaxCmdFreeListCnt</td>
<td>Sets the maximum count of the free list of SQL compiled commands for regular tables.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttOptSetMaxPriCmdFreeListCnt</td>
<td>Sets the maximum count of the free list of SQL compiled commands that perform materialized view maintenance.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttOptSetOrder</td>
<td>Specifies the order in which tables should be joined by the optimizer.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttOptSetTblStats</td>
<td>Modifies the statistics for the specified table.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttOptShowJoinOrder</td>
<td>Returns the join order of the last prepared or executed SQL statement in the current transaction.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttOptStatsExport</td>
<td>Returns the set of statements required to restore the table statistics to the current state.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttOptUpdateStats</td>
<td>Updates the statistics for the specified table.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttOptUseIndex</td>
<td>Alters the generation of execution plans by the TimesTen query optimizer.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttPLSQLMemoryStats</td>
<td>Returns result statistics about PL/SQL library cache performance and activity.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
### Table 3–1 (Cont.) TimesTen built-in procedures

<table>
<thead>
<tr>
<th>Built-in Name</th>
<th>Description</th>
<th>Classic Support</th>
<th>Grid Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>ttRamPolicyAutoReloadGet</td>
<td>Returns the RAM autoreload policy used to determine if a database is reloaded into RAM after an invalidation.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttRamPolicyAutoReloadSet</td>
<td>Determines the RAM autoreload policy if a database is invalidated.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttRamPolicyGet</td>
<td>Returns the RAM policy used to determine when a database is loaded into memory.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttRamPolicySet</td>
<td>Defines the policy used to determine when a database is loaded into memory.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttRedundantIndexCheck</td>
<td>Scans tables to find redundant indexes.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttRepDeactivate</td>
<td>Changes the state of the active database in an active standby pair from <code>ACTIVE</code> to <code>IDLE</code>.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttReplicationStatus</td>
<td>Returns the status of one or more replication peer databases.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttRepPolicyGet</td>
<td>Returns the replication restart policy</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttRepPolicySet</td>
<td>Specifies the replication restart policy</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttRepQueryThresholdGet</td>
<td>Returns the number of seconds that was most recently specified as the query threshold for the replication agent.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttRepQueryThresholdSet</td>
<td>Specifies the number of seconds that a query can be executed by the replication agent before TimesTen writes a warning to the support log.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttRepStart</td>
<td>Starts the TimesTen replication agent for the connected database.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttRepStateGet</td>
<td>Returns the current replication state of a database in an active standby pair.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttRepStateSave</td>
<td>Saves the state of a remote peer database in an active standby pair to the currently connected database.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttRepStateSet</td>
<td>Sets the replication state of a database in an active standby pair replication scheme.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttRepStop</td>
<td>Stops the TimesTen replication agent for the connected database.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttRepSubscriberStateSet</td>
<td>Changes a replicating subscriber’s state with respect to the executing master store.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttRepSubscriberWait</td>
<td>Causes the caller to wait until all transactions that committed before the call have been transmitted to the subscriber.</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>
### Table 3–1 (Cont.) TimesTen built-in procedures

<table>
<thead>
<tr>
<th>Built-in Name</th>
<th>Description</th>
<th>Classic Support</th>
<th>Grid Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>ttRepSyncGet</td>
<td>Returns static attributes associated with the caller's use of the replication-based return service.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttRepSyncSet</td>
<td>Sets static attributes associated with the caller's use of the replication-based return service.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttRepSyncSubscriberStatus</td>
<td>Queries a subscriber database in a replication scheme configured with a return service and a RETURN DISABLE failure policy to determine whether return service blocking for the subscriber has been disabled by the failure policy.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttRepTransmitGet</td>
<td>Returns the status of transmission of updates to subscribers for the current transaction.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttRepTransmitSet</td>
<td>Updates on the connection it is executed in from being replicated to any subscriber.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttRepXactStatus</td>
<td>Checks the status of a RETURN RECEIPT or RETURN TWOSAFE replication transaction.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttRepXactTokenGet</td>
<td>Returns a token for RETURN RECEIPT or RETURN TWOSAFE replication transactions.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttSetUserColumnID</td>
<td>Sets the value for the user-specified column ID.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttSetUserTableID</td>
<td>Sets the value of the user table ID.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ttSize</td>
<td>Estimates the size of a table or view and the size of indexes.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttSQLCmdCacheInfo</td>
<td>Returns information about all prepared SQL statements in the TimesTen SQL command cache.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttSQLCmdCacheInfoGet</td>
<td>Returns information about the commands in the TimesTen SQL command cache.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttSQLCmdQueryPlan</td>
<td>Returns all detailed runtime query plans for SQL statements in the TimesTen SQL command cache.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttSQLExecutionTimeHistogram</td>
<td>Returns a histogram of SQL execution times.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttStatsConfig</td>
<td>Controls statistics collection and parameters for the ttStats utility.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttStatsConfigGet</td>
<td>Returns parameters of the ttStats utility that you can set with the ttStatsConfig built-in procedure.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ttTableSchemaFromOraQueryGet</td>
<td>Evaluates a SELECT query on a table in an Oracle database and generates a CREATE TABLE SQL statement that you can choose to execute.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
### Table 3–1 (Cont.) TimesTen built-in procedures

<table>
<thead>
<tr>
<th>Built-in Name</th>
<th>Description</th>
<th>Classic Support</th>
<th>Grid Support</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ttVersion</code></td>
<td>Returns TimesTen release information.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><code>ttWarnOnLowMemory</code></td>
<td>Specifies that operations executed on the current connection should return a warning if they allocate memory and find that memory is low.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><code>ttXactIdGet</code></td>
<td>Returns transaction ID information for interpreting lock messages.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><code>ttXlaBookmarkCreate</code></td>
<td>Creates the specified bookmark.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><code>ttXlaBookmarkDelete</code></td>
<td>Deletes the specified bookmark.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><code>ttXlaSubscribe</code></td>
<td>Configures persistent XLA tracking of a table.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><code>ttXlaUnsubscribe</code></td>
<td>Stops persistent XLA tracking of a table.</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>
ttAgingLRUConfig

Description
This procedure sets the Least Recently Used (LRU) aging attributes on all regular tables that have been defined with an LRU aging policy. LRU aging enables you to maintain the amount of memory used in a TimesTen database within a specified threshold by deleting the least recently used data. Data is removed if the database space in-use exceeds the specified threshold values.

For cache groups, LRU aging is defined at the root table for the entire cache instance. LRU aging can be defined for all cache group types except for explicitly loaded autorefresh cache groups. LRU aging is defined by default on dynamic cache groups. For explicitly loaded cache groups, use time-based aging.

For cache tables, the aging policy is defined on the root table but applies to all tables in the cache group. The aging policy is defined on tables when they are created or altered, using the CREATE TABLE or ALTER TABLE SQL statements.

Required privilege
This procedure requires no privilege to query the current values. It requires the ADMIN privilege to change the current values.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax
```
ttAgingLRUConfig([LowUsageThreshold], [HighUsageThreshold], [AgingCycle])
```

Parameters
ttAgingLRUConfig has these optional parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lowUsageThreshold</td>
<td>BINARY_FLOAT</td>
<td>Sets, displays or resets the low end of percentage of database PermSize, specified in decimals. The bottom of the threshold range in which LRU aging should be deactivated. Default is 80 percent.</td>
</tr>
<tr>
<td>highUsageThreshold</td>
<td>BINARY_FLOAT</td>
<td>Sets, displays or resets the high end of percentage of database PermSize, specified in decimals. The top of the threshold range in which LRU aging should be activated. Default is 90 percent.</td>
</tr>
</tbody>
</table>
Result set

`ttAgingLRUConfig` returns these results:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lowUsageThreshold</td>
<td>BINARY_FLOAT NOT NULL</td>
<td>The current setting for the low end of percentage of database <code>PermSize</code>, specified in decimals.</td>
</tr>
<tr>
<td>highUsageThreshold</td>
<td>BINARY_FLOAT NOT NULL</td>
<td>The current setting for the high end of percentage of database <code>PermSize</code>, specified in decimals.</td>
</tr>
<tr>
<td>agingCycle</td>
<td>TT_INTEGER NOT NULL</td>
<td>The current setting for the number of minutes between aging cycles, specified in minutes.</td>
</tr>
</tbody>
</table>

Examples

To set the aging threshold to a low of 75 percent and a high of 95 percent and the aging cycle to 5 minutes, use:

```sql
CALL ttAgingLRUConfig (.75, .90, 5);
<.7500000, .9000000, 5>
```

To display the current LRU aging policy for all tables that defined with an LRU aging policy, call `ttAgingLRUConfig` without any parameters:

```sql
Call ttAgingLRUConfig();
```

If the tables are defined with the default thresholds and aging cycle, the procedure returns:

```sql
<.8000000, .9000000, 1>
1 row found.
```

To change the low usage threshold to 60 percent, the aging cycle to 5 minutes and to retain the previous high usage threshold, use:

```sql
Call ttAgingLRUConfig (60, ,5);
< .6000000, .9000000, 5 >
1 row found.
```

Notes

The values of this procedure are persistent, even across system failures.
If no parameters are supplied, this procedure only returns the current LRU aging attribute settings.

See also

ttAgingScheduleNow

*Oracle TimesTen Application-Tier Database Cache User’s Guide*
**ttAgingScheduleNow**

**Description**

This procedure starts the aging process, regardless of the value of the aging cycle. The aging process begins right after the procedure is called unless there is an aging process in progress. In that case, the new aging process begins when the aging process that was in process at the time the built-in was called has completed.

Aging occurs only once when you call this procedure. This procedure does not change any aging attributes. The previous aging state is unchanged. For example, if aging state is OFF when you call `ttAgingScheduleNow`, the aging process starts. When aging is complete, if your aging state is OFF, aging does not continue. To continue aging, you must call `ttAgingScheduleNow` again or change the aging state to ON, in which case aging occurs next based on the value of the aging cycle.

For tables with aging ON, the aging cycle is reset to the time when `ttAgingScheduleNow` was called. For example, if you call this procedure at 12:00 p.m. and the aging cycle is 15 minutes, aging occurs immediately and again at 12:15, 12:30, 12:45, and so on.

If used in an external scheduler, such as a cron job, or executed manually, this procedure starts the aging process at the time the procedure is executed, if there is no aging process in progress, or as soon as the current aging process has completed. In the case that you want aging to occur only when the external scheduler executes the `ttAgingScheduleNow` procedure or you call it manually, set the aging state to OFF.

Aging is performed by a background thread that wakes up every second to check if any work must be done. Calling `ttAgingScheduleNow` only guarantees that the aging thread works on the specified tables within the next second, at best. If the aging thread is working on a different table at the time the built-in procedure is called, it may take some time to reach the specified table. The rows are visible until the aging thread commits the delete.

**Required privilege**

This procedure requires the DELETE privilege on the table being aged, or the DELETE ANY TABLE privilege when you do not specify a table.

**Usage in TimesTen Scaleout**

TimesTen Scaleout applications can call this built-in procedure.

This procedure executes locally on the element from which it is called.

**Related views**

This procedure has no related views.

**Syntax**

`ttAgingScheduleNow ('tblname')`

**Parameters**

`ttAgingScheduleNow` has the parameter:
ttAgingScheduleNow

Result set

ttAgingScheduleNow returns no results.

Examples

To schedule aging on all tables, including tables defined with both LRU aging and time-based aging, call `ttAgingScheduleNow` without any parameter values:

```
CALL ttAgingScheduleNow ();
```

This example creates the table `agingex` with time-based aging policy and the aging state set to OFF. `ttAgingScheduleNow` is called, using the `ttIsql` utility, to start the aging process once. Rows are deleted from the table. After `ttAgingScheduleNow` is called, the aging state remains OFF. To continue aging, alter the table and set the aging state to OFF.

```
Command> CREATE TABLE agingex (col1 TT_INTEGER PRIMARY KEY NOT NULL,
   ts TIMESTAMP NOT NULL)
   AGING USE ts LIFETIME 1 MINUTES CYCLE 30 MINUTES OFF;
```

```
Command> DESCRIBE agingex;
Table TTUSER.AGINGEX:
Columns:
*COL1             TT_INTEGER NOT NULL
TS               TIMESTAMP (6) NOT NULL
Aging use TS lifetime 1 minute cycle 30 minutes off
1 table found.
(primary key columns are indicated with *)
```

```
Command> INSERT INTO agingex VALUES (1, SYSDATE);
1 row inserted.
```

```
Command> INSERT INTO agingex VALUES (2, SYSDATE);
1 row inserted.
```

```
Command> SELECT * FROM agingex;
< 1, 2011-03-25 13:06:29.000000 >
< 2, 2011-03-25 13:06:42.000000 >
2 rows found.
```

```
Command> CALL ttAgingScheduleNow ('agingex');
```

```
Command> SELECT * FROM agingex;
0 rows found.
```

See also

`ttAgingLRUCfg`

Oracle TimesTen Application-Tier Database Cache User’s Guide
ttApplicationContext

Description
This procedure sets application-defined context for the next update record (either an UPDATE or commit) to pass application specific data to XLA readers.

Required privilege
This procedure requires no privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax

```sql
ttApplicationContext (cmd)
```

Parameters

ttApplicationContext has the parameter:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cmd</td>
<td>VARBINARY(16384) NOT NULL</td>
<td>Context information to be passed to the XLA readers.</td>
</tr>
</tbody>
</table>

Result set

ttApplicationContext returns no results.

Examples

```sql
CALL ttApplicationContext (0x123);
```

See also

"XLA Reference" in Oracle TimesTen In-Memory Database C Developer’s Guide
ttBackupStatus

Description
This procedure returns a single row with information about the current or last backup of the database. If a backup is in progress, this information represents the current backup. If no backup is in progress, this information represents the last backup taken.

If no backup has been taken on the database since the last first-connect, the status field is 0 and the rest of the columns are NULL.

Required privilege
This procedure requires the ADMIN privilege.

Usage in TimesTen Scaleout
TimesTen Scaleout applications can call this built-in procedure.
This procedure executes locally on the element from which it is called.

Related views
This procedure has these related views:
- SYS.GV$BACKUP_STATUS
- SYS.V$BACKUP_STATUS

Syntax

```
ttBackupStatus ()
```

Parameters

ttBackupStatus has no parameters.

Result set

`ttBackupStatus` returns the results:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>status</td>
<td>TT_INTEGER NOT NULL</td>
<td>An INTEGER code representing the current progress of a backup or the completion status of the last backup. Values are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 - No backup has been taken on the database since the last first-connect.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - A backup is currently in progress.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 - The last backup completed successfully.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 - The last backup failed. In this case the error column contains the error code for the failure.</td>
</tr>
</tbody>
</table>
### Built-In Procedures

#### ttBackupStatus

**3 Examples**

```sql
CALL ttBackupStatus();
< 2, 2, 1, 2018-01-12 13:10:32.587557, 2005-08-12 13:10:33.193269, 1, 1531840, 0, 6968 >
1 row found.
```

**Notes**

Does not return information about previous backups, other than the current or last one.

Information returned is not persistent across database startup or shutdown.
ttBlockInfo

Description

This procedure provides information about perm blocks and the amount of block-level fragmentation in a database.

Required privilege

This procedure requires no privilege.

Usage in TimesTen Scaleout

TimesTen Scaleout applications can call this built-in procedure.

This procedure returns a row for the element from which it was called. To see information about other elements, query the SYS.GV$BLOCK_INFO system table.

Related views

This procedure has these related views.

SYS.GV$BLOCK_INFO
SYS.V$BLOCK_INFO

Syntax

ttBlockInfo()

Parameters

ttBlockInfo has no parameters.

Result set

ttBlockInfo returns the result set:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TotalBlocks</td>
<td>TT_BIGINT NOT NULL</td>
<td>Total number of blocks in the database.</td>
</tr>
<tr>
<td>FreeBlocks</td>
<td>TT_BIGINT NOT NULL</td>
<td>Total number of free blocks in the database.</td>
</tr>
<tr>
<td>FreeBytes</td>
<td>TT_BIGINT NOT NULL</td>
<td>Total size of the free blocks.</td>
</tr>
<tr>
<td>LargestFree</td>
<td>TT_BIGINT NOT NULL</td>
<td>Size of the largest free block.</td>
</tr>
</tbody>
</table>

Examples

CALL ttBlockInfo();
< 1537, 16, 236036720, 235991352 >
1 row found.
**ttBookmark**

**Description**

This procedure returns information about the TimesTen transaction log. Records in the transaction log are identified by pairs of integers:

- A transaction log file number.
- An offset in that transaction log file.

Transaction log file numbers correspond to the file system names given to transaction log files. For example, the transaction log file `SalesData.log29` has the transaction log file number 29.

Three log records are identified in the result row of `ttBookmark`:

- The identity of the most recently written log record.
- The identity of the log record most recently forced to the disk.
- The replication bookmark. The replication bookmark is the oldest log record that represents an update not yet replicated to another system.

**Required privilege**

This procedure requires no privilege.

**Usage in TimesTen Scaleout**

TimesTen Scaleout applications can call this built-in procedure.

This procedure executes locally on the element from which it is called.

**Related views**

This procedure has these related views.

`SYS.GV$BOOKMARK`

`SYS.V$BOOKMARK`

**Syntax**

`ttBookmark()`

**Parameters**

`ttBookmark` has no parameters.

**Result set**

`ttBookmark` returns the result set:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>writeLFN</td>
<td>TT_INTEGER</td>
<td>Last written transaction log file.</td>
</tr>
<tr>
<td>writeLFO</td>
<td>TT_BIGINT</td>
<td>Last written offset in transaction log file.</td>
</tr>
<tr>
<td>forceLFN</td>
<td>TT_INTEGER</td>
<td>Last transaction log file forced to disk.</td>
</tr>
<tr>
<td>forceLFO</td>
<td>TT_BIGINT</td>
<td>Offset of last transaction log file forced to disk.</td>
</tr>
</tbody>
</table>
### Examples

CALL ttBookmark();

\[<379, 60193048, 379, 60192768, -1, -1>\]

1 row found.

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>holdLFN</td>
<td>TT_INTEGER</td>
<td>Replication bookmark transaction log file.</td>
</tr>
<tr>
<td>holdLFO</td>
<td>TT_BIGINT</td>
<td>Replication bookmark log offset.</td>
</tr>
</tbody>
</table>
**ttCacheADGStandbyStateGet**

Returns the state for the standby Oracle database that was specified with the ttCacheADGStandbyStateSet built-in procedure.

**Required privilege**

This procedure requires no privileges.

**Usage in TimesTen Scaleout**

This procedure is not supported in TimesTen Scaleout.

**Related views**

This procedure has no related views.

**Syntax**

```sql
ttCacheADGStandbyStateGet()
```

**Parameters**

ttCacheADGStandbyStateGet has no parameters.

**Result set**

ttCacheADGStandbyStateGet returns the state of the standby Oracle database.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| ADGStandbyState | TT_VARCHAR(20) | OK: The standby Oracle database in an Active Data Guard configuration is considered to be up.  
|                  |             | FAILED: The standby Oracle database in the Active Data Guard configuration has failed. The cache agent does not try to contact the standby Oracle database and continues the autorefresh with only the primary Oracle database. |

**Examples**

The following example shows how to call the ttCacheADGStandbyStateSet built-in procedure to set the state of the standby Oracle database in an Active Data Guard environment first to OK and then to FAILED. The ttCacheADGStandbyStateGet built-in procedure retrieves the value of the current state of the standby Oracle database.

```
Command> call ttCacheADGStandbyStateSet('OK');
Command> call ttCacheADGStandbyStateGet();
< OK >
1 row found.

Command> call ttCacheADGStandbyStateSet('FAILED');
Command> call ttCacheADGStandbyStateGet();
< FAILED >
1 row found.
```
See also

- `ttCacheADGStandbyStateGet`
- `ttCacheADGStandbyTimeoutGet`
- `ttCacheADGStandbyTimeoutSet`
For an Active Data Guard environment, the user can call the `ttCacheADGStandbyStateSet` built-in procedure to inform the cache agent of the state of the standby Oracle database.

- Set the state of the standby Oracle database to **OK** and the cache agent autorefreshes only those transactions that have been replicated from the primary Oracle database to the standby Oracle database.
  - If the standby Oracle database fails and if you have set a timeout with the `ttCacheADGStandbyTimeoutSet` built-in procedure, then the state changes to **FAILED** if the standby Oracle database does not respond after the timeout is reached.
  - If the standby Oracle database fails and you did not set the timeout, then autorefresh stalls until the standby Oracle database recovers (unless you set the state of the standby Oracle database to **FAILED**).
- Set the state of the standby Oracle database to **FAILED** if you know the standby Oracle database has failed and it should not be used as part of the autorefresh. When you set the state to **FAILED**, the cache agent does not wait for transactions to be replicated to the standby Oracle database and continues the autorefresh with only the primary Oracle database. The cache agent does not include the standby Oracle database in the autorefresh, even if it has recovered and is currently active, until you change the state to **OK**.

**Required privilege**

This procedure requires no privileges.

**Usage in TimesTen Scaleout**

This procedure is not supported in TimesTen Scaleout.

**Related views**

This procedure has no related views.

**Syntax**

```
(ttCacheADGStandbyStateSet('OK | FAILED'))
```

**Parameters**

`ttCacheADGStandbyStateSet` has the parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADGStandbyState</td>
<td>TT_VARCHAR(20)</td>
<td><strong>OK</strong>: Tells the cache agent that the standby Oracle database in an Active Data Guard configuration is active. This is the default. <strong>FAILED</strong>: Tells the cache agent that the standby Oracle database in the Active Data Guard configuration has failed. The cache agent does not try to contact the standby Oracle database and continues the autorefresh with only the primary Oracle database.</td>
</tr>
</tbody>
</table>
Result set

ttCacheADGStandbyStateSet returns no results.

Examples

The following example shows how to call the ttCacheADGStandbyStateSet built-in procedure to set the state of the standby Oracle database in an Active Data Guard environment first to OK and then to FAILED. The ttCacheADGStandbyStateGet built-in procedure retrieves the value of the current state of the standby Oracle database.

Command> call ttCacheADGStandbyStateSet('OK');
Command> call ttCacheADGStandbyStateGet();
< OK >
1 row found.

Command> call ttCacheADGStandbyStateSet('FAILED');
Command> call ttCacheADGStandbyStateGet();
< FAILED >
1 row found.

See also

ttCacheADGStandbyStateGet
ttCacheADGStandbyTimeoutGet
ttCacheADGStandbyTimeoutSet
ttCacheADGStandbyTimeoutGet

Retrieve the timeout specified with the ttCacheADGStandbyTimeoutSet built-in procedure.

Required privilege
This procedure requires no privileges.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax

```
ttCacheADGStandbyTimeoutGet()
```

Parameters
ttCacheADGStandbyTimeoutGet has no parameters.

Result Set
ttCacheADGStandbyTimeoutGet returns the timeout:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADGStandbyTimeout</td>
<td>TT_INTEGER</td>
<td>The timeout in seconds on how long to wait for a response from the standby Oracle database in an Active Data Guard configuration before using only the primary Oracle database to autorefresh the cache groups.</td>
</tr>
</tbody>
</table>

Examples

The following example shows how to use the ttCacheADGStandbyTimeoutSet built-in procedure to set the timeout that indicates the time to wait for a response from the standby Oracle database. And then, it shows how to call the ttCacheADGStandbyTimeoutGet built-in procedure to retrieve the value of the timeout.

Command> call ttCacheADGStandbyTimeoutSet('60');
Command> call ttCacheADGStandbyTimeoutGet();
< 60 >
1 row found.

See also

- ttCacheADGStandbyStateSet
- ttCacheADGStandbyStateGet
- ttCacheADGStandbyTimeoutSet
ttCacheADGStandbyTimeoutSet

You can set a timeout with the `ttCacheADGStandbyTimeoutSet` built-in procedure to designate how long to wait for a response from the standby Oracle database in an Active Data Guard configuration. If the standby Oracle database does not respond after this period, then the state of the standby Oracle database is automatically changed to FAILED and the cache agent facilitates autorefresh using only the primary Oracle database.

**Note:** At any time, the user can restore the standby Active Data Guard state by executing the `ttCacheADGStandbyStateSet` built-in procedure and set the state to OK.

### Required privilege

This procedure requires no privileges.

### Usage in TimesTen Scaleout

This procedure is not supported in TimesTen Scaleout.

### Related views

This procedure has no related views.

### Syntax

```
    ttCacheADGStandbyTimeoutSet(timeout)
```

### Parameters

`ttCacheADGStandbyTimeoutSet` has the parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADGStandbyTimeout</td>
<td>TT_INTEGER</td>
<td>A timeout specified in seconds on how long to wait for the standby Oracle</td>
</tr>
<tr>
<td></td>
<td>NOT NULL</td>
<td>database to respond before using only the primary Oracle database to</td>
</tr>
<tr>
<td></td>
<td></td>
<td>autorefresh the cache groups.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default is 0, which indicates that no timeout is used and the state of the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>standby Oracle database does not change from OK to FAILED as a result of</td>
</tr>
<tr>
<td></td>
<td></td>
<td>this timeout.</td>
</tr>
</tbody>
</table>

### Result set

`ttCacheADGStandbyTimeoutSet` returns no results.

### Examples

The following example shows how to use the `ttCacheADGStandbyTimeoutSet` built-in procedure to set the timeout that indicates the time to wait for a response from the standby Oracle database. And then, it shows how to call the `ttCacheADGStandbyTimeoutGet` built-in procedure to retrieve the value of the timeout.

```
    Command> call ttCacheADGStandbyTimeoutSet('60');
    Command> call ttCacheADGStandbyTimeoutGet();
```
< 60 >
1 row found.

See also

ttCacheADGStandbyStateSet
ttCacheADGStandbyTimeoutGet
ttCacheADGStandbyStateGet
ttCacheAllowFlushAwtSet

Description

The `ttCacheAllowFlushAwtSet` built-in procedure enables you to execute a FLUSH CACHE GROUP statement against an AWT cache group and should only be used in a specific recovery scenario, as described in "When there is unsynchronized data in the cache groups" section in the Oracle TimesTen Application-Tier Database Cache User’s Guide.

Set auto commit to off before executing the `ttCacheAllowFlushAwtSet` built-in procedure when setting the `enableFlush` parameter to 1; otherwise, this parameter automatically resets to 0 directly after executing the built-in procedure. Then, perform a commit after you execute the FLUSH CACHE GROUP statement and execute the `ttCacheAllowFlushAwtSet` built-in procedure to reset the `enableFlush` parameter back to 0.

Required privilege

This procedure requires no privileges.

Usage in TimesTen Scaleout

This procedure is not supported in TimesTen Scaleout.

Related views

This procedure has no related views.

Syntax

`ttCacheAllowFlushAwtSet (enableFlush)`

Parameters

`ttCacheAllowFlushAwtSet` has the parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>allow</td>
<td>TT_INTEGER</td>
<td>0 - The user is prevented from executing a FLUSH CACHE GROUP statement against an AWT cache group, which is the intended restriction.</td>
</tr>
<tr>
<td></td>
<td>NOT NULL</td>
<td>1 - The user is allowed to execute a FLUSH CACHE GROUP statement against an AWT cache group, which should only be done for recovery, as described in &quot;When there is unsynchronized data in the cache groups&quot; section in the Oracle TimesTen Application-Tier Database Cache User’s Guide.</td>
</tr>
</tbody>
</table>

Result set

`ttCacheAllowFlushAwtSet` returns no results.

Examples

The following example shows how to execute the `ttCacheAllowFlushAwtSet` built-in procedure to first allow and then disallow a FLUSH CACHE GROUP statement to be executed against the marketbasket AWT cache group.

---
Command> set autocommit off;
   CALL ttCacheAllowFlushAwtSet(1);
   FLUSH CACHE GROUP marketbasket;
   CALL ttCacheAllowFlushAwtSet(0);
   COMMIT;

See also

"When there is unsynchronized data in the cache groups" section in the Oracle TimesTen Application-Tier Database Cache User’s Guide.
ttCacheAutorefIntervalStatsGet

Description

The `ttCacheAutorefIntervalStatsGet` built-in procedures returns statistical information about the last 10 autorefresh cycles for a particular autorefresh interval.

Required privilege

This procedure requires no privileges.

Usage in TimesTen Scaleout

This procedure is not supported in TimesTen Scaleout.

Related views

This procedure has no related views.

Syntax

```

ttCacheAutorefIntervalStatsGet (autoRefInterval, isStatic)
```

Parameters

`ttCacheAutorefIntervalStatsGet` has the parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>autoRefInterval</code></td>
<td>TT_BIGINT NOT NULL</td>
<td>The <code>autorefreshInterval</code> designates the cache group (the one with this autorefresh interval value) on which to gather statistics. The integer value for the autorefresh interval (in milliseconds) is the same value that was originally specified when the autorefresh cache group was created to indicate how often autorefresh is scheduled.</td>
</tr>
<tr>
<td><code>isStatic</code></td>
<td>TT_INTEGER</td>
<td>Indicates if you are to retrieve information on static or dynamic cache groups with the interval value: 0 - dynamic cache groups 1 - static (non-dynamic) cache groups The default is static.</td>
</tr>
</tbody>
</table>

Result set

`ttCacheAutorefIntervalStatsGet` returns statistical information about the last 10 autorefresh cycles for a particular autorefresh interval:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>autorefInterval</code></td>
<td>TT_BIGINT</td>
<td>Autorefresh interval in milliseconds.</td>
</tr>
</tbody>
</table>
ttCacheAutorefIntervalStatsGet

Examples

The following example shows how to execute `ttCacheAutorefIntervalStatsGet` built-in procedure to retrieve statistics for autorefresh cache groups that have been defined as static and have the interval of seven seconds:

Command> call ttCacheAutorefIntervalStatsGet(7000,1);

< 7000, 1, 41, 2013-04-25 15:17:00.000000, 0, 0, 1, 0, 0, <NULL>,
132121, 132121, 13, 21, 0, 0, 0, 0 >
< 7000, 1, 40, 2013-04-25 15:16:53.000000, 0, 0, 1, 0, 0, <NULL>,
132121, 132121, 12, 21, 0, 0, 0, 0 >
< 7000, 1, 39, 2013-04-25 15:16:46.000000, 0, 0, 1, 0, 0, <NULL>,
132121, 132121, 11, 21, 0, 0, 0, 0 >
< 7000, 1, 38, 2013-04-25 15:16:39.000000, 0, 0, 1, 0, 0, <NULL>,
132121, 132121, 10, 21, 0, 0, 0, 0 >
< 7000, 1, 37, 2013-04-25 15:16:32.000000, 0, 0, 1, 0, 0, <NULL>,
132121, 132121, 9, 21, 0, 0, 0, 0 >
< 7000, 1, 36, 2013-04-25 15:16:24.000000, 0, 15616, 15616, 1, 0, 131072,
<NULL>, 132121, 132121, 8, 21, 0, 0, 0, 0 >
< 7000, 1, 35, 2013-04-25 15:16:17.000000, 0, 18176, 18176, 1, 0, 131072,
<NULL>, 132121, 132121, 7, 21, 0, 0, 0, 0 >
< 7000, 1, 34, 2013-04-25 15:16:10.000000, 0, 14336, 14336, 1, 0, 131072,
<NULL>, 92024, 92024, 6, 21, 0, 0, 0, 0 >

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| isStatic       | TT_INTEGER| Indicates that the information is for static or dynamic cache groups with the interval value:
|                |           | 0 - dynamic cache groups
|                |           | 1 - static (non-dynamic) cache groups                                         |
| autorefNumber  | TT_BIGINT | Autorefresh number.                                                         |
| startTimestamp | TT_TIMESTAMP| Autorefresh start time.                                                  |
| selectLimit    | TT_BIGINT | Select row limit set for incremental autorefresh cache group.               |
| numRows        | TT_BIGINT | Number of rows refreshed.                                                   |
| numOps         | TT_BIGINT | Number of SQL operations executed.                                         |
| numCommits     | TT_BIGINT | Number of commits.                                                         |
| commitBufSize  | TT_BIGINT | Maximum commit buffer size in bytes.                                       |
| commitBufMaxReached | TT_BIGINT | Amount of memory used for commit processing in bytes.                      |
| commitBufNumOverflows | TT_BIGINT | Number of times the commit buffer overflowed for each transaction.       |
| totalNumRows   | TT_BIGINT | Number of rows refreshed since the autorefresh thread was started.        |
| totalNumOps    | TT_BIGINT | Number of SQL operations were executed since the autorefresh thread was started. |
| totalNumCommits| TT_BIGINT | Number of commits since the autorefresh thread was started.               |
| totalNumRollbacks | TT_BIGINT | Number of rollbacks since the autorefresh thread started.                 |
| totalNumSnapshotOld | TT_BIGINT | Number of "Snapshot too old" errors received since the autorefresh thread started |
Notes

This procedure is available only for TimesTen Cache.

See also

ttCacheAutorefreshSelectLimit
ttCacheAutorefreshXactLimit

"Executing large transactions with incremental autorefresh read-only cache groups" and "Configuring a select limit when using incremental autorefresh for read-only cache groups" in the Oracle TimesTen Application-Tier Database Cache User’s Guide.
ttCacheAutorefresh

Description
This procedure starts an immediate autorefresh on the set of cache groups that are associated by sharing the same autorefresh interval with the specified cache group. This set of associated cache groups would normally be refreshed together automatically. The effect on the autorefresh process is the same as that of adding a new cache group with the same refresh interval as that of the specified cache group. This procedure is useful if updates have occurred on the Oracle database and you would like to refresh them on the cache group before the next scheduled autorefresh.

If there is an existing transaction with locks on table objects that belong to the set of cache groups to be autorefreshed, this procedure returns an error without taking any action. This procedure establishes a condition that requires that you commit or rollback before you can perform other work in the session.

Required privilege
This procedure requires the CACHE_MANAGER or ADMIN privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax
```
ttCacheAutorefresh ('cgOwner', 'cgName', synchronous)
```

Parameters
ttCacheAutorefresh has the parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cgOwner</td>
<td>VARCHAR2 (30)</td>
<td>Name of the cache group owner.</td>
</tr>
<tr>
<td>cgName</td>
<td>VARCHAR2 (30) NOT NULL</td>
<td>Name of the cache group.</td>
</tr>
<tr>
<td>synchronous</td>
<td>TT_INTEGER</td>
<td>Species whether data is updated on synchronously or asynchronously.</td>
</tr>
</tbody>
</table>

 0 or NULL - Asynchronous mode. The procedure returns immediately.

1 - Synchronous mode. The procedure returns after the refresh operation has completed on all associated cache groups.

Result set

ttCacheAutorefresh returns no results.
Examples

This example autorefreshes the testcache cache group and all cache groups with the same autorefresh interval. The procedure returns synchronously.

Command> call ttcacheautorefresh('user1', 'testcache', 1);

Notes

The specified cache group AUTOREFRESH state must be ON. While, other associated cache groups can be in any state, they are not refreshed if they are not in the autorefresh ON state.

An autorefresh of the specified associated cache groups cannot be in progress.

You cannot call this procedure on the standby node of an active standby pair.

This procedure is available only for TimesTen Cache.
ttCacheAutorefreshLogDefrag

Description
The `ttCacheAutorefreshLogDefrag` built-in procedure compact the trigger log space for a cache autorefresh table.

For usage details, see "Defragmenting change log tables in the tablespace" in the Oracle TimesTen Application-Tier Database Cache User’s Guide.

Required privilege
This procedure requires the `CACHE_MANAGER` privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax
```
ttCacheAutorefreshLogDefrag ('action')
```

Parameters
`ttCacheAutorefreshLogDefrag` has the parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action</td>
<td>VARCHAR (50)</td>
<td>NOT NULL</td>
</tr>
</tbody>
</table>

Acceptable values are:
- `Compact` - Defragments only the trigger log space.
- `CompactAndReclaim` - Defragments the trigger log space and the transaction commit buffer (reclaim space).

NOTE: The reclaim phase takes a lock on the trigger log table for a brief moment. This can suspend the workload from writing into the base table.

Result set
`ttCacheAutorefreshLogDefrag` returns no results.

Examples
In this example, the call compacts or defragments only the trigger log space.
```
Command> call ttCacheAutorefreshLogDefrag('CompactOnly');
```
Notes

This procedure is available only for TimesTen Cache.

See also

ttCacheConfig
ttCacheAutorefreshStatsGet
ttCacheAutorefreshStatsGet

Description
This procedure returns information about the last ten autorefresh transactions on the specified cache group. This information is only available when the AUTOREFRESH state is ON or PAUSED, and the cache agent is running.

The information returned by this built-in procedure is reset whenever:

- The cache agent is restarted
- The state is set to OFF and then back to ON or PAUSED
- The cache group is dropped and recreated

Required privilege
This procedure requires no privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax
```
ttCacheAutorefreshStatsGet ('cgOwner', 'cgname')
```

Parameters
```
Parameter          Type        Description
cgOwner            VARCHAR2 (30) Name of the cache group owner.
cgName             VARCHAR2 (30) NOT NULL Name of the cache group for which autorefresh information should be returned.
```

Result set
The ttCacheAutorefreshStatsGet built-in procedure returns only a subset of column information for a cache group with autorefresh mode FULL. A column value of 0 returns for information that is not available.

```
Column name       Column type     Description                                           Returned for full autorefresh
cgId              TT_BIGINT       The cache group ID.                                         Y
startTimestamp    TT_TIMESTAMP    Timestamp when autorefresh started for this interval. See "Notes" below. Y
```
<table>
<thead>
<tr>
<th>Column name</th>
<th>Column type</th>
<th>Description</th>
<th>Returned for full autorefresh</th>
</tr>
</thead>
<tbody>
<tr>
<td>cacheAgentUpTime</td>
<td>TT_BIGINT</td>
<td>Number of cache agent clock ticks in milliseconds at the time the autorefresh transaction started for this interval. This value is cumulative and is reset when the cache agent process starts. See &quot;Notes&quot; below.</td>
<td>Y</td>
</tr>
<tr>
<td>autorefNumber</td>
<td>TT_BIGINT</td>
<td>Autorefresh number for a cache group indicates the number of times this cache group has been incrementally refreshed since the cache agent started. This number is initialized to 0 when the cache agent is started.</td>
<td>Y</td>
</tr>
<tr>
<td>autorefDuration</td>
<td>TT_BIGINT</td>
<td>The number of milliseconds spent in this autorefresh transaction.</td>
<td>Y</td>
</tr>
<tr>
<td>autorefNumRows</td>
<td>TT_BIGINT</td>
<td>The number of rows autorefreshed in this autorefresh. This includes all rows, including those in the root table and the child tables. If there are cache groups with multiple tables, child table rows get updated multiple times. Therefore, the number of rows autorefreshed may be more than the number of rows updated on the Oracle database.</td>
<td>N</td>
</tr>
<tr>
<td>numOracleBytes</td>
<td>TT_BIGINT</td>
<td>The number of bytes transferred from the Oracle database in this autorefresh transaction.</td>
<td>N</td>
</tr>
<tr>
<td>autorefNumRootTblRows</td>
<td>TT_BIGINT</td>
<td>The number of root table rows autorefreshed in this autorefresh transaction.</td>
<td>Y</td>
</tr>
<tr>
<td>autorefQueryExecDuration</td>
<td>TT_BIGINT</td>
<td>The duration in milliseconds that it takes for the autorefresh query to execute on the Oracle database.</td>
<td>N</td>
</tr>
<tr>
<td>autorefQueryFetchDuration</td>
<td>TT_BIGINT</td>
<td>The duration in milliseconds that it takes for the autorefresh query to fetch rows from the Oracle database.</td>
<td>N</td>
</tr>
<tr>
<td>autorefTtApplyDuration</td>
<td>TT_BIGINT</td>
<td>The duration in milliseconds that it takes for TimesTen to apply the autorefresh.</td>
<td>N</td>
</tr>
<tr>
<td>totalNumRows</td>
<td>TT_BIGINT</td>
<td>The total number of rows autorefreshed since the cache agent started. The total number of rows autorefreshed may not be the same as number of rows updated on the Oracle database. This is because of a delay in marking the log; some updates may get autorefreshed and counted multiple times.</td>
<td>N</td>
</tr>
<tr>
<td>totalNumOracleBytes</td>
<td>TT_BIGINT</td>
<td>The total number of bytes transferred from the Oracle database since the cache agent started.</td>
<td>N</td>
</tr>
<tr>
<td>totalNumRootTblRows</td>
<td>TT_BIGINT</td>
<td>The total number of root table rows autorefreshed since the cache agent started.</td>
<td>Y</td>
</tr>
</tbody>
</table>
Examples

In this example, testcache is a READONLY cache group with one table and an incremental autorefresh interval of 10 seconds.

Command> call ttcacheautorefreshstatsget('user1','testcache');

< 1164260, 2011-07-23 15:43:52.000000, 850280, 44, 0, 75464, 528255, 75464, 310, 110, 6800, 1890912, 12439795, 1890912, 160020, InProgress, 2, 74 >
< 1164260, 2011-07-23 15:43:33.000000, 831700, 43, 13550, 108544, 759808, 108544, 1030, 230, 12290, 1815448, 11911540, 1815448, 160020, Complete, 2, 72 >
< 1164260, 2011-07-23 15:43:12.000000, 810230, 42, 17040, 115712, 809984, 115712, 610, 330, 16090, 1706904, 11151732, 1706904, 146470, Complete, 2, 70 >
< 1164260, 2011-07-23 15:42:52.000000, 790190, 41, 14300, 94208, 699456, 94208, 560, 320, 13410, 1591192, 8986996, 1591192, 129430, Complete, 2, 68 >
< 1164260, 2011-07-23 15:42:12.000000, 750130, 39, 13530, 112640, 700768, 112640, 530, 230, 12780, 1311640, 8388628, 1311640, 92670, Complete, 2, 64 >
< 1164260, 2011-07-23 15:41:52.000000, 730130, 38, 13530, 112640, 700768, 112640, 530, 220, 12780, 1311640, 8388628, 1311640, 92670, Complete, 2, 62 >
< 1164260, 2011-07-23 15:41:32.000000, 710120, 37, 9370, 56320, 326810, 56320, 310, 160, 8900, 1199000, 8388628, 1199000, 114970, Complete, 2, 60 >

<table>
<thead>
<tr>
<th>Column name</th>
<th>Column type</th>
<th>Description</th>
<th>Returned for full autorefresh</th>
</tr>
</thead>
<tbody>
<tr>
<td>totalDuration</td>
<td>TT_BIGINT</td>
<td>The total autorefresh duration in milliseconds since the cache agent started.</td>
<td>Y</td>
</tr>
<tr>
<td>status</td>
<td>VARCHAR2 (128)</td>
<td>A string description of the status of the current autorefresh. See &quot;Notes&quot; below. Supported values for this field are: Complete inProgress Failed</td>
<td>Y</td>
</tr>
<tr>
<td>numlogrows</td>
<td>TT_BIGINT</td>
<td>Number of rows fetched from the Oracle database in this autorefresh.</td>
<td>Y</td>
</tr>
<tr>
<td>totalnumlogrows</td>
<td>TT_BIGINT</td>
<td>The cumulative number of rows fetched from the Oracle database in this autorefresh.</td>
<td>Y</td>
</tr>
<tr>
<td>autorefLogFragmentationPct</td>
<td>TT_BIGINT</td>
<td>A low-water mark for table usage by percentage. If less than the specified percent of the table is used, the table is compacted.</td>
<td>Y</td>
</tr>
<tr>
<td>autorefLogFragmentationTs</td>
<td>TT_TIMESTAMP</td>
<td>The timestamp when the last utilization/fragmentation ratio was calculated</td>
<td>Y</td>
</tr>
<tr>
<td>autorefLogDefragGcnt</td>
<td>TT_BIGINT</td>
<td>The number of times the table has been compacted.</td>
<td>Y</td>
</tr>
</tbody>
</table>

The total autorefresh duration in milliseconds since the cache agent started.

A string description of the status of the current autorefresh. See "Notes" below. Supported values for this field are: Complete inProgress Failed

Number of rows fetched from the Oracle database in this autorefresh.

The cumulative number of rows fetched from the Oracle database in this autorefresh.

A low-water mark for table usage by percentage. If less than the specified percent of the table is used, the table is compacted.

The timestamp when the last utilization/fragmentation ratio was calculated

The number of times the table has been compacted.
Notes

Most of the column values reported above are collected at the cache group level. For example, autorefDuration and autorefNumRows only include information for the specified cache group. Exceptions to this rule are column values cacheAgentUpTime, startTimestamp and autorefreshStatus. These values are reported at the autorefresh interval level.

StartTimestamp is taken at the beginning of the autorefresh for the autorefresh interval. A cache group enters the in progress state as soon as the autorefresh for the interval starts. It is not marked complete until the autorefresh for all cache groups in the interval are complete.

This procedure is available only for TimesTen Cache.
ttCacheAutorefreshSelectLimit

Description
Configuring the incremental autorefresh to join the Oracle database base table with a limited number of rows from the autorefresh change log table is known as configuring a select limit. This is accomplished with the ttCacheAutorefreshSelectLimit built-in procedure.

Required privilege
This procedure requires the ADMIN or CACHE_MANAGER privileges.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax
```
ttCacheAutorefreshSelectLimit ( autorefreshInterval, value )
```

Parameters
`ttCacheAutorefreshSelectLimit` has the parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>param</code></td>
<td>TT_VARCHAR2 (50) NOT NULL</td>
<td>The <code>autorefreshInterval</code> designates the cache group (the one with this autorefresh interval value) on which to apply the <code>value</code>. The integer value for the autorefresh interval (in milliseconds) is the same value that was originally specified when the autorefresh cache group was created to indicate how often autorefresh is scheduled.</td>
</tr>
<tr>
<td><code>value</code></td>
<td>TT_VARCHAR2 (200)</td>
<td>The <code>value</code> denotes a limit of the number of rows to select from the autorefresh change log file to apply to the cached table. These changes are applied incrementally until all the rows in the autorefresh change log table have been applied. If the value changes, it takes effect at the start of the next autorefresh cycle. The <code>value</code> can be one of the following:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 'ON': Select at most 1000 rows at a time from the autorefresh change log table to apply for every autorefresh cycle.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ <code>number</code>: Select at most a user specified number of rows from the autorefresh change log table during the autorefresh cycle. If the user specified a limit size of 2000 rows, then autorefresh selects at most 2000 rows at a time from the autorefresh change log table. If you specify a negative number, an error is returned.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ 'OFF': Disables the select limit. The incremental autorefresh selects all rows from the change log table during the autorefresh cycle.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ <code>NULL</code>: If the <code>value</code> provided is <code>NULL</code> or not specified, the current setting is returned.</td>
</tr>
</tbody>
</table>
ttCacheAutorefreshSelectLimit returns the select limit value that has been set for a particular autorefresh interval:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>param</td>
<td>TT_VARCHAR2(50)</td>
<td>The autorefreshInterval that designates the cache group (the one with this autorefresh interval value).</td>
</tr>
<tr>
<td>value</td>
<td>TT_VARCHAR2(200)</td>
<td>The current value that shows the number of rows that is selected from the autorefresh change log file to apply to the cached table.</td>
</tr>
</tbody>
</table>

Examples

You can show the current setting by either providing a NULL value or no parameter. The following example shows the setting for incremental autorefresh cache groups with an interval value of 7 seconds.

```
Command> call ttCacheAutorefreshSelectLimit('7000', NULL);
< 7000, 2000 >
1 row found.
Command> call ttCacheAutorefreshSelectLimit('7000');
< 7000, 2000 >
1 row found.
```

The following example set a select limit to 2000 rows for incremental autorefresh cache groups with an interval value of 7 seconds.

```
Command> call ttCacheAutorefreshSelectLimit('7000', '2000');
< 7000, 2000 >
1 row found.
```

Notes

- This procedure is available only for TimesTen Cache.
- The ttCacheAutotrefreshSelectLimit built-in procedure can set a select limit only on an interval that is defined for a single cache group that contains one table, where the cache group is defined as a static read-only cache group with incremental autorefresh.
- The setting for ttCacheAutorefreshSelectLimit is not replicated or duplicated. The user must execute the built-in on both the active and standby nodes.
- The settings do not reset if you drop all cache groups for the interval.
- The ttMigrate, ttBackup, and ttRestore built-in procedures do not preserve the setting of ttCacheAutorefreshSelectLimit.
- If you alter the cache group autorefresh interval, it does not modify what was set previously through execution of ttCacheAutorefreshSelectLimit for the cache group. You can only alter the select limit for the cache group with the ttCacheAutorefreshSelectLimit built-in procedure.

See also

- ttCacheAutorefIntervalStatsGet
- "Configuring a select limit when using incremental autorefresh for read-only cache groups" in the Oracle TimesTen Application-Tier Database Cache User’s Guide.
**ttCacheAutorefreshXactLimit**

**Description**
This procedure starts an immediate autorefresh on single table cache groups within a specified autorefresh interval and commits after the specified number of operations. This procedure is useful if updates have occurred on the Oracle database and you want to refresh them on the cache group before the next scheduled autorefresh.

To modify the reclaim buffer size, use the `ttDBConfig` built-in procedure.

**Required privilege**
This procedure requires the `CACHE_MANAGER` or `ADMIN` privilege.

**Usage in TimesTen Scaleout**
This procedure is not supported in TimesTen Scaleout.

**Related views**
This procedure has no related views.

**Syntax**
```
ttCacheAutorefreshXactLimit ('IntervalValue', 'Value')
```

**Parameters**
`ttCacheAutorefreshXactLimit` has the parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>param</td>
<td>VARCHAR2 (50) NOT NULL</td>
<td>Indicates the interval at which the autorefresh cache groups are defined to occur in units of milliseconds. <code>IntervalValue</code> is an integer value in milliseconds that was specified when the autorefresh cache group was created on how often autorefresh is scheduled.</td>
</tr>
<tr>
<td>value</td>
<td>VARCHAR2 (200)</td>
<td>The <code>Value</code> can be one of the following:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 'ON' - Enables autorefresh to commit after every 256 operations.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 'OFF' - Disables the transaction limit for autorefresh cache groups and sets autorefresh back to using a single transaction.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- number - Denotes when to commit after a certain number of operations. For example, if the user specifies 1024, then autorefresh commits after every 1024 operations in the transaction. If you specify a negative number, an error is returned.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- NULL - When the value is NULL, 0 or not specified, the current setting is returned.</td>
</tr>
</tbody>
</table>
Result set

`ttCacheAutorefreshXactLimit` returns the results:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>param</td>
<td>VARCHAR2 (50) NOT NULL</td>
<td>The interval at which the autorefresh cache groups are defined to occur in units of milliseconds.</td>
</tr>
<tr>
<td>value</td>
<td>VARCHAR2 (200)</td>
<td>The value can be one of the following:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 'ON' - Enables autorefresh to commit after every 256 operations.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 'OFF' - Disables the transaction limit for autorefresh cache groups and sets autorefresh back to using a single transaction.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- number - Denotes when to commit after a certain number of operations. For example, if the user specifies 1024, then autorefresh commits after every 1024 operations in the transaction. If you specify a negative number, an error is returned.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- NULL - When the value is NULL or not specified, the current setting is returned.</td>
</tr>
</tbody>
</table>

Examples

The following example sets up the transaction limit to commit after every 256 operations for all incremental autorefresh read-only cache groups that are defined with an interval value of 10 seconds.

```plaintext
call ttCacheAutorefreshXactLimit('10000', 'ON');
```

After the month end process has completed and the incremental autorefresh read-only cache groups are refreshed, disable the transaction limit for incremental autorefresh read-only cache groups that are defined with the interval value of 10 seconds.

```plaintext
call ttCacheAutorefreshXactLimit('10000', 'OFF');
```

To enable the transaction limit for incremental autorefresh read-only cache groups to commit after every 2000 operations, provide 2000 as the value as follows:

```plaintext
call ttCacheAutorefreshXactLimit('10000', '2000');
```

Notes

- This procedure is available only for TimesTen Cache. This built-in procedure only applies for static read-only cache groups with incremental autorefresh.
- While autorefresh is in-progress and is being applied in several small transactions, transactional consistency cannot be maintained. Once the autorefresh cycle has completed, the data is transactional consistent.
- The setting for `ttCacheAutorefreshXactLimit` is not replicated or duplicated. The user must execute the built-in procedure on both the active and standby nodes.
- The settings do not reset if you drop all cache groups for the interval.
- The `ttMigrate`, `ttBackup`, and `ttRestore` built-in procedures do not preserve the setting of `ttCacheAutorefreshXactLimit`. 
If you alter the cache group autorefresh interval, it does not modify the setting of `ttCacheAutorefreshXactLimit`.

See also

- `ttCacheAutorefIntervalStatsGet`

"Executing large transactions with incremental autorefresh read-only cache groups" in the *Oracle TimesTen Application-Tier Database Cache User's Guide*. 


ttCacheAWTMonitorConfig

Description
This procedure enables monitoring to determine the amount of time spent in each component of the workflow of an AWT cache group. To display the monitoring results, use the ttRepAdmin utility with the -awtmoninfo and -showstatus commands.

If the replication agent is restarted, monitoring is turned off. Setting the monitoring state to OFF resets the internal counters of the monitoring tool.

Run this procedure on the replication node that is replicating AWT changes to the Oracle database. If the active standby pair is functioning normally, the node replicating AWT changes is the standby. If the active is operating standalone, the node replicating AWT changes is the active.

If a failure occurs on the node where the active database resides, the standby node becomes the new active node. In that case you would run this procedure on the new active node.

Required privilege
This procedure requires the CACHE_MANAGER privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax

```
ttCacheAWTMonitorConfig [{'state'}, [samplingRate]]
```

Parameters

```
ttCacheAWTMonitorConfig has the optional parameters:
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>state</td>
<td>TT_CHAR(10)</td>
<td>Enables and disables AWT monitoring. Its value can be ON or OFF. Default is OFF</td>
</tr>
<tr>
<td>samplingRate</td>
<td>TT_INTEGER</td>
<td>Positive integer that specifies the frequency with which the AWT workflow is sampled. If samplingRate is set to 1, every AWT operation is monitored. Greater values indicate less frequent sampling. The value recommended for accuracy and performance is 16. If state is set to ON, the default for samplingRate is 16. If state is set to OFF, the default for samplingRate is 0.</td>
</tr>
</tbody>
</table>
ttCacheAWTMonitorConfig returns the following result if you do not specify any parameters. It returns an error if the replication agent is not running or if an AWT cache group has not been created.

**Result set**

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>state</td>
<td>TTVARCHAR (10) NOT NULL</td>
<td>Current state of AWT monitoring. The value can be ON or OFF.</td>
</tr>
<tr>
<td>AWTSamplingFactor</td>
<td>TT_INTEGER NOT NULL</td>
<td>Positive integer that specifies the frequency with which the AWT workflow is sampled.</td>
</tr>
</tbody>
</table>

**Examples**

**Example 3–1**
Retrieve the current state and sampling factor when monitoring is disabled.

Command> CALL ttCacheAWTMonitorConfig;
< OFF, 0 >
1 row found.

**Example 3–2**
Enable monitoring and set the sampling frequency to 16.

Command> CALL ttCacheAWTMonitorConfig ('ON', 16);
< ON, 16 >
1 row found.

**Example 3–3**
Disable monitoring.

Command> CALL ttCacheAWTMonitorConfig ('OFF')
< OFF, 0 >
1 row found.

**See also**

"ttRepAdmin" on page 5-110
ttCacheAWTThresholdGet

Description
This procedure returns the current transaction log file threshold for databases that include AWT cache groups.

Required privilege
This procedure requires no privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax

```
ttCacheAWTThresholdGet()
```

Parameters

`ttCacheAWTThresholdGet` has no parameters.

Result set

`ttCacheAWTThresholdGet` returns the result:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>threshold</td>
<td>TT_INTEGER NOT NULL</td>
<td>The number of transaction log files for all AWT cache groups associated with the database. If the result is 0, there is no set limit.</td>
</tr>
</tbody>
</table>

Examples

```
CALL ttCacheAWTThresholdGet();
```

Notes

This procedure is available only for TimesTen Cache.

See also

`ttCacheAWTThresholdSet`
ttCacheAWTThresholdSet

Description
This procedure sets the threshold for the number of transaction log files that can accumulate before AWT is considered either terminated or too far behind to catch up. This setting applies to all subscribers to the database. When the threshold is exceeded, updates are no longer sent to the Oracle database. If no threshold is set then the default is zero.

Using this built-in procedure, the threshold can be set after an AWT cache group has been created.

This setting can be overwritten by a `CREATE REPLICATION` statement that resets the Log Failure Threshold for the database.

Required privilege
This procedure requires the `CACHE_MANAGER` privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax
```
ttCacheAWTThresholdSet(threshold)
```

Parameters
`ttCacheAWTThresholdSet` has the parameter:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>threshold</td>
<td>TT_INTEGER</td>
<td>Specifies the number of transaction log files for all AWT cache groups associated with the database. If the threshold is <code>NULL</code>, the log failure threshold is set to zero.</td>
</tr>
</tbody>
</table>

Result set
`ttCacheAWTThresholdSet` returns no results.

Examples
To set the threshold to allow 12 transaction log files to accumulate, use:
```
CALL ttCacheAWTThresholdSet(12);
```

Notes
This procedure is available for TimesTen Cache. The user is responsible to recover when the threshold is exceeded.
See also

`ttCacheAWTThresholdGet`
ttCacheCheck

Description

The ttCacheCheck built-in procedure performs a check for missing constraints for cached tables on the Oracle database.

Any unique index, unique constraint, or foreign key constraint on columns in Oracle Database tables that are to be cached should also be created on asynchronous writethrough cache tables within TimesTen. If you have not created these constraints on the AWT cache tables and you have configured the cache group for parallel propagation, TimesTen serializes any transactions with DML operations to those tables with missing constraints.

This procedure provides information about missing constraints and the tables marked for serialized propagation.

Call ttCacheCheck to manually check for missing constraints, under these conditions:

- After completing a series of DROP CACHE GROUP statements.
- After creating or dropping a unique index or foreign key on the Oracle database.
- To determine why some transactions are being serialized.

This procedure updates system tables to indicate if DML executed against a table should or should not be serialized, therefore you must commit or roll back after the ttCacheCheck built-in completes.

For more details on parallel propagation, see "Configuring parallel propagation to Oracle Database tables" in the Oracle TimesTen Application-Tier Database Cache User’s Guide.

Required privilege

This procedure requires the CACHE_MANAGER privilege.

Usage in TimesTen Scaleout

This procedure is not supported in TimesTen Scaleout.

Related views

This procedure has no related views.

Syntax

```
ttCacheCheck('operation', cgOwner, cgName)
```

Parameters

ttCacheCheck has these parameters:
Result set

**ttCacheCheck** returns the result set:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>cgOwner</strong></td>
<td>TT_VARCHAR(30)</td>
<td>The owner of the cache group.</td>
</tr>
<tr>
<td><strong>cgName</strong></td>
<td>TT_VARCHAR(30)</td>
<td>The name of the cache group.</td>
</tr>
<tr>
<td><strong>tblOwner</strong></td>
<td>TT_VARCHAR(30)</td>
<td>The owner of the table.</td>
</tr>
<tr>
<td><strong>tblName</strong></td>
<td>TT_VARCHAR(30)</td>
<td>The name of the table.</td>
</tr>
<tr>
<td><strong>objectType</strong></td>
<td>TT_VARCHAR(15)</td>
<td>The type of Oracle object: unique index, constraint or foreign key.</td>
</tr>
<tr>
<td><strong>objectOwner</strong></td>
<td>TT_VARCHAR(30)</td>
<td>The owner of the Oracle object.</td>
</tr>
<tr>
<td><strong>objectName</strong></td>
<td>TT_VARCHAR(30)</td>
<td>The object name.</td>
</tr>
<tr>
<td><strong>msgType</strong></td>
<td>TT_SMALLINT NOT NULL</td>
<td>The type of message:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = Informational</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = Warning</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-1 = Error</td>
</tr>
<tr>
<td><strong>msg</strong></td>
<td>TT_VARCHAR(100000) NOT NULL</td>
<td>Message describing the issue.</td>
</tr>
<tr>
<td><strong>objectDesc</strong></td>
<td>VARCHAR2(200000)</td>
<td>A description of the object. If the object is AWT checking, the description is the SQL statement that describes the object.</td>
</tr>
</tbody>
</table>
Examples

The following example determines if there are any missing constraints for the cache group update_orders that is owned by cacheuser. A result set is returned that includes the warning message. The ordertab table in the update_orders cache group is marked for serially propagated transactions.

Command> call ttCacheCheck( NULL, 'cacheuser', 'update_orders');

< CACHEUSER, UPDATE_ORDERS, CACHEUSER, ORDERTAB, Foreign Key, CACHEUSER, CUST_FK, 1, Transactions updating this table will be serialized to Oracle because: The missing foreign key connects two AWT cache groups., table CACHEUSER.ORDERTAB constraint CACHEUSER.CUST_FK foreign key(CUSTID) references CACHEUSER.ACTIVE_CUSTOMER(CUSTID) >
1 row found.

Notes

This procedure is available only for TimesTen Cache.

See also

ttCacheDbCgStatus
ttCachePolicyGet
ttCachePolicySet
ttCacheStart
ttCacheStop
ttCacheUidGet
ttCacheUidPwdSet
"ttAdmin" on page 5-5
ttCacheConfig

Description

For all cache groups that cache data from the same Oracle instance, this procedure specifies a timeout value and recovery policies in the case that the Oracle database server is unreachable and the cache agent or database is considered terminated.

The automatic refresh state of the database and cache groups can be determined from the procedure ttCacheDbCgStatus.

Required privilege

This procedure requires the CACHE_MANAGER privilege.

Usage in TimesTen Scaleout

This procedure is not supported in TimesTen Scaleout.

Related views

This procedure has no related views.

Syntax

```
ttCacheConfig(Param, tblOwner, tblName, Value)
```

Parameters

ttCacheConfig has these parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Param</td>
<td>VARCHAR2(50)</td>
<td>Specifies the parameter to be set by Value:</td>
</tr>
<tr>
<td></td>
<td>NOT NULL</td>
<td>- AgentFailoverTimeout - When working in an Oracle RAC environment, sets the TAF timeout, in minutes. Configures how long TAF retries when establishing a connection. The default is four minutes.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- AgentTimeout - Number of seconds before a database is declared terminated if the cache agent cannot connect to the Oracle database server.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- AutoRefreshLogFragmentationWarningPCT - The percent of table usage that must occur before warning the user to compact the table.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- AutoRefreshLogDefragmentAction - Compaction mode for the specified tables.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- AutoRefreshLogTblSpaceUsagePCT - Specifies the cache administration user's tablespace usage warning threshold as a percentage.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- DeadDbRecovery - Specifies the type of autorefresh recovery when the cache agent restarts.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- TblSpaceFullRecovery - Specifies the action that TimesTen takes when the cached Oracle database table is updated and the cache administration user's tablespace is full.</td>
</tr>
</tbody>
</table>
### Built-In Procedures

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tblOwner</td>
<td>VARCHAR2(30)</td>
<td>Specifies the owner of the cached Oracle database table. This parameter is required if <code>Param</code> is set <code>TblspaceFullRecovery</code>. Do not specify <code>tblOwner</code> for other values of <code>Param</code>. A synonym cannot be used to specify a table name.</td>
</tr>
<tr>
<td>tblName</td>
<td>VARCHAR2(30)</td>
<td>Specifies the name of the cached Oracle database table. This parameter is required if <code>Param</code> is set <code>TblspaceFullRecovery</code>. Do not specify <code>tblOwner</code> for other values of <code>Param</code>. Using a synonym to specify a table name is not supported.</td>
</tr>
<tr>
<td>Value</td>
<td>VARCHAR2(200)</td>
<td>Specifies the value to be set for <code>Param</code>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- When <code>Param</code> is <code>AgentFailoverTimeout</code>, it specifies the number of minutes before TAF retries when establishing a connection, when working in an Oracle RAC environment. The default is four minutes.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- When <code>Param</code> is <code>AgentTimeout</code>, it specifies the number of seconds before a database is declared terminated if the cache agent cannot connect to the Oracle database server. The default is 0, which means that the database is never declared terminated.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- When <code>Param</code> is <code>AutoRefreshLogFragmentationWarningPCT</code>, the value of the fourth parameter must be an integer between 1 and 100, representing a percentage of the table.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- When <code>Param</code> is <code>AutoRefreshLogDeFragmentAction</code>, the value can be <code>Manual</code>, <code>CompactOnly</code> or <code>CompactandReclaim</code>. If <code>Manual</code> is specified no action is taken. The user can run <code>ttCacheAutorefreshLogDefrag</code> built-in procedure to defragment the logs. If <code>CompactOnly</code> is specified trigger log space is compacted. If <code>CompactandReclaim</code> is specified both the trigger log space and the transaction log buffer (reclaim space) are compacted. The default is <code>Manual</code>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- When <code>Param</code> is <code>AutoRefreshLogTblSpaceUsagePCT</code>, the value can be 0 to 100. The default is 0, which means no warning is returned regardless of the tablespace usage.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- When <code>Param</code> is <code>DeadDbRecovery</code>, the value can be <code>Normal</code> or <code>Manual</code>. <code>Normal</code> specifies a full automatic refresh. <code>Manual</code> specifies that <code>REFRESH CACHE GROUP</code> statement must be issued. The default is <code>Normal</code>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- When <code>Param</code> is <code>TblSpaceFullRecovery</code>, the value can be <code>Reload</code> or <code>None</code>. <code>Reload</code> specifies that rows are deleted from the change log table and a full automatic refresh is performed. <code>None</code> specifies that an Oracle database error is returned when the cached Oracle database table is updated. The default is <code>None</code>.</td>
</tr>
<tr>
<td>Or</td>
<td></td>
<td>Specifies the value to be set by <code>AwtErrorXmlOutput</code>:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ASCII - A text file that contains the AWT error report. <em>(Default)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- XML - An XML file that contains the AWT error report and the associated DTD file.</td>
</tr>
</tbody>
</table>
**Result set**

`ttCacheConfig` returns no results when an application uses it to set parameter values. When it is used to return parameter settings, `ttCacheConfig` returns the following results.

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Param</td>
<td>VARCHAR2(50)</td>
<td>Parameter name:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AgentTimeout</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AgentFailoverTimeout</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AutoRefreshLogTblSpaceUsagePCT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DeadDbRecovery</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TblSpaceFullRecovery</td>
</tr>
<tr>
<td>tblOwner</td>
<td>VARCHAR2(30)</td>
<td>Owner of the cached Oracle database table.</td>
</tr>
<tr>
<td>tblName</td>
<td>VARCHAR2(30)</td>
<td>Name of the cached Oracle database table.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Using a synonym to specify a table name is not supported.</td>
</tr>
<tr>
<td>Value</td>
<td>VARCHAR2(200)</td>
<td>Specifies the value set for Param.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ When Param is AgentTimeout, it specifies the number of seconds before a database is declared terminated if the cache agent cannot connect to the Oracle database server.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ When Param is AutoRefreshLogTblSpaceUsagePCT, the value can be 0 to 100.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ When Param is DeadDbRecovery, the value can be Normal or Manual.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ When Param is TblSpaceFullRecovery, the value can be Reload or None.</td>
</tr>
</tbody>
</table>

**Examples**

To set the cache agent timeout to 600 seconds (10 minutes), enter:

```sql
CALL ttCacheConfig('AgentTimeout',,'600');
```

To determine the current cache agent timeout setting, enter:

```sql
CALL ttCacheConfig('AgentTimeout');
< AgentTimeout, <NULL>, <NULL>, 600 >
1 row found.
```

To set the recovery method to Manual for cache groups whose automatic refresh status is dead, enter:

```sql
CALL ttCacheconfig('DeadDbRecovery',,'Manual');
```

Configure the TimesTen Cache to prevent an automatic full refresh and receive an Oracle database error when there is an update on a cached Oracle database table while the cache administration user's tablespace is full. The Oracle database table is `terry.customer`.

```sql
CALL ttCacheConfig('TblSpaceFullRecovery',,'terry', 'customer', 'None');
```

To determine the current setting for `TblSpaceFullRecovery` on the `terry.customer` cached Oracle database table, enter:
CALL ttCacheConfig('TblSpaceFullRecovery','terry','customer');
< TblSpaceFullRecovery, TERRY, CUSTOMER, none >
1 row found.

To configure a warning to be returned when the cache administration user's tablespace is 85 percent full and an update operation occurs on the cached Oracle database table, enter:

CALL ttCacheConfig('AutoRefreshLogTblSpaceUsagePCT',,,,'85');

When working in an Oracle RAC environment, the following shows how to retrieve the value of the failover timeout:

Command> call ttCacheConfig('AgentFailoverTimeout');
< AgentFailoverTimeout, <NULL>, <NULL>, 4 >
1 row found.

The following sets the failover timeout to 5 minutes:

Command> call ttCacheConfig('AgentFailoverTimeout',,,5);
< AgentFailoverTimeout, <NULL>, <NULL>, 5 >
1 row found.

Notes

This procedure is available only for TimesTen Cache.

You must call the ttCacheConfig built-in procedure from every node in an active standby pair.

See also

ttCacheDbCgStatus
ttCachePolicyGet
ttCachePolicySet
ttCacheStart
ttCacheStop
ttCacheUidGet
ttCacheUidPwdSet
"ttAdmin" on page 5-5
ttCacheDbCgStatus

Description
This procedure returns the automatic refresh status of the database and the specified cache group. If you do not specify any values for the parameters, the procedure returns the automatic refresh status for the database.

Required privilege
This procedure requires no privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax
ttCacheDbCgStatus([cgOwner], [cgName])

Parameters
ttCacheDbCgStatus has these optional parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cgOwner</td>
<td>VARCHAR2(30)</td>
<td>Specifies the user name of the cache group owner.</td>
</tr>
<tr>
<td>cgName</td>
<td>VARCHAR2(30)</td>
<td>Specifies the cache group name.</td>
</tr>
</tbody>
</table>

Result set
ttCacheDbCgStatus returns the result:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbStatus</td>
<td>VARCHAR2(20)</td>
<td>Specifies the autorefresh status of all the cache groups in the database. The status is one of: alive - The database is active. The status of all cache groups is ok. The cache agent has been in contact with the Oracle database server. dead - The cache agent was not able to contact the Oracle database within the timeout period. The status of all the cache groups with the AUTOREFRESH attribute is terminated. recovering - Some or all the cache groups with the AUTOREFRESH attribute are being resynchronized with the Oracle database server. The status of at least one cache group is recovering.</td>
</tr>
</tbody>
</table>
Examples

This example shows that the automatic refresh status of the database is alive. The automatic refresh status of the cache group is ok.

CALL ttCacheDbCgStatus ('terry', 'cgemployees');
< alive, ok >
1 row found.

To determine the automatic refresh status of the database, call ttCacheDbCgStatus with no parameters:

CALL ttCacheDbCgStatus;
< dead, <NULL> >
1 row found.

Notes

This procedure is available only for TimesTen Cache.

See also

ttCacheConfig
ttCachePolicyGet
ttCachePolicySet
ttCacheStart
ttCacheStop
ttCacheUidGet
ttCacheUidPwdSet
"ttAdmin" on page 5-5
ttCacheDDLTrackingConfig

This procedure enables or disables tracking of DDL statements issued on cached Oracle database tables. By default, DDL statements are not tracked.

DDL tracking saves the change history for all the cached Oracle database tables. One DDL tracking table is created to store DDL statements issued on any cached Oracle database table. You can use this information to diagnose autorefresh problems.


Required privilege
This procedure requires the CACHE_MANAGER privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax
```
ttCacheDDLTrackingConfig('trackingStatus')
```

Parameters
`ttCacheDDLTrackingConfig` has the parameter:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>trackingStatus</td>
<td>TT_VARCHAR(10)</td>
<td>Specifies whether DDL statements issued on cached Oracle database tables are tracked. Valid values are: enable - Enables tracking, disable (default) - Disables tracking.</td>
</tr>
</tbody>
</table>

Result set
`ttCacheDDLTrackingConfig` returns no results.

Examples
```
Command> CALL ttCacheDDLTrackingConfig('enable');
```
ttCachePolicyGet

Description
This procedure returns the current policy used to determine when the TimesTen cache agent for the connected database should run. The policy can be either always or manual.

Required privilege
This procedure requires no privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax
`ttCachePolicyGet()`

Parameters
`ttCachePolicyGet` has no parameters.

Result set
`ttCachePolicyGet` returns the result:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>cachePolicy</td>
<td>TT_VARCHAR(10)</td>
<td>Specifies the policy used to determine when the TimesTen cache agent for the database should run. Valid values are: always - Specifies that the agent for the database is always running. This option immediately starts the TimesTen cache agent. When the TimesTen daemon restarts, TimesTen automatically restarts the cache agent. manual (default) - Specifies that you must manually start the cache agent using either the <code>ttCacheStart</code> built-in procedure or the <code>ttAdmin -cacheStart</code> command. You must explicitly stop the cache agent using either the <code>ttCacheStop</code> built-in procedure or the <code>ttAdmin -cacheStop</code> command.</td>
</tr>
</tbody>
</table>

Examples
To get the current policy for the TimesTen cache agent, use:

```sql
CALL ttCachePolicyGet ();
```

Notes
This procedure is available only for TimesTen Cache.
See also

ttCacheConfig
ttCacheDbCgStatus
ttCachePolicySet
ttCacheStart
ttCacheStop
ttCacheUidGet
ttCacheUidPwdSet
"ttAdmin" on page 5-5
**ttCachePolicySet**

**Description**

The procedure defines the policy used to determine when the TimesTen cache agent for the connected database should run. The policy can be either **always** or **manual**.

**Required privilege**

This procedure requires the `CACHE_MANAGER` privilege.

**Usage in TimesTen Scaleout**

This procedure is not supported in TimesTen Scaleout.

**Related views**

This procedure has no related views.

**Syntax**

```
CALL ttCachePolicySet('cachePolicy');
```

**Parameters**

`ttCachePolicySet` has these parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| cachePolicy | TT_VARCHAR(10) NOT NULL | Specifies the policy used to determine when the TimesTen cache agent for the database should run. Valid values are:  
always - Specifies that the agent for the database is always running. This option immediately starts the TimesTen cache agent. When the TimesTen daemon restarts, TimesTen automatically restarts the cache agent.  
manual (default) - Specifies that you must manually start the cache agent using either the `ttCacheStart` built-in procedure or the `ttAdmin -cacheStart` command. You must explicitly stop the cache agent using either the `ttCacheStop` built-in procedure or the `ttAdmin -cacheStop` command.  
norestart - Specifies that the cache agent for the database is not to be restarted after a failure. |

**Result set**

`ttCachePolicySet` returns no results.

**Examples**

To set the policy for TimesTen cache agent to **always**, use:

```
CALL ttCachePolicySet ('always');
```
Notes

This procedure is available only for TimesTen Cache.

Always specify the TimesTen database location as a full path. If a relative path is specified, TimesTen would look relative to the working directory of the daemon, `timesten_home/info`.

Successfully setting the policy to `always` automatically starts the cache agent if it was stopped.

See also

- `ttCacheConfig`
- `ttCacheDbCfgStatus`
- `ttCachePolicyGet`
- `ttCacheStart`
- `ttCacheStop`
- `ttCacheUidGet`
- `ttCacheUidPwdSet`
- "ttAdmin" on page 5-5
ttCachePropagateFlagSet

Description
This procedure enables you to disable propagation of committed updates (the result of executing DML statements) within the current transaction to the Oracle database. Any updates from executing DML statements after the flag is set to zero are never propagated to the back-end Oracle database. Thus, these updates exist only on the TimesTen database. You can then re-enable propagation for DML statements by resetting the flag.

Required privilege
This procedure requires the CACHE_MANAGER privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax
\[
\texttt{ttCachePropagateFlagSet(CommitsOn)}
\]

Parameters
\texttt{ttCachePropagateFlagSet} has the parameter:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CommitsOn</td>
<td>TT_INTEGER NOT NULL</td>
<td>If 0, sets a flag to stop updates from being sent to the Oracle database. The flag remains set until the end of the transaction or until the procedure is set to 1. If 1, updates are sent to the Oracle database.</td>
</tr>
</tbody>
</table>

Result set
\texttt{ttCachePropagateFlagSet} returns no results.

Notes
This procedure is available only for TimesTen Cache.
If the value of \texttt{ttCachePropagateFlagSet} is reenabled several times during a single transaction, the transaction is only partially propagated to the Oracle database.
\texttt{ttCachePropagateFlagSet} is the only built-in procedure that applications can use in the same transaction as any of the other cache group operation, such as FLUSH, LOAD, REFRESH and UNLOAD.
The propagate flag is reset after a commit or rollback.
When using this procedure, it is important to turn off AutoCommit, otherwise after the procedure is called the transaction ends and propagation to the Oracle database is turned back on.
Examples

This example sets autocommit off to prevent the propagation flag from toggling from off to on after a commit. Calls the `ttCachePropagateFlagSet` to turn off propagation. A row is inserted into the TimesTen Cache detail table for `oratt.writetab`. Then, propagation is reenabled by calling the `ttCachePropagateFlagSet` built-in procedure and setting the flag to one.

```
Command> set autocommit off;
       call ttCachePropagateFlagSet(0);
       INSERT INTO oratt.writetab VALUES (103, 'Agent');
1 row inserted.
Command> COMMIT;
Command> SELECT * FROM oratt.writetab;
< 100, Oracle >
< 101, TimesTen >
< 102, Cache >
< 103, Agent >
4 rows found.
Command> call ttCachePropagateFlagSet(1);
```

When you select all rows on the Oracle database, the row inserted when propagation was turned off is not present in the `oratt.writetab` table on Oracle.

```
Command> set passthrough 3;
       SELECT * FROM oratt.writetab;
< 100, Oracle >
< 101, TimesTen >
< 102, Cache >
3 rows found.
```
ttCacheSqlGet

Description
This procedure generates the Oracle SQL statements to install or uninstall Oracle database objects for:

- Read-only cache groups
- User managed cache groups with incremental autorefresh
- Asynchronous writethrough (AWT) cache groups

This is useful when the user creating the cache group does not have adequate privilege to write on the Oracle database. The Oracle DBA can then use the script generated by this built-in procedure to create the Oracle database objects.

Required privilege
This procedure requires the CACHE_MANAGER privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax
```
ttCacheSqlGet('feature_name', 'cache_group_name', install_flag)
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>feature_name</td>
<td>TT_VARCHAR (100)</td>
<td>Can be specified as INCREMENTAL_AUTOREFRESH or ASYNCHRONOUS_WRITETHROUGH.</td>
</tr>
<tr>
<td>cache_group_name</td>
<td>TT_VARCHAR (100)</td>
<td>The name of the cache group. Specify NULL when installing objects for asynchronous writethrough cache groups or to uninstall all Oracle database objects in the autorefresh user's account.</td>
</tr>
<tr>
<td>install_flag</td>
<td>TT_INTEGER NOT NULL</td>
<td>If install_flag is 1, ttCacheSqlGet returns Oracle SQL to install the autorefresh or asynchronous writethrough Oracle database objects. If install_flag is 0, ttCacheSqlGet returns SQL to uninstall the previously created objects.</td>
</tr>
</tbody>
</table>

Result set

ttCacheSqlGet returns the result set:
CALL ttCacheSqlGet('INCREMENTAL_AUTOREFRESH', 'westernCustomers', 1);

To remove all Oracle database objects in the autorefresh user's account, use:
CALL ttCacheSqlGet('INCREMENTAL_AUTOREFRESH', NULL, 0);

### Notes

This procedure is available only for TimesTen Cache.

Each returned `retval` field contains a separate Oracle SQL statement that may be directly executed on the Oracle database. A row may end in the middle of a statement, as indicated by the `continueFlag` field. In this case, the statement must be concatenated with the previous row to produce a usable SQL statement.

The script output of this procedure is not compatible with Oracle's SQL*Plus utility. However, you can use the `ttIsql cachenget` command to generate a script that is compatible with the SQL*Plus utility.

You can specify `NULL` for the `cache_group_name` option to generate Oracle SQL to clean up Oracle database objects after a database has been destroyed by the `ttDestroy` utility.

### Examples

CALL ttCacheSqlGet('INCREMENTAL_AUTOREFRESH', 'westernCustomers', 1);

To remove all Oracle database objects in the autorefresh user's account, use:
CALL ttCacheSqlGet('INCREMENTAL_AUTOREFRESH', NULL, 0);
**ttCacheStart**

**Description**
This procedure starts the TimesTen cache agent for the connected database.

**Required privilege**
This procedure requires the `CACHE_MANAGER` privilege.

**Usage in TimesTen Scaleout**
This procedure is not supported in TimesTen Scaleout.

**Related views**
This procedure has no related views.

**Syntax**
```
ttCacheStart()
```

**Parameters**
`ttCacheStart` has no parameters.

**Result set**
`ttCacheStart` returns no results.

**Examples**
To start the TimesTen cache agent, use:
```
CALL ttCacheStart ();
```

**Notes**
This procedure is available only for TimesTen Cache.

The cache administration user ID and password must be set with the `ttCacheUidPwdSet` built-in procedure before starting the cache agent when there are or might be autorefresh or asynchronous writethrough cache groups in the database.

If you attempt to start the TimesTen cache agent (by changing the policy from manual to always) for a database with a relative path, TimesTen looks for the database relative to where the TimesTen Data Manager is running, and fails. For example, on Windows, if you specify the path for the database as `DataStore=./payroll` and attempt to start the TimesTen cache agent with this built-in procedure, the agent is not started because TimesTen Data Manager looks for the database in the `\srv` directory. On UNIX and Linux systems, the TimesTen Data Manager looks in the `/var/TimesTen/instance` directory.

When using this procedure, no application, including the application making the call, can be holding a connection that specifies database-level locking (`LockLevel=1`).

**See also**

`ttCacheConfig`
ttCacheStart

- ttCacheDbCgStatus
- ttCachePolicyGet
- ttCachePolicySet
- ttCacheStop
- ttCacheUidPwdSet
- ttCacheUidGet

"ttAdmin" on page 5-5
ttCacheStop

Description
This procedure stops the TimesTen cache agent for the connected database.

Required privilege
This procedure requires the CACHE_MANAGER privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax
```
ttCacheStop(timeout)
```

Parameters
`ttCacheStop` has the parameter:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>timeout</td>
<td>TT_INTEGER</td>
<td>Specifies that the TimesTen daemon should stop the cache agent if it does not stop within <code>timeout</code> seconds. If set to 0, the daemon waits forever for the cache agent. The default value is 100.</td>
</tr>
</tbody>
</table>

Result set
`ttCacheStop` returns no results.

Examples
To stop the TimesTen cache agent, use:
```
CALL ttCacheStop();
```

Notes
This procedure is available only for TimesTen Cache.

Do not shut down the cache agent immediately after dropping or altering a cache group. Instead, wait for at least two minutes. Otherwise, the cache agent may not get a chance to clean up the Oracle database objects that were used by the AUTOREFRESH feature.

When using this procedure, no application, including the application making the call, can be holding a connection that specifies database-level locking (`LockLevel=1`).

See also
`ttCachePolicySet`
`ttCacheStart`
ttCacheUidPwdSet

ttCacheUidGet

"ttAdmin" on page 5-5
ttCacheUidGet

Description
This procedure returns the cache administration user ID for the database. If the cache administration user ID and password have not been set for the database with the ttCacheUidPwdSet built-in procedure, ttCacheUidGet returns NULL.

Required privilege
This procedure requires CACHE_MANAGER privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax
```
ttCacheUidGet()
```

Parameters
ttCacheUidGet has no parameters.

Result set
ttCacheUidGet returns the results:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UID</td>
<td>TT_VARCHAR (30)</td>
<td>The current cache administration user ID, used for autorefresh and asynchronous writethrough cache groups.</td>
</tr>
</tbody>
</table>

Examples
```
CALL ttCacheUidGet();
```

Notes
This procedure is available only for TimesTen Cache.

See also
```
ttCacheUidPwdSet
"ttAdmin" on page 5-5
```
**ttCacheUidPwdSet**

**Description**
This procedure sets the cache administration user ID and password. You only need to specify the cache administration user ID and password once for each new database. The cache administration password can be changed at any time.

**Required privilege**
This procedure requires the `CACHE_MANAGER` privilege.

**Usage in TimesTen Scaleout**
This procedure is not supported in TimesTen Scaleout.

**Related views**
This procedure has no related views.

**Syntax**
```
ttCacheUidPwdSet('UID', 'PWD')
```

**Parameters**
`ttCacheUidPwdSet` has these parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UID</td>
<td>TT_VARCHAR (30)</td>
<td>The cache administration user ID, used for autorefresh and asynchronous writethrough cache groups.</td>
</tr>
<tr>
<td>PWD</td>
<td>TT_VARCHAR (30)</td>
<td>The password for the cache administration user.</td>
</tr>
</tbody>
</table>

**Result set**
`ttCacheUidPwdSet` returns no results.

**Examples**
```
CALL ttCacheUidPwdSet('myid', 'mypwd');
```

**Notes**
This procedure cannot be called from a Client/Server connection.
This procedure is available only for TimesTen Cache.

For all levels of `DDLReplicationLevel`, you can set the cache administration user ID and password while the cache or replication agents are running. For more details on changing the cache administration user ID or password, see "Changing cache user names and passwords" in the *Oracle TimesTen Application-Tier Database Cache User's Guide*.

The cache administration user ID cannot be reset while there are cache groups on the database. The cache administration password can be changed at any time.
See also

- `ttCacheUidGet`
- `ttCacheUidPwdSet`
- "ttAdmin" on page 5-5
**ttCkpt**

**Description**

This procedure performs a nonblocking checkpoint operation. For information on blocking checkpoints, see “ttCkptBlocking” on page 3-78. A checkpoint operation makes a record of the current state of the database on disk and to purge transaction log files. A nonblocking checkpoint does not require any locks on the database.

Applications should checkpoint databases periodically either by setting the background checkpointing attributes (CkptFrequency and CkptLogVolume) or by explicitly calling this procedure. Applications can call this procedure asynchronously to any other application running on the database.

By default, TimesTen performs background checkpoints at regular intervals.

In the case that your application attempts to perform a checkpoint operation while a backup is in process, the backup waits until the checkpoint finishes. Regardless of whether the checkpoint is a background checkpoint or an application-requested checkpoint, the behavior is:

- If a backup or checkpoint is running and you try to do a backup, it waits for the running backup or checkpoint to finish.
- If a backup or checkpoint is running and you try to do a checkpoint, it does not wait. It returns an error immediately.

To turn off background checkpointing, set CkptFrequency=0 and CkptLogVolume=0.

When a database crashes, and the checkpoints on disk are nonblocking checkpoints, TimesTen uses the log to recover.

**Required privilege**

This procedure requires the ADMIN privilege.

**Usage in TimesTen Scaleout**

This procedure is not supported in TimesTen Scaleout.

**Related views**

This procedure has no related views.

**Syntax**

```sql
ttCkpt([timeout], [retries])
```

**Parameters**

ttCkpt has these optional parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>timeout</td>
<td>TT_INTEGER</td>
<td>The time (in seconds) that ttCkpt should wait to get a database lock before timing out. The value of timeout can be between 0 and one million, inclusively. If not specified, the checkpoint never times out.</td>
</tr>
</tbody>
</table>
Result set

\texttt{ttCkpt} returns no results.

Examples

\begin{verbatim}
CALL ttCkpt();
\end{verbatim}

Notes

For a description of checkpoints, see "Transaction Management" in \textit{Oracle TimesTen In-Memory Database Operations Guide}.

See also

\texttt{ttCkptBlocking}
\texttt{ttCkptConfig}
\texttt{ttCkptHistory}
ttCkptBlocking

Description
This procedure performs a blocking checkpoint operation. A checkpoint operation makes a record of the current state of the database on disk, and to purge transaction log files. This checkpoint requires exclusive access to the database, and so may cause other applications to be blocked from the database while the checkpoint is in progress.

When this procedure is called, TimesTen performs a blocking checkpoint when the current transaction is committed or rolled back. If, at that time, other transactions are in progress, the checkpointing connection waits until the other transactions have committed or rolled back. While the checkpoint connection is waiting, any other new transactions that should start will form a queue behind the checkpointing transaction. As a result, if any transaction is long-running, it may cause many other transactions to be held up. So, use this blocking checkpoint with caution. To perform a nonblocking checkpoint, use the ttCkpt procedure.

No log is needed to recover when blocking checkpoints are used. TimesTen uses the log, if present, to bring the database up to date after recovery.

Required privilege
This procedure requires the ADMIN privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax
	ttCkptBlocking([timeout], [retries])

Parameters
ttCkptBlocking has these optional parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>timeout</td>
<td>TT_INTEGER</td>
<td>The time (in seconds) that ttCkptBlocking should wait to get a database lock before timing out. The value of timeout can be between 0 and one million, inclusively. If not specified, the checkpoint never times out.</td>
</tr>
<tr>
<td>retries</td>
<td>TT_INTEGER</td>
<td>The number of times that ttCkptBlocking should attempt to get a database lock, if timeouts occur. The value of retries can be between 0 and 10, inclusive. If not specified, defaults to zero.</td>
</tr>
</tbody>
</table>

Result set
ttCkptBlocking returns no results.
Examples

CALL ttCkptBlocking();
CALL ttCkptBlocking(1,10);

Notes

Because the checkpoint takes place at commit or rollback, the call to `ttCkptBlocking` always succeed. At commit or rollback, any problems with the checkpoint operation, such as a lack of disk space or a timeout, result in a warning being returned to the application. Checkpoint problems are not reflected as errors, since the commit or rollback of which they are a part can succeed even if the checkpoint fails. Warnings are reflected in ODBC with the return code `SQL_SUCCESS_WITH_INFO`.

For more information on checkpoints, see "Transaction Management" in Oracle TimesTen In-Memory Database Operations Guide.

See also

- `ttCkpt`
- `ttCkptConfig`
- `ttCkptHistory`
**ttCkptConfig**

**Description**
This procedure reconfigures the background checkpointer dynamically or returns the currently active settings of the configuration parameters. Changes made using ttCkptConfig become effective immediately. Thus, changes to **ckptRate** can take effect on a checkpoint that is currently in progress.

Changes made to the background checkpointer using ttCkptConfig are persistent. Subsequent loads of the database retain the new settings, unless the **CkptFrequency** and **CkptLogVolume** connection attributes are specified in the DSN or connection string, in which case the attribute values are used instead.

**Required privilege**
This procedure requires no privilege to query the current values. It requires the **ADMIN** privilege to change the current values.

**Usage in TimesTen Scaleout**
TimesTen Scaleout applications can call this built-in procedure.
This procedure returns a row for the element from which it was called.

**Related views**
This procedure has these related views.

```sql
SYS.GV$CKPT_CONFIG
SYS.V$CKPT_CONFIG
```

**Syntax**

```sql
ttCkptConfig(ckptFrequency, ckptLogVolume, ckptRate)
```

**Parameters**
**ttCkptConfig** has these parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ckptFrequency</strong></td>
<td><strong>TT_INTEGER</strong></td>
<td>Checkpoint frequency in seconds. Values from 0 to MAXINT are allowed. A value of 0 means that checkpoint frequency is not considered when scheduling checkpoints.</td>
</tr>
<tr>
<td><strong>ckptLogVolume</strong></td>
<td><strong>TT_INTEGER</strong></td>
<td>Log volume between checkpoints in megabytes. Values from 0 to MAXINT are allowed. A value of 0 means that checkpoint log volume is not considered when scheduling checkpoints.</td>
</tr>
<tr>
<td><strong>ckptRate</strong></td>
<td><strong>TT_INTEGER</strong></td>
<td>Specifies the rate in MB per second at which a checkpoint should be written to disk. A value of 0 indicates that the rate should not be limited, a value of NULL means that the rate should be left unchanged. Changes to this parameter take effect even on a checkpoint that is currently in-progress.</td>
</tr>
</tbody>
</table>
ttCkptConfig returns the following results.

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ckptFrequency</td>
<td>TT_INTEGER NOT NULL</td>
<td>Currently active setting for checkpoint frequency in seconds.</td>
</tr>
<tr>
<td>ckptLogVolume</td>
<td>TT_INTEGER NOT NULL</td>
<td>Currently active setting for log volume between checkpoints in MB.</td>
</tr>
<tr>
<td>ckptRate</td>
<td>TT_INTEGER NOT NULL</td>
<td>Current rate at which TimesTen writes checkpoints to disk.</td>
</tr>
</tbody>
</table>

Examples

To view the current settings of the background checkpointer configuration parameters, use:

CALL ttCkptConfig;
< 600, 32, 0 >
1 row found.

To stop the background checkpointer from initiating checkpoints unless the log reaches its limit, use:

CALL ttCkptConfig(0);
< 0, 32, 0 >
1 row found.

To stop the background checkpointer from initiating checkpoints, use:

CALL ttCkptConfig(0, 0);
< 0, 0, 0 >
1 row found.

To set the background checkpointer configuration to initiate a checkpoint every 600 seconds or to checkpoint when the log reaches 32 MB (whichever comes first), use:

CALL ttCkptConfig(600, 32);
< 600, 32, 0 >
1 row found.

Notes

By default, TimesTen performs background checkpoints at regular intervals.

In the case that your application attempts to perform a checkpoint operation while a backup is in process, the backup waits until the checkpoint finishes. Regardless of whether the checkpoint is a background checkpoint or an application-requested checkpoint, the behavior is:

- If a backup or checkpoint is running and you try to do a backup, it waits for the running backup or checkpoint to finish.
- If a backup or checkpoint is running and you try to do a checkpoint, it does not wait. It returns an error immediately.

To turn off background checkpointing, set CkptFrequency=0 and CkptLogVolume=0.

See also

CkptFrequency
CkptLogVolume
ttCkpt
ttCkptHistory
ttCkptHistory

Description
This procedure returns information about the last eight checkpoints of any type. Also see "Displaying checkpoint history and status" in Oracle TimesTen In-Memory Database Operations Guide.

Required privilege
This procedure requires no privilege.

Usage in TimesTen Scaleout
TimesTen Scaleout applications can call this built-in procedure. This procedure returns a row for the element from which it was called.

Related views
This procedure has these related views.
SYS.GV$CKPT_HISTORY
SYS.V$CKPT_HISTORY

Syntax
ttCkptHistory( )

Parameters
ttCkptHistory has no parameters.

Result set
ttCkptHistory returns the result set:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>startTime</td>
<td>TT_TIMESTAMP NOT NULL</td>
<td>Time when the checkpoint was begun.</td>
</tr>
<tr>
<td>endTime</td>
<td>TT_TIMESTAMP</td>
<td>Time when the checkpoint completed.</td>
</tr>
<tr>
<td>type</td>
<td>TT_CHAR (16) NOT NULL</td>
<td>The type of checkpoint taken. Value is one of:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Static - Automatically taken at database creation and at last disconnect.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Blocking - Transaction-consistent checkpoint.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fuzzy - nonblocking checkpoint. The background checkpointer performs this type if possible.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>None - For temporary databases, which have no checkpoint files.</td>
</tr>
</tbody>
</table>
### Column Types and Descriptions

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| status       | TT_CHAR (16) NOT NULL    | Result status of the checkpoint operation. Value is one of:  
  - In Progress - The checkpoint is currently in progress. Only the most recent result row can have this status.  
  - Completed - The checkpoint completed successfully.  
  - Failed - The checkpoint failed. Only the most recent result row can have this status. In this case the error column indicates the reason for the failure. |
| initiator    | TT_CHAR (16) NOT NULL    | The source of the checkpoint request. Value is one of:  
  - User - A user-level application. This includes TimesTen utilities such as ttIsql.  
  - Checkpointer - The background checkpoint.  
  - Subdaemon - The managing subdaemon of the database. For a shared database, the final disconnect checkpoint is taken by the subdaemon. |
| reason       | TT_CHAR (16) NOT NULL    | The reason for this checkpoint. For example: after database creation, after recovery, final checkpoint after shutdown, after the user runs a built-in procedure, or after a flush operation. |
| error        | TT_INTEGER               | If a checkpoint fails, this column indicates the reason for the failure. The value is one of the TimesTen error numbers. |
| ckptFileNum  | TT_INTEGER NOT NULL      | The database file number used by the checkpoint. This corresponds to the number in the checkpoint file extension datastore.ds0 or datastore.ds1. |
| ckptLFN      | TT_INTEGER               | The transaction log file number of the checkpoint log record. |
| ckptLFO      | TT_BIGINT                | The transaction log file offset of the checkpoint log record. |
| blksTotal    | TT_BIGINT                | The number of permanent blocks currently allocated in the database. These blocks are subject to consideration for checkpointing. |
| bytesTotal   | TT_BIGINT                | The number of bytes occupied by blksTotal. |
| blksInUse    | TT_BIGINT                | Of blksTotal, the number of blocks currently in use. |
| bytesInUse   | TT_BIGINT                | The number of bytes occupied by blksInUse. |
| blksDirty    | TT_BIGINT                | The number of dirty blocks written by this checkpoint. |
| bytesDirty   | TT_BIGINT                | The number of bytes occupied by blksDirty. |
| bytesWritten | TT_BIGINT                | The total number of bytes written by this checkpoint. |
### Examples

This example shows a checkpoint in progress:

```sql
< 2019-02-05 16:56:34.169520, <NULL>,
  Fuzzy           , In Progress     , User       ,
  BuiltIn         , <NULL>,
  0, <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL>,
  <NULL>, <NULL>, <NULL>, 13, 6, 0, <NULL>, <NULL> >

< 2019-02-05 16:55:47.703199, <NULL>, <NULL>,
  Fuzzy           , Completed       , Checkpointer    , <NULL>,
  Background      , <NULL>,
  1, 0, 8964304, 294, 33554432, 291, 5677288, 291, 1019512, 5677288, <NULL>, 5, 0, Checkpoint, <NULL> >

< 2019-02-05 16:54:47.703199, <NULL>, <NULL>,
  Fuzzy           , Completed       , Checkpoint, <NULL>,
  Background      , <NULL>,
  1, 0, 8964304, 294, 33554432, 291, 5677288, 291, 1019512, 5677288, <NULL>, 5, 0, Checkpoint, <NULL> >

< 2019-02-05 16:54:41.633792, <NULL>, <NULL>,
  Blocking        , Completed       , User       ,
  BuiltIn         , <NULL>,
  1, 0, 8958160, 294, 33554432, 291, 5677288, 256, 33157172, 5321548, <NULL>, 4, 0, Checkpoint, <NULL> >

< 2019-02-05 16:54:41.633792, <NULL>, <NULL>,
  Blocking        , Completed       , User       ,
  BuiltIn         , <NULL>,
  1, 0, 8958160, 294, 33554432, 291, 5677288, 256, 33157172, 5321548, <NULL>, 4, 0, Checkpoint, <NULL> >

< 2019-02-05 16:54:37.438827, <NULL>, <NULL>,
  Blocking        , Completed       , User       ,
  BuiltIn         , <NULL>,
  1, 0, 8958160, 294, 33554432, 291, 5677288, 256, 33157172, 5321548, <NULL>, 4, 0, Checkpoint, <NULL> >
```

If there is an in-progress checkpoint, indicates the percentage of the checkpoint that has been completed. If no checkpoint is in-progress, the value is NULL. The returned value is calculated by comparing the block ID of the last-written block against the database's `PermSize`. The value does not necessarily indicate the precise time remaining to complete the checkpoint, although it does give some indication of the remaining time needed to complete the disk write. The field shows only the progress of the writing of dirty blocks and does not include additional bookkeeping at the end of the checkpoint. The value is non-NULL if you call this procedure while a checkpoint is in progress.

### Columns

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percent_Complete</td>
<td>TT_INTEGER</td>
<td>If there is an in-progress checkpoint, indicates the percentage of the checkpoint that has been completed. If no checkpoint is in-progress, the value is NULL. The returned value is calculated by comparing the block ID of the last-written block against the database's <code>PermSize</code>. The value does not necessarily indicate the precise time remaining to complete the checkpoint, although it does give some indication of the remaining time needed to complete the disk write. The field shows only the progress of the writing of dirty blocks and does not include additional bookkeeping at the end of the checkpoint. The value is non-NULL if you call this procedure while a checkpoint is in progress.</td>
</tr>
<tr>
<td>ckptVNo</td>
<td>TT_INTEGER NOT NULL</td>
<td>The checkpoint sequence number that is incremented for each checkpoint.</td>
</tr>
<tr>
<td>logsPurged</td>
<td>TT_BIGINT</td>
<td>The number of log files purged by this checkpoint.</td>
</tr>
<tr>
<td>bookmarkName</td>
<td>TT_VARCHAR (30) INLINE</td>
<td>The name of the log hold up to which this checkpoint purged log records. For example: Backup, Checkpoint, or Oldest Transaction Undo.</td>
</tr>
<tr>
<td>additional_details</td>
<td>TT_VARCHAR (1000)</td>
<td>Additional information provided for this checkpoint, such as error codes and timestamps for multiple failed checkpoints.</td>
</tr>
</tbody>
</table>
This example shows that an error occurred during the most recent checkpoint attempt, which was a user-initiated checkpoint:

```sql
  Fuzzy           , Failed , User            ,
  BuiltIn         , 847,
  1, <NULL>, <NULL>, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, <NULL>, 7, 0, <NULL>,
```

This example selects specific columns from the checkpoint history:

```sql
select type, reason, bookmarkname, logsPurged from ttCkptHistory;
```

```sql
< Fuzzy           , BuiltIn         , Oldest Transaction Undo, 0 >
< Static          , FinalCkpt       , Checkpoint, 6 >
```
< Blocking , BuiltIn , Checkpoint, 0 >
< Blocking , BuiltIn , Checkpoint, 0 >
< Blocking , BuiltIn , Checkpoint, 0 >
< Blocking , BuiltIn , Backup, 5 >
< Blocking , BuiltIn , Backup, 0 >
< Blocking , BuiltIn , Backup, 0 >

The bottom (oldest) checkpoints could not purge log files because there was a log hold set by incremental backup, but eventually the log hold moved and five log files could be purged.

Notes

Results are ordered by start time, with the most recent first.
A failed row is overwritten by the next checkpoint attempt.

See also

ttCkpt
ttCkptBlocking
ttCommitBufferStats

Description
This built-in procedure returns the number of commit buffer overflows and the high watermark for memory used by transaction reclaim records during transaction commit process.

The information provided by the results of this procedure call is useful information when you want to explicitly set the maximum size of commit buffer, using the CommitBufferSizeMax connection attribute or the ALTER SESSION SQL statement, described in Oracle TimesTen In-Memory Database SQL Reference. This procedure helps you choose the right size for the reclaim buffer, based on the number of overflows and the maximum memory used by the reclaim records.

If there are buffer overflows, you may consider increasing the commit buffer maximum size. If there are no overflows and the highest amount of memory usage is well under the commit buffer maximum size, you may consider decreasing the maximum size.

For more information on reclaim operations, including details about setting the commit buffer size, see “Transaction reclaim operations” in the Oracle TimesTen In-Memory Database Operations Guide.

Required privilege
This procedure requires no privileges.

Usage in TimesTen Scaleout
TimesTen Scaleout applications can call this built-in procedure.

This procedure returns a row for the element from which it was called. To see information about other elements, query the SYS.GV$COMMIT_BUFFER_STATS system table.

Related views
This procedure has these related views.

SYS.GV$COMMIT_BUFFER_STATS
SYS.V$COMMIT_BUFFER_STATS

Syntax

```sql
ttCommitBufferStats()
```

Parameters

`ttCommitBufferStats` takes no parameters.

Result set

`ttCommitBufferStats` returns these results:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>overflows</td>
<td>TT_BIGINT NOT NULL</td>
<td>Total number of commit buffer overflows.</td>
</tr>
</tbody>
</table>
### Examples

This shows the result for a session where there have been no commit buffer overflows and the transaction commit buffer is set to 500 MB.

```
Command> ALTER SESSION SET COMMIT_BUFFER_SIZE_MAX = 500;
Session altered.
Command> CALL ttCommitBufferStats( );
< 0, 524288000 >
1 row found
```

For a session where there have been 10 commit buffer overflows and the transaction commit buffer is set to 2 MB, the output of this procedure is:

```
Command> ALTER SESSION SET COMMIT_BUFFER_SIZE_MAX = 2;
Session altered.
Command> CALL ttCommitBufferStats( );
< 10, 2097152 >
1 row found
```

### Notes

When you call the built-in procedure `ttCommitBufferStatsReset`, the commit buffer statistics are expressed in bytes. However, the `ttConfiguration` output and the value set by the connection attribute `CommitBufferSizeMax` are expressed in MB.

### See also

`ttCommitBufferStatsReset`
ttCommitBufferStatsReset

Description
The ttCommitBufferStatsReset procedure resets transaction commit buffer statistics to 0. This is useful, for example, if you have set a new value for the commit buffer maximum size and want to restart the statistics.

For more information on reclaim operations, including details about setting the commit buffer size, see "Transaction reclaim operations" in the Oracle TimesTen In-Memory Database Operations Guide.

Required privilege
This procedure requires no privileges.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has these related views.
SYS.GV$CONFIGURATION
SYS.V$CONFIGURATION

Syntax
```
ttCommitBufferStatsReset()
```

Parameters
```
ttCommitBufferStatsReset takes no parameters.
```

Result set
```
ttCommitBufferStatsReset returns no result set.
```

Examples
```
CALL ttCommitBufferStatsReset;
```

See also
```
 ttCommitBufferStats
```
**ttCompact**

**Description**

This procedure compacts both the permanent and temporary data partitions of the database.

`ttCompact` merges adjacent blocks of free space, but does not move any items that are allocated. Therefore, fragmentation that is caused by small unallocated blocks of memory surrounded by allocated blocks of memory is not eliminated by using `ttCompact`.

**Required privilege**

This procedure requires the `ADMIN` privilege.

**Usage in TimesTen Scaleout**

TimesTen Scaleout applications can call this built-in procedure.

This procedure executes on all elements in the grid.

**Related views**

This procedure has no related views.

**Syntax**

```sql
ttCompact()
```

**Parameters**

`ttCompact` has no parameters.

**Result set**

`ttCompact` returns no results.

**Examples**

```sql
CALL ttCompact;
```

**Notes**

Compacting data does not modify result addresses.
**ttComputeTabSizes**

**Description**

The `ttComputeTabSizes` built-in procedure refreshes table size statistics stored in TimesTen system tables. After calling this built-in procedure, you can review the statistics updates by querying the `DBA_TAB_SIZES`, `USER_TAB_SIZES` or `ALL_TAB_SIZES` view.

This procedure computes the different types of storage allocated for the specified table, such as the amount of storage allocated for inline row storage, dictionary tables, out-of-line buffers and system usage. If no table is specified, the procedure computes the sizes for all tables on which the user has `SELECT` privileges.

The execution of this built-in behaves like a DDL statement: the transaction commits just before the procedure begins and commits again upon its successful termination.

**Required privilege**

This procedure requires the `SELECT` privilege on the specified table.

**Usage in TimesTen Scaleout**

TimesTen Scaleout applications can call this built-in procedure.

This procedure executes on all elements in the grid.

**Related views**

This procedure has no related views.

**Syntax**

```
null
```

**Parameters**

`ttComputeTabSizes` has the parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tblName</td>
<td><code>TT_CHAR(61)</code></td>
<td>Name of an application table. Can include the table owner. If a value of <code>NULL</code> or an empty string is provided, updates the statistics for all the current tables. The type of tables that can be estimated are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- User tables, including cache group tables</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Materialized views</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- System tables</td>
</tr>
</tbody>
</table>
**ttComputeTabSizes**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>includeOutOfLine</td>
<td>TT_INTEGER</td>
<td>0 (no) or 1 (yes). Default is 1 (yes). If value is 0 (no), the procedure does not compute the size of out-of-line values for any table that has out-of-line columns. The out-of-line fields are displayed as NULL. Avoiding the computation of out-of-line values significantly decreases the latency of this procedure.</td>
</tr>
</tbody>
</table>

**Result set**

`ttComputeTabSizes` returns no results.

**Examples**

To compute the size of `my_table` without including out-of-line columns, use:

```
CALL ttComputeTabSizes ('my_table', 0);
```

**Notes**

The built-in procedure allows concurrent insertions while `ttComputeTabSizes` is executing. For this reason, the size computed by `ttComputeTabSizes` for each table is any value between the minimum size of the table during the computation and the maximum size of the table during the computation. For example, if the size of a table is 250 MB when `ttComputeTabSizes` is executed, and a transaction running concurrently raises the size of the table to 300 MB, `ttComputeTabSizes` estimates a value between 250 and 300 MB.

**See also**

`ttSize`
**Description**

The `ttConfiguration` built-in procedure returns the values for most, but not all, connection attributes for the current database connection. Specifically, the `ttConfiguration` built-in procedure returns the values for these connection attributes:

- CacheAwtMethod
- CacheAwtParallelism
- CkptFrequency
- CkptLogVolume
- CkptRate
- CkptReadThreads
- CommitBufferSizeMax
- ConnectionCharacterSet
- ConnectionName
- Connections
- CreateEpochatCommit
- DDLReplicationAction
- DDLReplicationLevel
- DataBaseCharacterSet
- DataStore
- DynamicLoadEnable
- Durability
- DurableCommits
- DynamicLoadErrorMode
- EpochInterval
- ForceDisconnectEnabled
- IncludeInCore
- Isolation
- LockLevel
- LockWait
- LogAutoTruncate
- LogBufMB
- LogBufParallelism
- LogDir
- LogFileSize
- LogFlushMethod
- LogPurge
- MemoryLock
- NLS_LENGTH_SEMANTICS
- NLS_NCHAR_CONV_EXCP
- NLS_SORT
- OptimizerHint
- OracleNetServiceName
- PLSCOPE_SETTINGS
- PLSQL_CCFLAGS
- PLSQL_CODE_TYPE
- PLSQL_CONN_MEM_LIMIT
- PLSQL_MEMORY_ADDRESS
- PLSQL_MEMORY_SIZE
- PLSQL_OPTIMIZE_LEVEL
- PLSQL_SESSION_CACHE_ENABLED
- PLSQL_TIMEOUT
ttConfiguration

Benefits
- Built-In Procedures
- 3-95
- PassThrough
- PermSize
- PermWarnThreshold
- Preallocate
- PrivateCommands
- QueryThreshold
- RACCallback
- RecoveryThreads
- ReplicationApplyOrdering
- ReplicationParallelism
- ReplicationTrack
- SQLQueryTimeout
- TempSize
- TempWarnThreshold
- Temporary
- UID

Required privilege
This procedure requires no privilege.

Usage in TimesTen Scaleout
TimesTen Scaleout applications can call this built-in procedure.
This procedure executes locally on the element from which it is called.

Related views
This procedure has these related views.
SYS.GV$CONFIGURATION
SYS.V$CONFIGURATION

Syntax
ttConfiguration(['paramName'])

Parameters
ttConfiguration has the optional parameter:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>paramName</td>
<td>TT_VARCHAR (30)</td>
<td>The name of a connection attribute for which you want this procedure to return the value.</td>
</tr>
</tbody>
</table>

Result set

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>paramName</td>
<td>TT_VARCHAR (30)</td>
<td>The names of the connection attributes specified in the connection string, returned in alphabetical order.</td>
</tr>
<tr>
<td>paramValue</td>
<td>TT_VARCHAR (1024)</td>
<td>The values of the connection attributes specified in the connection string.</td>
</tr>
</tbody>
</table>
Examples

To see the value of the QueryThreshold connection attribute, use

CALL ttConfiguration('querythreshold');
<QueryThreshold, 0>
1 row found

To see the values of all attributes, use:

CALL ttConfiguration();
< CkptFrequency, 600 >
< CkptLogVolume, 0 >
...

Notes

The values of client driver attributes are not returned by this procedure.
The values of other attributes, such as ForceConnect, may not be returned by this procedure, as well.

See also

Chapter 2, "Connection Attributes"
ttContext

Description
This procedure returns the context value of the current connection as a BINARY(8) value. You can use the context to correlate a unique connection to a database from the list of connections presented by the ttStatus utility and the ttDataStoreStatus built-in procedure.

Required privilege
This procedure requires no privilege.

Usage in TimesTen Scaleout
TimesTen Scaleout applications can call this built-in procedure.
This procedure executes locally on the element from which it is called.

Related views
This procedure has these related views.
SYS.GV$CONTEXT
SYS.V$CONTEXT

Syntax

```sql
ttContext()
```

Parameters

`ttContext` has no parameters.

Result set

`ttContext` returns the result set:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>context</td>
<td>BINARY(8)</td>
<td>Current connection context value.</td>
</tr>
</tbody>
</table>

Examples

```
CALL ttContext;
```

Notes

The context value numbers are unique only within a process. The context value number is not unique within the entire database. Therefore you may see the same context value number for different processes.

See also

"`ttStatus`" on page 5-161
ttDataStoreStatus

Description
This procedure returns the list of processes connected to a database. If the `dataStore` parameter is specified as `NULL`, then the status of all active databases is returned.
The result set is similar to the printed output of the `ttStatus` utility.

Required privilege
This procedure requires no privilege.

Usage in TimesTen Scaleout
TimesTen Scaleout applications can call this built-in procedure.
This procedure executes locally on the element from which it is called.

Related views
This procedure has these related views.

- `SYS.GV$DATASTORE_STATUS`
- `SYS.V$DATASTORE_STATUS`

Syntax
`ttDataStoreStatus('dataStore')`

Parameters
`ttDataStoreStatus` has the parameter:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>dataStore</code></td>
<td><code>TT_VARCHAR (256)</code></td>
<td>Full path name of desired database or <code>NULL</code> for all databases.</td>
</tr>
</tbody>
</table>

Result set
`ttDataStoreStatus` returns the result set:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>dataStore</code></td>
<td><code>TT_VARCHAR (256)</code></td>
<td>Full path name of database.</td>
</tr>
<tr>
<td><code>PID</code></td>
<td><code>TT_INTEGER NOT NULL</code></td>
<td>Process ID.</td>
</tr>
<tr>
<td><code>Context</code></td>
<td><code>BINARY(8) NOT NULL</code></td>
<td>Context value of connection.</td>
</tr>
</tbody>
</table>
### Built-In Procedures

#### ttDataStoreStatus

**Examples**

```sql
CALL ttDataStoreStatus('/data/Purchasing');
```

**See also**

- `ttContext`
- "ttStatus" on page 5-161
ttDBCompactConfig

Description

The ttDBCompactConfig built-in procedure turns on automatic database compaction. By default, TimesTen does not compact databases automatically.

Required privilege

This procedure requires ADMIN privilege.

Usage in TimesTen Scaleout

TimesTen Scaleout applications can call this built-in procedure. This procedure executes locally on the element from which it is called.

Related views

This procedure has these related views.

SYS.GV$DB_COMPACT_CONFIG
SYS.V$DB_COMPACT_CONFIG

Syntax

```
ttDBCompactConfig([[[value],[value]],[[value]]])
```

Parameters

ttDBCompactConfig has the parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>quantum</td>
<td>TT_INTEGER</td>
<td>Specifies the number of data blocks to be compacted. Values from 0 to 100000 are allowed. A value of 0 mean that automatic database compaction is disabled.</td>
</tr>
<tr>
<td>compactsPerSecond</td>
<td>TT_INTEGER</td>
<td>Number of compaction operations that can occur per second. Values from 0 to 100 are allowed. A value of 0 means that automatic database compaction is disabled.</td>
</tr>
<tr>
<td>threshold</td>
<td>TT_INTEGER</td>
<td>Specifies the minimum amount of the database that needs to be compacted, before automatic database compaction occurs. The units are the same as for parameter quantum. Values from 0 to 100000 (100k) are allowed. A value of 0 means that the compaction operations executes whenever there is anything to compact in the database.</td>
</tr>
</tbody>
</table>

Result set

ttDBCompactConfig returns the result set:
Examples

To view the current settings for automatic database compaction, use:

Call ttDbCompactConfig;
< 0,0,0 >
1 row found.

Note: These are the default settings. Automatic database compaction is disabled if either of the first two parameters is 0.

To enable automatic database compaction on 1,000 blocks once a second, use:

Call ttDbCompactConfig (1000,1,0);
< 1000,1,0 >
1 row found.

To enable automatic database compaction on 5,000 blocks ten times a second, use:

Call ttDbCompactConfig (5000,10,0);
< 5000,10,0 >
1 row found.

To enable automatic database compaction on 2,000 blocks five times a second, but only perform compaction when there is at least this much to compact, use:

Call ttDbCompactConfig (2000,5,2000);
< 2000,5,2000 >
1 row found.

To stop automatic database compaction from doing further compaction, after it was turned on (note that it is off by default), use:

Call ttDbCompactConfig (0,0,0);

You can set just one or two values at a time, using commas as placeholders:

call ttDbCompactConfig(2000,5,2000);
< 2000, 5, 2000 >
1 row found.
call ttDbCompactConfig(3000);
< 3000, 5, 2000 >
1 row found.
call ttDbCompactConfig(,10);
< 3000, 10, 2000 >
1 row found.
call ttDbCompactConfig(,2500);
< 3000, 10, 2500 >
1 row found.
call ttDbCompactConfig(3500,,3000);
< 3500, 10, 3000 >
1 row found.
Notes

After using this built-in procedure to set a parameter value, initiate a checkpoint to ensure the persistence of the parameter change. See details about the \texttt{ttCkpt} procedure in "Checkpoint operations" in the \textit{Oracle TimesTen In-Memory Database Operations Guide}. For details about the checkpoint built-in procedure, see "\texttt{ttCkpt}" in this chapter.

You can specify one, two, or three input values, using commas as placeholders, or no input to see the current values.

Changes to parameter values made by \texttt{ttDBCompactConfig} cannot be rolled back.

See Also

\texttt{ttDBConfig}
ttDBConfig

Description

The ttDBConfig built-in enables users to set or view the value of a TimesTen database system parameter.

Required privilege

This procedure requires ADMIN privilege.

Usage in TimesTen Scaleout

TimesTen Scaleout applications can call this built-in procedure. This procedure executes locally on the element from which it is called.

Related views

This procedure has these related views.

SYS.GV$DB_CONFIG
SYS.V$DB_CONFIG

Syntax

ttDBConfig(["param", "," value])

Parameters

ttDBConfig has the parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>param</td>
<td>VARCHAR2(30)</td>
<td>A system parameter for which you either want to set a value or see the current value. Accepted values for this argument are: CacheAgentCommitBufSize, CacheAwtMethod, CacheParAwtBatchSize, ParReplMaxDrift, PLSQL_OPEN_CURSORS, RepAgentCommitBufSize</td>
</tr>
<tr>
<td>value</td>
<td>VARCHAR2(200)</td>
<td>The value of the system parameter. If you do not specify a value, this procedure returns the current value of the specified parameter.</td>
</tr>
</tbody>
</table>

Parameter / Value Pairs

These name/value pairs can be returned in the result set:
<table>
<thead>
<tr>
<th>Name</th>
<th>Value Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CacheAgentCommitBufSize</td>
<td>Size in MB</td>
<td>Specifies the reclaim buffer maximum size for the cache agent. The cache agent periodically checks to see if the value has changed. The size cannot be greater than the temporary partition size.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For more details, see &quot;Improving performance when reclaiming memory during autorefresh operations&quot; in the <em>Oracle TimesTen Application-Tier Database Cache User's Guide</em>.</td>
</tr>
<tr>
<td>CacheAwtMethod</td>
<td>0 - SQL Array execution method 1 - PL/SQL Execution method</td>
<td>Determines whether PL/SQL execution method or SQL array execution method is used for AWT propagation to apply changes to the Oracle database server.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>See the description of the CacheAwtMethod connection attribute for details.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If set with this built-in procedure, overrides the connection attribute value.</td>
</tr>
<tr>
<td>CacheParAwtBatchSize</td>
<td>Number of rows in a batch</td>
<td>Configures a threshold value for the number of rows included in a single batch. Once the maximum number of rows is reached, TimesTen includes the rest of the rows in the transaction (TimesTen does not break up any transactions), but does not add any more transactions to the batch.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>NOTE:</strong> You should not change the value of this parameter unless advised by Oracle TimesTen technical support.</td>
</tr>
<tr>
<td>ParReplMaxDrift</td>
<td>Number of seconds</td>
<td>Specifies the number of seconds of drift to allow between the parallel replication tracks. When you use automatic parallel replication with disabled commit dependencies, some of the tracks may move ahead of the others. Once this threshold is passed, TimesTen synchronizes all replication tracks so that they catch up to each other. By default, this is set to zero, which means that checking for drift between tracks is disabled.</td>
</tr>
<tr>
<td>PLSQL_OPEN_CURSORS</td>
<td>Maximum number of PL/SQL cursors</td>
<td>Specifies the maximum number of PL/SQL cursors that can be open in a session at one time, a value from 0 to 65535, inclusive. Use this to prevent a session from opening an excessive number of cursors.</td>
</tr>
<tr>
<td>RepAgentCommitBufSize</td>
<td>Size in MB</td>
<td>Specifies the reclaim buffer maximum size for the replication agent. The replication agent periodically checks to see if the value has changed. The size cannot be greater than the temporary partition size.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For more details, see &quot;Improving performance when reclaiming memory during autorefresh operations&quot; in the <em>Oracle TimesTen Application-Tier Database Cache User's Guide</em>.</td>
</tr>
</tbody>
</table>
**ttDBConfig** returns the result set:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>param</td>
<td>VARCHAR2(30)</td>
<td>The name of the specified parameter.</td>
</tr>
<tr>
<td>value</td>
<td>VARCHAR2(200)</td>
<td>The current value of the specified parameter. This is the new value, if you specified a value.</td>
</tr>
</tbody>
</table>

**Examples**

To retrieve the current value of the `CacheParAwtBatchSize`, use:

```sql
CALL ttDBConfig('CacheParAwtBatchSize');
<CACHEPARAWTBATCHSIZE, 125>
1 row found.
```

To set the value of the `RepAgentCommitBufSize` to 50 MB, use:

```sql
CALL ttDBConfig('RepAgentCommitBufSize', '50');
<REPAGENTCOMMITBUFSIZE, 50>
1 row found.
```

To set the current value of the `CacheAgentCommitBufSize` to 100, use:

```sql
Command> call ttDBConfig('CacheAgentCommitBufSize', '100');
< CACHEAGENTCOMMITBUFSIZE, 100 >
1 row found.
```

**Notes**

After using this built-in procedure to set a parameter value, initiate a checkpoint to ensure the persistence of the parameter change. See details about the `ttCkpt` procedure in "Checkpoint operations" in the *Oracle TimesTen In-Memory Database Operations Guide*. For details about the checkpoint built-in procedure, see "ttCkpt" in this chapter.

Changes to parameter values made by `ttDBConfig` cannot be rolled back.

If you call `ttDBConfig` without an input parameter, it will return names and values of all supported parameters.

**See also**

ttDBWriteConcurrencyModeGet

Description

The ttDBWriteConcurrencyModeGet built-in returns information about the write concurrency mode of the database and the status of write concurrency mode operations and transactions.

Required privilege

This procedure requires no privilege.

Usage in TimesTen Scaleout

This procedure is not supported in TimesTen Scaleout.

Related views

This procedure has these related views.

SYS.GV$DB_WRTE_CONCURRENCY_MODE
SYS.V$DB_WRTE_CONCURRENCY_MODE

Syntax

ttDBWriteConcurrencyModeGet()

Parameters

ttDBWriteConcurrencyModeGet has no parameters:

Result set

ttDBWriteConcurrencyModeGet returns the result set:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ts</td>
<td>TIMESTAMP NOT NULL</td>
<td>Time at which the status information was collected.</td>
</tr>
<tr>
<td>mode</td>
<td>TT_INTEGER NOT NULL</td>
<td>The write concurrency mode:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 - Optimize according to hints and standard optimization techniques.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - Optimize for concurrent write operations.</td>
</tr>
<tr>
<td>operation</td>
<td>VARCHAR2 (50)</td>
<td>The transition status of the write concurrency mode. Either:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NULL - Not in transition.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TRANSITIONING TO MODE=n where n= 0 or 1.</td>
</tr>
<tr>
<td>status</td>
<td>VARCHAR2 (100) NOT NULL</td>
<td>The status of the write concurrency mode transition. Either:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IN TRANSITION or COMPLETE.</td>
</tr>
<tr>
<td>msg</td>
<td>VARCHAR2 (5000) NULL</td>
<td>NULL or a status explanation message.</td>
</tr>
</tbody>
</table>

Examples

The following example shows how to determine if your database is optimized for concurrent write operations:
Command> CALL ttDBWriteConcurrencyModeGet();

1 row found.

The results indicate that at approximately 1:48 pm on September 23, 2013 the database was optimized for concurrent write operations. The mode was not in transition.

See also

ttDBWriteConcurrencyModeSet
**ttDBWriteConcurrencyModeSet**

**Description**

The `ttDBWriteConcurrencyModeSet` built-in enables control over read optimization during periods of concurrent write operations.

Set the mode to one (1) to enable the enhanced write concurrency mode and disable read optimization. Set the mode to zero (0) to disable the enhanced write concurrency mode and re-enable read optimization.

When the mode is set to one (1), all transaction and statement table lock hints are suppressed. This affects hint-triggered Sn table locks for `SELECT` statements and subqueries and also hint-triggered W table locks for DML statements. Suppression of the table lock hint also suppresses other table-lock hint driven execution plans such as star joins. Regardless of the mode setting, table locks that are not triggered by table-lock hints are not affected.

**Required privilege**

This procedure requires `ADMIN` privilege.

**Usage in TimesTen Scaleout**

TimesTen Scaleout applications can call this built-in procedure.

This procedure executes on all elements in the grid.

**Related views**

This procedure has no related views.

**Syntax**

```
ttDBWriteConcurrencyModeSet(mode, wait)
```

**Parameters**

`ttDBWriteConcurrencyModeSet` has these parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| mode      | TT_INTEGER NOT NULL | The write concurrency mode:  
|           |                  | 0 - Optimize according to hints and standard optimization techniques.  
|           |                  | 1 - Optimize for concurrent write operations.  
| wait      | TT_INTEGER NOT NULL | 0 - Return immediately after starting mode transition.  
|           |                  | 1 - Wait until mode transition is complete before returning. This can be useful when setting the mode to a nonzero value. When setting the mode to zero, it is typically not necessary to specify `wait` to 1.  

**Result set**

`ttDBWriteConcurrencyModeSet` returns no result set:
Examples

The following example shows how to enable standard optimization techniques and return immediately after starting the operation:

Command> CALL ttDBWriteConcurrencyModeSet(0,0);

Notes

When the mode is set to one (1), all transaction and statement table lock hints are suppressed. This affects hint-triggered $s_n$ table locks for SELECT statements and subqueries and also hint-triggered $w$ table locks for DML statements. Suppression of the table lock hint also suppresses other table-lock hint driven execution plans such as star joins. Regardless of the mode setting, table locks that are not triggered by table-lock hints are not affected.

See also

ttDBWriteConcurrencyModeGet
ttDistributeProgress

Description
This built-in procedure provides a progress report of an ongoing redistribution process.

Required privilege
This procedure requires the ADMIN privilege.

Usage in TimesTen Scaleout
TimesTen Scaleout applications can call this built-in procedure.
This procedure returns a row for the element from which it was called.

Related views
This procedure has no related views.

Syntax
```
ttDistributeProgress()
```

Parameters
```

ttDistributeProgress has no parameters.
```

Result set
```
ttDistributeProgress returns the result:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>timestamp</td>
<td>TT_TIMESTAMP</td>
<td>Time at which the status information was collected.</td>
</tr>
<tr>
<td>elementId</td>
<td>TT_INTEGER</td>
<td>Element ID of the local element.</td>
</tr>
<tr>
<td>ptVersion</td>
<td>TT_INTEGER</td>
<td>Version number of the partition table.</td>
</tr>
<tr>
<td>threadId</td>
<td>TT_INTEGER</td>
<td>Thread ID.</td>
</tr>
<tr>
<td>phase</td>
<td>VARCHAR2 (32)</td>
<td>Current phase of the redistribution process. The redistribution process has the following phases: ■ Data Distribution ■ Data Checkpoint ■ Data Checkpoint Done ■ Reclaim Phase 1 ■ Reclaim Checkpoint ■ Reclaim Phase 2 ■ Reclaim Done</td>
</tr>
<tr>
<td>tblName</td>
<td>VARCHAR2 (64)</td>
<td>Name of the table currently being processed, if available.</td>
</tr>
<tr>
<td>processedTblRows</td>
<td>TT_BIGINI</td>
<td>Number of rows already processed of the current table, if available.</td>
</tr>
</tbody>
</table>
```
Examples

The following example shows an example result set for a call to the `ttDistributionProgress` built-in procedure.

Command> CALL ttDistributionProgress();
< 2018-12-04 14:49:41.065122, 1, 2, 1, Data Distribution, <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, 1910, 0, 176, 1910, 8, 8 >
1 row found.

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>insertedTblRows</td>
<td>TT_BIGINT</td>
<td>Number of rows already inserted to the current table in the local element.</td>
</tr>
<tr>
<td>deletedTblRows</td>
<td>TT_BIGINT</td>
<td>Number of rows already deleted from the current table in the local element.</td>
</tr>
<tr>
<td>totalTblRows</td>
<td>TT_BIGINT</td>
<td>Total number of rows in the table, if available.</td>
</tr>
<tr>
<td>processedRows</td>
<td>TT_BIGINT</td>
<td>Number of rows already processed for the element.</td>
</tr>
<tr>
<td>insertedRows</td>
<td>TT_BIGINT</td>
<td>Number of rows already inserted to the element.</td>
</tr>
<tr>
<td>deletedRows</td>
<td>TT_BIGINT</td>
<td>Number of rows already deleted from the element.</td>
</tr>
<tr>
<td>totalRows</td>
<td>TT_BIGINT</td>
<td>Total number of rows in the element for all tables.</td>
</tr>
<tr>
<td>processedTbls</td>
<td>TT_INTEGER</td>
<td>Number of tables already processed.</td>
</tr>
<tr>
<td>totalTbls</td>
<td>TT_INTEGER</td>
<td>Total number of tables in the database.</td>
</tr>
</tbody>
</table>
ttDurableCommit

Description

This procedure specifies that the current transaction should be made durable when it is committed. It only has an effect if the application is connected to the database with DurableCommits disabled.

Calling ttDurableCommit also makes durable the current transaction and any previously committed delayed durability transactions. There is no effect on other transactions that are committed after calling ttDurableCommit. ttDurableCommit does not commit transactions. The application must do the commit, for example with a call to SQLTransact.

Required privilege

This procedure requires no privilege.

Usage in TimesTen Scaleout

This procedure is not supported in TimesTen Scaleout.

Related views

This procedure has no related views.

Syntax

```
ttDurableCommit()
```

Parameters

ttDurableCommit has no parameters.

Result set

ttDurableCommit returns no results.

Examples

```
CALL ttDurableCommit;
```

Notes

Some controllers or drivers may only write data into cache memory in the controller or may write to disk some time after the operating system is told that the write is done. In these cases, a power failure may mean that some information you thought was durably committed does not survive the power failure. To avoid this loss of data, configure your disk to write all the way to the recording media before reporting completion or you can use an Uninterruptable Power Supply (UPS).
ttEpochCreate

Description
This procedure causes the next committed transaction to commit as an epoch transaction. An epoch is a transaction that marks a globally consistent point in time across all elements of the database. An epoch is durably committed in every replica set of a grid.

Required privilege
This procedure requires no privilege.

Usage in TimesTen Scaleout
TimesTen Scaleout applications can call this built-in procedure.
This procedure executes on all elements in the grid.

Restrictions with TimesTen Classic
TimesTen Classic applications cannot call this built-in procedure.

Related views
This procedure has no related views.

Syntax
```sql
ttEpochCreate()
```

Parameters
```sql
ttEpochCreate has no parameters.
```

Result set
```sql
ttEpochCreate returns no results.
```

Examples
```sql
CALL ttEpochCreate;
```
ttEpochSessionGet

Description
This procedure returns the epoch identifier of the last epoch created by the current
connection, if one is available.

Required privilege
This procedure requires no privilege.

Usage in TimesTen Scaleout
TimesTen Scaleout applications can call this built-in procedure.
This procedure executes on all elements in the grid.

Restrictions with TimesTen Classic
TimesTen Classic applications cannot call this built-in procedure.

Related views
This procedure has these related views.
SYS.GV$EPOCH_SESSION
SYS.V$EPOCH_SESSION

Syntax
```
ttEpochSessionGet()
```

Parameters
```
ttEpochSessionGet has no parameters.
```

Result set
```
ttEpochSessionGet returns the result set:
```

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>epoch</td>
<td>TT_VARCHAR (50)</td>
<td>The epoch session ID, if available.</td>
</tr>
</tbody>
</table>

Examples
```
CALL ttEpochSessionGet();
< 1023 >
```

Notes
**ttHeapInfo**

**Description**
This procedure reports heap memory usage in the database. For each heap in the database, it displays the allocated size, size in use, high water mark (the maximum amount of size in use) and the number of deferred free buffers.

**Required privilege**
This procedure requires no privilege.

**Usage in TimesTen Scaleout**
TimesTen Scaleout applications can call this built-in procedure.
This procedure executes locally on the element from which it is called.

**Related views**
This procedure has these related views.
SYS.GV$HEAP_INFO
SYS.V$HEAP_INFO

**Syntax**
ttHeapInfo('name')

**Result set**
ttHeapInfo has the parameter:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>TT_CHAR (30)</td>
<td>Name of the database for which you would like heap info.</td>
</tr>
</tbody>
</table>

**Result set**
ttHeapInfo returns the result:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>TT_CHAR (30) NOT NULL</td>
<td>Name of the database for which the heap memory info is being returned.</td>
</tr>
<tr>
<td>size</td>
<td>TT_BIGINT NOT NULL</td>
<td>The allocated sizes of the heap memory.</td>
</tr>
<tr>
<td>inUse</td>
<td>TT_BIGINT NOT NULL</td>
<td>The amount of heap memory in use.</td>
</tr>
<tr>
<td>highWater</td>
<td>TT_BIGINT NOT NULL</td>
<td>The maximum amount of heap memory used.</td>
</tr>
<tr>
<td>freeDeferred</td>
<td>TT_BIGINT NOT NULL</td>
<td>The number of deferred freed heap memory buffers.</td>
</tr>
</tbody>
</table>

**Examples**
CALL ttHeapInfo ('sampledb1');
< PERMANENT_0, 2515656, 2404112, 2582856, 0>
< PERMANENT_1, 1024, 1024, 1024, 0>
...
< INDEX_SNAPSHOT_VALUE_CONFIG_I, 2048, 776, 776, 0>
156 rows found
ttHostNameGet

Description
This procedure returns the name of the current local host for the database. The value returned is only for the current session. It is not a systemwide setting and does not persist after the current session has been disconnected.

Use this procedure to check whether a particular store name in a scheme refers to the current host. This can be helpful when configuring replication schemes.

Required privilege
This procedure requires no privilege.

Usage in TimesTen Scaleout
TimesTen Scaleout applications can call this built-in procedure.

This procedure returns a row for the element from which it was called.

Related views
This procedure has these related views.
SYS.GV$HOST_NAME
SYS.V$HOST_NAME

Syntax
ttHostnameGet()

Parameters
ttHostNameGet has no parameters.

Result set
ttHostNameGet returns the result:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hostName</td>
<td>TT_VARCHAR (200)</td>
<td>The current default local host setting for the database. If a default has not been supplied then the current host name is returned.</td>
</tr>
</tbody>
</table>

Examples
CALL ttHostNameGet ();
< myhost >
1 row found.

See also
ttHostNameSet
ttHostNameSet

Description
This procedure specifies the name of the default local host for the current database. The value is only used in the current session, it is not a systemwide setting and does not persist after the current session has been disconnected.

To configure master/subscriber relationships and replication object permissions correctly, Replication DDL processing relies on being able to determine whether a host name used in a replication scheme refers to the computer on which the script is currently being run. This procedure enables an application to set a default host name for the current session that Replication DDL processing uses whenever there is a need to establish the name of the current host.

Required privilege
This procedure requires the ADMIN privilege.

Usage in TimesTen Scaleout
TimesTen Scaleout applications can call this built-in procedure.
This procedure executes locally on the element from which it is called.

Related views
This procedure has no related views.

Syntax

```
ttHostNameSet('hostName')
```

Parameters
ttHostNameSet has the parameter:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hostName</td>
<td>TT_VARCHAR (200)</td>
<td>The required default name for the local computer.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>To clear the default value, specify NULL.</td>
</tr>
</tbody>
</table>

Result set

`ttHostNameSet` returns no results.

Examples

```
CALL ttHostNameSet ('alias1');
```

Notes

The legal value of `hostName` can be any host name or IP address string except 'localhost', '127.0.0.1' or '::1'. You cannot set the default host name to a value that is different from a local host name used in an existing replication scheme.
See also

ttHostNameGet
ttIndexAdviceCaptureDrop

Description
This procedure drops existing capture data for either the current connection or for the
database. Subsequent calls to ttIndexAdviceCaptureOutput at that level return no
rows.

This procedure and the procedures related to it are referred to as the Index Advisor.
For details on using these procedures, see "Using the Index Advisor to recommend
indexes" in the Oracle TimesTen In-Memory Database Operations Guide.

Required privilege
This procedure requires no privileges to drop a connection level capture.
This procedure requires ADMIN privileges to drop a database level capture.

Usage in TimesTen Scaleout
TimesTen Scaleout applications can call this built-in procedure.
This procedure executes locally on the element from which it is called.

Related views
This procedure has no related views.

Syntax

ttIndexAdviceCaptureDrop([captureLevel])

Parameters

ttIndexAdviceCaptureDrop has this optional parameter:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>captureLevel</td>
<td>TT_INTEGER</td>
<td>Legal values for the capture level are: 0 - Index advice capture is dropped at the connection level for the current connection. This is the default. 1 - Index advice capture is dropped at the database level.</td>
</tr>
</tbody>
</table>

Result set

ttIndexAdviceCaptureDrop returns no results.

Examples

CALL ttIndexAdviceCaptureDrop;

Notes

To drop both connection level and database level captures, invoke the command twice,
one for each capture level.

It is an error to call this command while a capture is in progress at the level you are
attempting to drop.
See also

"Using the Index Advisor to recommend indexes" in the Oracle TimesTen In-Memory Database Operations Guide
ttIndexAdviceCaptureEnd

Description

This procedure ends either an active connection level capture from the current connection or an active database level capture.

This procedure and the procedures related to it are referred to as the Index Advisor. For details on using these procedures, see "Using the Index Advisor to recommend indexes" in the Oracle TimesTen In-Memory Database Operations Guide.

Required privilege

This procedure requires no privilege to end a connection level capture.

This procedure requires ADMIN privileges to end a database level capture.

Usage in TimesTen Scaleout

TimesTen Scaleout applications can call this built-in procedure.

This procedure executes locally on the element from which it is called.

Related views

This procedure has no related views.

Syntax

ttIndexAdviceCaptureEnd([captureLevel])

Parameters

ttIndexAdviceCaptureEnd has this optional parameter:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>captureLevel</td>
<td>TT_INTEGER</td>
<td>Legal values for the capture level are:</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>- Ends index advice capture at the connection level for the current connection. This is the default.</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>- Ends index advice capture at the database level.</td>
</tr>
</tbody>
</table>

Result set

ttIndexAdviceCaptureEnd returns no results.

Examples

The following example ends the collection for the connection level capture:

Call ttIndexAdviceCaptureEnd(0)

Notes

To end both connection level and database level captures, invoke the command twice, once for each capture level.

It is an error to call this procedure without first starting a capture at the specified level by calling the ttIndexAdviceCaptureStart procedure.
See also

- `ttIndexAdviceCaptureDrop`
- `ttIndexAdviceCaptureInfoGet`
- `ttIndexAdviceCaptureOutput`
- `ttIndexAdviceCaptureStart`

"Using the Index Advisor to recommend indexes" in the *Oracle TimesTen In-Memory Database Operations Guide*
**ttIndexAdviceCaptureInfoGet**

**Description**

This procedure returns a row for each active capture. A capture is active if it has started capturing index advice or if it has stopped capturing index advice, but the capture data is still available.

One row relates to a connection level capture, if one exists. Another row relates to a database level capture, if one exists. At most there is one connection level and one database capture.

If no capture is in progress or no data exists, this procedure does not return any rows.

This procedure and the procedures related to it are referred to as the Index Advisor. For details on using these procedures, see "Using the Index Advisor to recommend indexes" in the *Oracle TimesTen In-Memory Database Operations Guide*.

**Required privilege**

This procedure requires no privilege to get information on a connection level capture.

This procedure requires `ADMIN` privileges to get information on a database level capture.

**Usage in TimesTen Scaleout**

TimesTen Scaleout applications can call this built-in procedure.

This procedure executes locally on the element from which it is called.

**Related views**

This procedure has no related views.

**Syntax**

```plaintext
ttIndexAdviceCaptureInfoGet()
```

**Parameters**

*ttIndexAdviceCaptureInfoGet* has no parameters.

**Result set**

*ttIndexAdviceCaptureInfoGet* returns the result set:

<table>
<thead>
<tr>
<th>Columns</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| captureState | TT_INTEGER NOT NULL | The state of the capture:  
|           |                          | 0 - A capture is not in progress.   
|           |                          | 1 - A capture is in progress.                                                                 |
| connID   | TT_INTEGER              | The connection ID of the connection that initiated the last capture, or the current capture if one is in progress.  
|          |                          | This row is not returned if no capture has been initiated.                     |
### Examples

This example shows capture information for a completed connection level capture for 363 prepared statements and 369 executed statements:

```
Command> CALL ttIndexAdviceCaptureInfoGet();
< 0, 1, 0, 0, 363, 369, 2018-02-27 11:44:08.136833,
  2018-02-27 12:07:35.410993 >
1 row found.
```

### Notes

If there is an active database level capture and you call this procedure on a connection that does not have ADMIN privilege, TimesTen returns an error.

### See also

- `ttIndexAdviceCaptureDrop`
- `ttIndexAdviceCaptureEnd`
- `ttIndexAdviceCaptureOutput`
- `ttIndexAdviceCaptureStart`
- "Using the Index Advisor to recommend indexes" in the *Oracle TimesTen In-Memory Database Operations Guide*
ttIndexAdviceCaptureOutput

Description
This built-in returns a list of index recommendations from the last recorded capture at the specified level. It also returns an executable `CREATE INDEX` SQL statement for creating the recommended index.

This procedure and the procedures related to it are referred to as the Index Advisor. For details on using these procedures, see "Using the Index Advisor to recommend indexes" in the Oracle TimesTen In-Memory Database Operations Guide.

For a connection level capture, run this procedure in the same connection that initiated the capture. For a database level capture, run this procedure in a connection with `ADMIN` privileges.

Required privilege
This procedure requires no privilege to get output on a connection level capture.
This procedure requires `ADMIN` privileges to get output on a database level capture.

Usage in TimesTen Scaleout
TimesTen Scaleout applications can call this built-in procedure.
This procedure executes locally on the element from which it is called.

Related views
This procedure has these related views.

SYS.GV$INDEX_ADVICE_OUTPUT
SYS.V$INDEX_ADVICE_OUTPUT

Syntax
```
ttIndexAdviceCaptureOutput([captureLevel])
```

Parameters
`ttIndexAdviceCaptureOutput` has this optional parameter:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>captureLevel</td>
<td>TT_INTEGER</td>
<td>Legal values for the capture level are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 - Outputs index advice at the connection level for the current connection.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is the default value.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - Outputs index advice at the database level.</td>
</tr>
</tbody>
</table>

Result set
`ttIndexAdviceCaptureOutput` returns the result set:
The following example provides the CREATE INDEX statement for an index called PURCHASE_i1 on the HR.PURCHASE table. There are four distinct statements that would benefit from the index in this SQL workload.

CALL ttIndexAdviceCaptureOutput();
< 4, create index PURCHASE_i1 on HR.PURCHASE(AMOUNT); >
1 row found.

Notes

All names returned are fully schema qualified.

See also

ttIndexAdviceCaptureDrop
ttIndexAdviceCaptureEnd
ttIndexAdviceCaptureInfoGet
ttIndexAdviceCaptureStart
"Using the Index Advisor to recommend indexes" in the Oracle TimesTen In-Memory Database Operations Guide
ttIndexAdviceCaptureStart

Description

This procedure enables index advice capture. It is recommended that statistics be updated before you call this procedure, using `ttOptEstimateStats` and setting the 'invalidate' parameter set to 'yes'. Updating the statistics in this way ensures statistics are up to date and forces statements to be re-prepared during the capture. To set statistics to known values instead, call `ttOptSetTblStats` with the 'invalidate' parameter set to 'yes'.

This procedure and the procedures related to it are referred to as the Index Advisor. For details on using these procedures, see "Using the Index Advisor to recommend indexes" in the Oracle TimesTen In-Memory Database Operations Guide.

Required privilege

This procedure requires no privilege to start a connection level capture.

This procedure requires `ADMIN` privileges to start a database level capture.

Usage in TimesTen Scaleout

TimesTen Scaleout applications can call this built-in procedure.

This procedure executes locally on the element from which it is called.

Related views

This procedure has no related views.

Syntax

```
ttIndexAdviceCaptureStart([captureLevel], [captureMode])
```

Parameters

`ttIndexAdviceCaptureStart` has these optional parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>captureLevel</code></td>
<td><code>TT_INTEGER</code></td>
<td>Legal values for the capture level are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 - Outputs index advice at the connection level for the current connection.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is the default value.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - Outputs index advice at the database level.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Legal values for the capture mode are:</td>
</tr>
<tr>
<td><code>captureMode</code></td>
<td><code>TT_INTEGER</code></td>
<td>0 - Provides complete capture of index advice including execution of the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SQL statements. This is the default.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>31 - Capture is based on the computed statistics and plan analysis. Queries</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(SELECT statements only) are prepared but not executed. This mode can only</td>
</tr>
<tr>
<td></td>
<td></td>
<td>be used with connection level captures (<code>captureLevel=0</code>).</td>
</tr>
</tbody>
</table>
**Result set**

```
ttIndexAdviceCaptureStart returns no results
```

**Examples**

The following example starts a collection for the Index Advisor at the connection-level.

Call `ttIndexAdviceCaptureStart(0,0);`

**Notes**

It is an error to call this procedure if index advice is already being captured at the level specified by the `captureLevel` parameter or at the connection level if no level is specified. Connection level captures can be issued concurrently on independent connections without conflict. Outstanding connection level captures that are in progress when a database level capture begins complete as intended.

**See also**

- `ttIndexAdviceCaptureDrop`
- `ttIndexAdviceCaptureEnd`
- `ttIndexAdviceCaptureInfoGet`
- `ttIndexAdviceCaptureOutput`
- "Using the Index Advisor to recommend indexes" in the *Oracle TimesTen In-Memory Database Operations Guide*
**ttLatchStatsGet**

**Description**

This procedure displays latch statistics. Statistics are useful for determining areas of contention in a running system.

This procedure is primarily meant to be used when requested by TimesTen technical support.

**Required privilege**

This procedure requires `ADMIN` privileges to show all active connections or database level statistics. No privileges are required to show the current connection's latch statistics.

**Usage in TimesTen Scaleout**

TimesTen Scaleout applications can call this built-in procedure

This procedure executes locally on the element from which it is called.

**Related views**

This procedure has no related views.

**Syntax**

```
ttLatchStatsGet (level, operation)
```

**Parameters**

`ttLatchStatsGet` has these parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>level</td>
<td>TT_CHAR (16)</td>
<td>The level controls the number of connections for which the stats are printed. Valid values are: db - All the active connections on the database. conn - The current connection. This is the default. connid - An specific connection (specified by connid).</td>
</tr>
<tr>
<td>operation</td>
<td>TT_CHAR (16)</td>
<td>This value controls the verbosity level of the output. Valid values are: show - Only show the contention points that have a high contention level showall - Show the contention points that have contention showallandtell - Show all the contention points</td>
</tr>
</tbody>
</table>

**Result set**

Results sets are are divided into two types: contention point and statistics.

`ttLatchStatsGet` returns the result set for contention points. These describe the location of contention.
ttLatchStatsGet returns the result set for statistics. These describe detailed statistics about this contention point:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>callerName</td>
<td>TT_VARCHAR(40) NOT NULL</td>
<td>Function name containing the contention point.</td>
</tr>
<tr>
<td>fileName</td>
<td>TT_VARCHAR(30) NOT NULL</td>
<td>The file that contains the callerName function.</td>
</tr>
<tr>
<td>lineNo</td>
<td>TT_INTEGER NOT NULL</td>
<td>The line number of the fileName file.</td>
</tr>
<tr>
<td>description</td>
<td>TT_VARCHAR(100) NOT NULL</td>
<td>Description of this contention point.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connName</td>
<td>TT_VARCHAR(40) NOT NULL</td>
<td>The name of the connection experiencing contention.</td>
</tr>
<tr>
<td>spinCount</td>
<td>TT_BIGINT NOT NULL</td>
<td>The number of times the connName connection has spun on this contention point.</td>
</tr>
<tr>
<td>access</td>
<td>TT_BIGINT NOT NULL</td>
<td>The number of times the connName connection has used this contention point.</td>
</tr>
<tr>
<td>sleepCnt</td>
<td>TT_INTEGER NOT NULL</td>
<td>The number of times the connName connection has slept on this contention point.</td>
</tr>
<tr>
<td>firstTry</td>
<td>TT_INTEGER NOT NULL</td>
<td>The number of times the connName connection has used this contention point without experiencing contention.</td>
</tr>
<tr>
<td>collisions</td>
<td>TT_INTEGER NOT NULL</td>
<td>The number of times the connName connection has used this contention point and experienced contention.</td>
</tr>
<tr>
<td>avgSpin</td>
<td>TT_BIGINT NOT NULL</td>
<td>The average number of times the connName connection has spun on this contention point.</td>
</tr>
</tbody>
</table>

Examples

The following example shows an example result set for a call to ttLatchStatsGet.

Command> CALL ttLatchStatsGet
< getSmallMed, heap.c 2675, Generic description, sampledb1,
  1, 0, 1, 0, -1, -1, 0 >
< sbhpallocAttempt, heap.c 3712, Generic description, sampledb1, 1, 0, 1, 0, -1, -1, 0>

See also

"ttXactAdmin" on page 5-173
ttLoadFromOracle

Description

This procedure takes a TimesTen table name, an Oracle SELECT statement and the
number of threads for parallel load. It executes the query on the Oracle database and
loads the result set into the specified TimesTen table. While performing the load, an
implicit commit is executed after every 256 rows inserted into the TimesTen database.

No character set conversion is performed when loading data from an Oracle database
into a TimesTen table. The TimesTen database and the Oracle database must use the
same character set.

The procedure requires the connection attribute UID, the connection attribute
OraclePWD and the connection attribute OracleNetServiceName to be specified. You
must commit after calling this procedure.

For more details and usage information, see "Loading data from an Oracle database
into a TimesTen table" in the Oracle TimesTen In-Memory Database Operations Guide.

Required privilege

This procedure requires INSERT privileges to the table to be loaded.

The session must have all the required privileges to execute the query on the Oracle
database.

Usage in TimesTen Scaleout

TimesTen Scaleout applications can call this built-in procedure.

This procedure executes locally on the element from which it is called.

Related views

This procedure has no related views.

Syntax

```
ttLoadFromOracle(['tblOwner'], 'tblName', 'Query' [,numThreads], 'Options')
```

Parameters

ttLoadFromOracle has these parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tblOwner</td>
<td>TT_CHAR (30)</td>
<td>TimesTen table owner (optional). If not provided, the connection ID is used.</td>
</tr>
<tr>
<td>tblName</td>
<td>TT_CHAR (30) NOT NULL</td>
<td>Name of the table to be loaded with data from the Oracle database. You can use the built-in procedure ttTableSchemaFromOraQueryGet to get a schema with which to build the table, if one does not already exist. The specified TimesTen table cannot be a system table, a synonym, a view, a materialized view or a detail table of a materialized view, a global temporary table or a cache group table.</td>
</tr>
</tbody>
</table>
The options are specified as key = value pairs and the pairs are separated by semi-colons.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
</table>
| localOnly       | Y or N                 | This option only loads rows from a specific instance. Load a specific instance in the grid and use this option. When you use this option, ttLoadFromOracle selects all rows from the table, but ignores any rows that are not hashed to the specific instance. This option is only supported in TimesTen Scaleout.
|                 |                        | The default value is N. |
| ErrorThreshold  | value > 0              | This option sets the error threshold for which the built-in procedure returns an error messages. The default value is 1. When the ttLoadFromOracle built-in procedure encounters an error, the built-in procedure stops and returns an error message. |
ttLoadFromOracle returns the result set:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numRows</td>
<td>TT_BIGINT</td>
<td>Number of rows loaded.</td>
</tr>
<tr>
<td>numErrors</td>
<td>TT_BIGINT</td>
<td>Number of rows with errors.</td>
</tr>
<tr>
<td>errCode</td>
<td>TT_INTEGER</td>
<td>TimesTen error code, one of:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0  - Load completed successfully without errors</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-1 - Load completed successfully with errors</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-2 - Load terminated early with errors</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-3 - Load terminated early with a fatal error, for example, an out-of-space error, a loss of connection or an invalidation.</td>
</tr>
<tr>
<td>errMsg</td>
<td>TT_VARCHAR(4000)</td>
<td>Error message, containing:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Start and end time of load</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Statement</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ SCN used to query the data</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Number of rows with errors</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Number of rows loaded</td>
</tr>
</tbody>
</table>
Examples

The following example selects loads the TimesTen table about employees from the Oracle database HR.EMPLOYEES table and loads it into the TimesTen HR.EMPLOYEES table. In this example an error is returned. In this example, the column STATE is a TT_TINYINT.

Command> CALL ttLoadFromOracle ('HR', 'EMPLOYEES',
'SELECT * FROM HR.EMPLOYEES');
Statement=ttLoadFromOracle('HR', 'SELECT * FROM HR.EMPLOYEES'); SCN=1234567;
Errors=1; Rows Loaded=99' >
< NULL, NULL, 2614,'Value outside of range supported by integral type. Column STATE=-1' >

Notes

TimesTen does not empty the table before the load.
The target table does not require a primary key.
TimesTen returns an error if the query output cannot be converted to rows in the target table due to a mismatch of column types or number of columns.
Loading data into TimesTen LOB columns is not supported. If the query on the Oracle database has LOB output, it is mapped to a VAR type.
The load process does not check that the column data types and sizes in the TimesTen table match the data types and sizes of the result set. Instead, the insert is attempted and if the column data types cannot be mapped or the Oracle Database data from the SQL query exceeds the TimesTen column size, TimesTen returns an error.
LOB columns are truncated to 4 MB.
When a table is altered to add columns, secondary partitions are added. Loading a table with multiple partitions is not supported by ttLoadFromOracle.

See also

ttTableSchemaFromOraQueryGet
ttLockLevel

Description
Changes the lock level between row-level and database-level locking on the next transaction and for all subsequent transactions for this connection. Applications can change the lock level again by calling ttLockLevel again. The initial value depends on the LockLevel connection attribute. See "LockLevel" on page 2-63 for full details of the different locking levels.

Required privilege
This procedure requires the ADMIN privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax
```
ttLockLevel('lockLevel')
```

Parameters
ttLockLevel has the parameter:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lockLevel</td>
<td>TT_CHAR (20) NOT NULL</td>
<td>Locking level for the connection.</td>
</tr>
</tbody>
</table>

The value of `lockLevel` may be one of two case-insensitive strings:
- **Row**: Locking should be set to row-level locking.
- **DS**: Locking should be set to database-level locking.

Result set
ttLockLevel returns no results.

Examples
```
CALL ttLockLevel ('Row');
```

Notes
This procedure does not affect the current transaction.
Row-level locking is required when caching tables from an Oracle database.
This procedure must be called from within a transaction. It has the effect of setting the locking level for subsequent transactions for the connection that invoked it. The new lock level does not affect the current transaction. It takes effect at the beginning of the next transaction.
See also

\texttt{ttLockWait}
**ttLockWait**

**Description**

This procedure enables an application to change the lock timeout interval of the current connection. The change takes effect immediately and applies to all subsequent statements in the current transaction and all subsequent transactions on the connection.

The lock wait interval is the number of seconds to wait for a lock when there is contention on it. You can also indicate a fraction of a second.

Lock wait intervals are imprecise, and may be exceeded, generally by no more than 100 milliseconds, due to the scheduling of the agent that detects timeouts. This imprecision does not apply to zero second timeouts, which are always reported immediately.

**Required privilege**

This procedure requires no privilege.

**Usage in TimesTen Scaleout**

TimesTen Scaleout applications can call this built-in procedure. This procedure executes on all elements in the grid.

**Related views**

This procedure has no related views.

**Syntax**

```
ttLockWait (seconds)
```

**Parameters**

ttLockWait has the required parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>seconds</td>
<td>NUMBER (8,1) NOT NULL</td>
<td>Number of seconds to wait for a lock when there is contention on it. You can also specify fractions of a second. Valid values are 0.0 to 1000000.0 inclusive.</td>
</tr>
</tbody>
</table>

**Result set**

ttLockWait returns no results.

**Examples**

To indicate a six second lock wait, use:

```
CALL ttLockWait (6);
```

To indicate a tenth of a second lock wait, use:

```
CALL ttLockWait (0.1);
```
Notes

When a lock is not immediately available to a TimesTen transaction, it waits a predetermined amount of time to try to get the lock. After that it times out the lock request and returns error TT6003 to the application. By default, TimesTen uses a value of 10 seconds for lock timeouts. If a value of 0 is specified, transactions do not wait for any unavailable locks.

See also

ttLockLevel
"LockLevel" on page 2-63
ttLogHolds

Description

This procedure returns information about transaction log holds, including those created on behalf of incremental backups, replication peers, active standby pairs (and any subscribers), AWT cache groups, persistent XLA subscribers, XA, long-running transactions and checkpoints. This procedure can help diagnose situations where it appears that checkpoint operations are not purging all unneeded transaction log files.

Applications should monitor log holds and the accumulation of log files. For more information, see “Show replicated log records” in the Oracle TimesTen In-Memory Database Replication Guide and “Monitoring accumulation of transaction log files” in the Oracle TimesTen In-Memory Database Operations Guide.

Required privilege

This procedure requires no privilege.

Usage in TimesTen Scaleout

TimesTen Scaleout applications can call this built-in procedure.

This procedure returns a row for the element from which it was called. To see information about other elements, query the SYS.GV$LOG_HOLDS system table.

Related views

This procedure has these related views.

SYS.GV$LOG_HOLDS
SYS.V$LOG_HOLDS

Syntax

```
ttLogHolds()
```

Parameters

ttLogHolds has no parameters.

Result set

ttLogHolds returns the result set:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HoldLFN</td>
<td>TT_INTEGER NOT NULL</td>
<td>Returns the transaction log file number of the hold.</td>
</tr>
<tr>
<td>HoldLFO</td>
<td>TT_BIGINT NOT NULL</td>
<td>Returns the transaction log file offset of the hold.</td>
</tr>
</tbody>
</table>
**Examples**

CALL ttLogHolds();
< 0, 1149544, Long-Running XA Transaction, 
0x1-476c6f62616c-5861637431 >
< 0, 1149752, Long-Running Transaction, 4.2 >
< 0, 1149992, Checkpoint, sample.ds1 >
< 0, 1150168, Checkpoint, sample.ds0 >
4 rows found.

The following example shows the output of `ttLogHolds` built-in procedure for an active standby pair replication scheme, where the active master is `master1` and the standby master is `master2` with a single subscriber, `subscriber1`.

Command> call ttLogHolds;
< 0, 3569664, Checkpoint, master1.ds0 >
< 0, 15742976, Checkpoint, master1.ds1 >
< 0, 16351496, Replication, ADC6160529:SUBSCRIBER1 >
< 0, 16351640, Replication, ADC6160529:MMASTER2 >
4 rows found.

The following example shows the progress of the asynchronous propagation for an AWT cache group to the Oracle database. The description field contains "_ORACLE" to identify the transaction log hold for the AWT cache group propagation.

Command> call ttLogHolds();
< 0, 18958336, Checkpoint, cachealone1.ds0 >
<table>
<thead>
<tr>
<th>Time stamp</th>
<th>Event Type</th>
<th>Site Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>0, 19048448</td>
<td>Checkpoint</td>
<td>cachealone1.ds1</td>
</tr>
<tr>
<td>0, 19050904</td>
<td>Replication</td>
<td>ADC6160529: _ORACLE</td>
</tr>
</tbody>
</table>

3 rows found.
ttMonitorHighWaterReset

Description
This procedure sets the value of PERM_IN_USE_HIGH_WATER column in the MONITOR table to the current value of the PERM_IN_USE_SIZE column and sets the value of the TEMP_IN_USE_HIGH_WATER column in the MONITOR table to the current value of TEMP_IN_USE_SIZE column. These columns are useful for sizing databases during application development and deployment.

Required privilege
This procedure requires the ADMIN privilege.

Usage in TimesTen Scaleout
TimesTen Scaleout applications can call this built-in procedure.
This procedure executes on all elements in the grid.

Related views
This procedure has no related views.

Syntax

```
ttMonitorHighWaterReset()
```

Parameters

```
ttMonitorHighWaterReset has no parameters.
```

Result set

```
ttMonitorHighWaterReset returns no results.
```

Examples

```
CALL ttMonitorHighWaterReset();
```
ttOptClearStats

Description
This procedure clears the statistics for the specified table, causing the TimesTen query optimizer to use estimates or default values for subsequent queries involving the table. The procedure is useful if statistics are assumed to be out of date and an application wants to use built-in default values. This procedure removes all rows from the TBL_STATS and COL_STATS system tables that pertain to the specified tables. See "SYS.TBL_STATS" and "SYS.COL_STATS" in Oracle TimesTen In-Memory Database System Tables and Views Reference.

Required privilege
This procedure requires no privilege for the table owner. This procedure requires no privilege if tblName is not specified, because the procedure operates on the current user's tables if tblName is not specified.

This procedure requires the ALTER ANY TABLE privilege if user is not the table owner.

Usage in TimesTen Scaleout
TimesTen Scaleout applications can call this built-in procedure.

This procedure executes locally on the element from which it is called.

Related views
This procedure has no related views.

Syntax

```
ttOptClearStats('tblName', invalidate)
```

Parameters

**Parameter** | **Type** | **Description**
--- | --- | ---
`tblName` | TT_CHAR (61) | Name of an application table. Can include table owner. If `tblName` is the empty string or is not specified, statistics are cleared for all the current user's tables in the database.

Using a synonym to specify a table name is not supported.

`invalidate` | TT_INTEGER | 0 (no) or 1 (yes). Default is 0.

If `invalidate` is 1, all commands that reference the affected tables are reprepared automatically when they are re-executed, including commands prepared by other users.

If `invalidate` is 0, the statistics are not considered modified and existing commands are not reprepared.

Result set

`ttOptClearStats` returns no results.
Examples

CALL ttOptClearStats ( 'SALLY.ACCTS', 1 );

Clears the statistics for the SALLY.ACCTS table and reprepares all commands that affect the ACCTS table.

CALL ttOptClearStats();

Clears the statistics for all the current user's tables and reprepares all commands that affect these tables.

CALL ttOptClearStats('', 0);

Clears the statistics for all the current user's tables without repreparing commands that reference these tables.

See also

ttOptEstimateStats
ttOptSetColIntvlStats
ttOptSetFlag
ttOptSetOrder
ttOptSetTblStats
ttOptUpdateStats
ttPLSQLMemoryStats
ttOptCmdCacheInvalidate

Description
This built-in procedure either forces a recompilation should a dependent command be invoked again, or removes such command from the cache and it must be re-prepared by the user.

Scenarios in which you may want to call this procedure include:

- After all needed statistics have been collected.
- When table cardinalities have been changed significantly.

The procedure either marks a command as needing recompilation or as invalidated. Neither option stops execution of a command.

Required privilege
This procedure requires the \texttt{DDL} privilege.

Usage in TimesTen Scaleout
TimesTen Scaleout applications can call this built-in procedure.
This procedure executes locally on the element from which it is called.

Related views
This procedure has no related views.

Syntax
\texttt{ttOptCmdCacheInvalidate('tblName', invalidate)}

Parameters
\texttt{ttOptCmdCacheInvalidate} has these parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tblName</td>
<td>\texttt{TT_CHAR(61)}</td>
<td>The name of the table for which the dependent commands should be invalidated or recompiled.</td>
</tr>
</tbody>
</table>
| invalidate | \texttt{TT\_INTEGER} | Forces recompilation or invalidates the dependent commands.  
1 - Indicates that the commands should be recompiled. The command is recompiled during its first use after calling this built-in procedure. (default)  
2 - Indicates that the commands should be invalidated. The command is not reused or recompiled again. If you call the command after you have marked it for invalidation, TimesTen returns an error. |

Result set
\texttt{ttOptCmdCacheInvalidate} returns no results.
Examples

To recompile dependent commands on the table tab1, use:
CALL ttOptCmdCacheInvalidate ('tab1', 1);

To invalidate the dependent commands on table tab1, use:
CALL ttOptCmdCacheInvalidate ('tab1', 2);

See also

ttOptClearStats
ttOptEstimateStats
ttOptSetColIntvlStats
ttOptSetFlag
ttOptSetOrder
ttOptSetTblStats
ttOptUpdateStats
ttPLSQLMemoryStats
The `ttOptEstimateStats` procedure updates the statistics for the specified table. This procedure estimates statistics by looking at a random sample of the rows in the specified table(s). The sample size is the number of rows specified (if `sampleStr` has the form 'n ROWS') or a percentage of the total number of rows (if `sampleStr` has the form 'p PERCENT').

The procedure operates on all tables owned by the current user if `.tblName` is not specified. If the user is the instance administrator, only tables owned by the instance administrator are updated. If the tables are not owned by the user, the user can qualify the table name with their own user name to update stats for the current user.

To determine if your stats are updated, look at the system tables, `SYS.COL_STATS` and `SYS.TBL_STATS`, before and after you perform this operation.

**Required privilege**

This procedure requires no privilege if the user is the table owner, or if `tblName` is not specified.

This procedure requires the `ALTER ANY TABLE` privilege if the user is not the table owner.

**Usage in TimesTen Scaleout**

TimesTen Scaleout applications can call this built-in procedure.

This procedure executes on all elements in the grid.

**Related views**

This procedure has these related views.

- `SYS.GV$OPT_COL_STATS`
- `SYS.V$OPT_COL_STATS`

**Syntax**

```
ttOptEstimateStats(["tblName"], [invalidate], 'sampleStr')
```

**Parameters**

`ttOptEstimateStats` has these parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>tblName</code></td>
<td><code>TT_CHAR(61)</code></td>
<td>Name of an application table. Can include table owner. If <code>tblName</code> is an empty string, statistics are estimated for all the current user's tables in the database. Using a synonym to specify a table name is not supported.</td>
</tr>
</tbody>
</table>
ttOptEstimateStats

**Result set**

`ttOptEstimateStats` returns no results.

**Examples**

```
CALL ttOptEstimateStats ( 'ACCTS', 1, '5 PERCENT' );

CALL ttOptEstimateStats ( 'ACCTS', 1, '75 ROWS' );
```

**Notes**

The TimesTen statistics include the number of rows in each table, the number of unique values in each column, and the minimum and maximum values in each column. TimesTen assumes a uniform distribution of column values.

This procedure only runs faster than `ttOptUpdateStats` when you sample less than 50 percent of the rows in the table.

Estimates are not computed on columns that are longer than 2,048 bytes, and statistics for these columns are not updated. To update statistics on columns longer than 2,048 bytes, use the `ttOptUpdateStats` built-in procedure. (For varying length columns, this procedure updates statistics only if the column has a maximum length of 2,048 bytes or less.)

If a very small value is chosen for the `sampleStr` parameter, this procedure runs quickly but may result in suboptimal execution plans. For “good” distributions of data, a 10 percent selection is a good choice for computing statistics quickly without sacrificing plan accuracy. If the number of rows specified is large or the table in question is small, to improve performance TimesTen computes exact statistics on all columns that have a length of 2,048 bytes or less. For example, the only difference between

```
  ttOptEstimateStats ('ACCTS', 1, '100 PERCENT')
```

and

```
  ttOptUpdateStats ( 'ACCTS', 1 )
```

is that the former does not compute statistics for long columns.

The statistics are stored in the TBL_STATS and COL_STATS system tables.

---

**Parameter Type Description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>invalidate</td>
<td>TT_INTEGER</td>
<td>0 (no) or 1 (yes). If <code>invalidate</code> is 1, all commands that reference the affected tables are automatically prepared again when re-executed, including commands prepared by other users. If <code>invalidate</code> is 0, the statistics are not considered to have been modified and existing commands are not reprepared. The <code>invalidate</code> parameter is optional and defaults to 0.</td>
</tr>
<tr>
<td>sampleStr</td>
<td>TT_VARCHAR (255) NOT NULL</td>
<td>String of the form 'n ROWS', where n is an INTEGER greater than zero; or 'p PERCENT', where p is a floating point number between 0.0 and 100.0 inclusive.</td>
</tr>
</tbody>
</table>
For performance reasons, \texttt{ttOptEstimateStats} does not hold a lock on tables or rows when computing statistics. Computing statistics can still slow performance. Estimating statistics generally provides better performance than computing exact statistics.

If you estimate or update statistics with an empty table list, statistics on system tables are updated also, if you have privileges to update the system tables.

\textbf{See also}

- \texttt{ttOptSetColIntvlStats}
- \texttt{ttOptSetFlag}
- \texttt{ttOptSetOrder}
- \texttt{ttOptSetTblStats}
- \texttt{ttOptUpdateStats}
- \texttt{ttPLSQLMemoryStats}
ttOptGetColStats

Description

This procedure returns statistics information in text format.

Required privilege

This procedure requires the SELECT privilege on the specified tables.

Usage in TimesTen Scaleout

TimesTen Scaleout applications can call this built-in procedure.

This procedure executes locally on the element from which it is called.

Related views

This procedure has these related views.

SYS.GV$OPT_COL_STATS
SYS.V$OPT_COL_STATS

Syntax

```	ttOptGetColStats('tblName', 'colName')
```

Parameters

```
tttOptGetColStats has these parameters:
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tblName</td>
<td>TT CHAR (61)</td>
<td>Name of the table whose statistics are to be returned. If NULL is passed, then values for all tables are returned. Using a synonym to specify a table name is not supported.</td>
</tr>
<tr>
<td>colName</td>
<td>TT CHAR (30)</td>
<td>Name of the column for which statistics should be returned. If NULL is passed, statistics for all columns in the specified table are returned.</td>
</tr>
</tbody>
</table>

Result set

```
tttOptGetColStats returns the result set:
```

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tblName</td>
<td>TT CHAR (30)</td>
<td>Name of the table. Using a synonym to specify a table name is not supported.</td>
</tr>
<tr>
<td>colName</td>
<td>TT CHAR (30)</td>
<td>Name of the column.</td>
</tr>
<tr>
<td>stats</td>
<td>TT VARCHAR (409600) NOT NULL</td>
<td>Statistics in text form.</td>
</tr>
</tbody>
</table>
Examples

CALL ttOptGetColStats ();
< T1 , X1, (2, 10, 10, 100 (,4, 40, 10 ,1, 10, 5) ,
(4, 20, 20 ,11, 20, 15) )>

See also

ttOptSetColStats
ttOptSetColIntvlStats
ttOptGetFlag

Description
This procedure returns the optimizer flag settings for the current transaction. The results are returned as a result set that can be retrieved using the ODBC SQLFetch function or the JDBC ResultSet.getXXX() method, just like the result of a SQL SELECT statement. Applications can request the value of a specific optimizer flag by passing the flag name to ttOptGetFlag. Alternatively, applications can request the values of all the optimizer flags by passing NULL. The optimizer flags and their meanings are described under the ttOptSetFlag built-in procedure.

Required privilege
This procedure requires no privilege.

Usage in TimesTen Scaleout
TimesTen Scaleout applications can call this built-in procedure.
This procedure executes locally on the element from which it is called.

Related views
This procedure has no related views.

Syntax
```
ttOptGetFlag('flagName')
```

Parameters
ttOptGetFlag has the parameter:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>flagName</td>
<td>TT_CHAR (32)</td>
<td>Name of the flag whose value is to be returned. If NULL is passed, the values of all flags are returned.</td>
</tr>
</tbody>
</table>

Result set

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>flagName</td>
<td>TT_VARCHAR (32) NOT NULL</td>
<td>Name of the flag. See &quot;ttOptSetFlag&quot; on page 3-161 for a description of possible flag values.</td>
</tr>
<tr>
<td>value</td>
<td>TT_INTEGER NOT NULL</td>
<td>Current flag value, either 0 or 1.</td>
</tr>
</tbody>
</table>

Examples
```
CALL ttOptGetFlag('TmpHash');
```

See also
```
ttOptSetFlag
```
ttOptGetMaxCmdFreeListCnt

**Description**

This procedure returns the size of the free list of SQL compiled command cache. To reset the size of the cache, use `ttOptSetMaxPriCmdFreeListCnt` for materialized views and `ttOptSetMaxCmdFreeListCnt` for regular tables.

**Required privilege**

This procedure requires no privilege.

**Parameters**

`ttOptGetMaxCmdFreeListCnt` has no parameters.

**Usage in TimesTen Scaleout**

TimesTen Scaleout applications can call this built-in procedure. This procedure executes locally on the element from which it is called.

**Related views**

This procedure has these related views.

- `SYS.GV$OPT_MAX_CMD_FREELIST_CNT`
- `SYS.V$OPT_MAX_CMD_FREELIST_CNT`

**Syntax**

`ttOptGetMaxCmdFreeListCnt()`

**Result set**

`ttOptGetMaxCmdFreeListCnt` returns the results.

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>retVal</td>
<td>TT_VARCHAR (200) NOT NULL</td>
<td>The size of the SQL compiled command cache.</td>
</tr>
</tbody>
</table>

**Examples**

```sql
CALL ttOptGetMaxCmdFreeListCnt();
```

**See also**

- `ttOptSetMaxPriCmdFreeListCnt`
- `ttOptSetMaxCmdFreeListCnt`
ttOptGetOrder

Description
This procedure returns a single-row result set containing the join order for the current transaction. This result set can be retrieved using the ODBC SQLFetch function or the JDBC ResultSet.getXXX() method, just like the result of a SQL SELECT statement. Join orders are described under the ttOptSetOrder built-in procedure.

Required privilege
This procedure requires no privilege.

Usage in TimesTen Scaleout
This procedure is supported in TimesTen Scaleout.
This procedure returns a row for the element from which it was called. To see information about other elements, query the SYS.GV$OPT_ORDER system table.

Related views
This procedure has these related views.
SYS.GV$OPT_ORDER
SYS.V$OPT_ORDER

Syntax
ttOptGetOrder( )

Parameters
ttOptGetOrder has no parameters.

Result set
ttOptGetOrder returns the result set:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>joinOrder</td>
<td>TT_VARCHAR(1024) NOT NULL</td>
<td>Optimizer join order for the current transaction.</td>
</tr>
</tbody>
</table>

Examples
CALL ttOptGetOrder;

See also
ttOptSetOrder
ttOptSetColIntvlStats

Description
This procedure modifies the statistics for the specified columns with interval information. This procedure enables an application to set statistics manually rather than have TimesTen automatically compute them. This feature is useful for preparing commands before the data has been inserted or for seeing how table characteristics can affect the choice of execution plan. This procedure modifies the relevant row(s) in the COL_STATS system table. Modifying interval statistics for a column that is not currently indexed has no effect.

Because this procedure can be used before any data is in the table, the values specified do not need to bear any relation to the actual values, although some basic validity checking is performed.

Required privilege
This procedure requires no privilege (if owner) or ALTER ANY TABLE privilege (if not owner).

Usage in TimesTen Scaleout
TimesTen Scaleout applications can call this built-in procedure. This procedure executes on all elements in the grid.

Related views
This procedure has no related views.

Syntax
`ttOptSetColIntvlStats('tblName', 'colName', invalidate, (stats))`

Parameters
`ttOptSetColIntvlStats` has these parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tblName</td>
<td>TT_CHAR(61) NOT NULL</td>
<td>Name of an application table. Can include table owner. Using a synonym to specify a table name is not supported.</td>
</tr>
<tr>
<td>colName</td>
<td>TT_CHAR(30) NOT NULL</td>
<td>Name of a column in that table.</td>
</tr>
<tr>
<td>invalidate</td>
<td>TT_INTEGER</td>
<td>0 (no) or 1 (yes). If <code>invalidate</code> is 1, all commands that reference the affected tables are automatically prepared again when re-executed. This includes commands prepared by other users. If <code>invalidate</code> is 0, the statistics are not considered to have been modified and existing commands are not reprepared.</td>
</tr>
</tbody>
</table>
**ttOptSetColIntvlStats**

### Result set

`ttOptSetColIntvlStats` returns no results.

### Examples

To set the following statistics for column `t1.x1`:

- Two intervals
- Integer type
- 10 rows with null value
- 10 unique value
- 100 rows
- **Interval 1** (4 unique values besides the most frequently occurring value, 40 rows with values other than most frequently occurring value, 10 rows with most frequently occurring value, `min = 1, max = 10, mod = 5`)
- **Interval 2** (4 unique values besides the most frequently occurring value, 20 rows with values other than most frequently occurring, 20 rows with most frequently occurring value, `min = 11, max = 20, mod = 15`)

Use the statement:

```sql
CALL ttOptSetColIntvlStats('t1', 'x1', 1, (2, 10, 10, 100, (4, 40, 10, 10, 5), (4, 20, 20, 11, 20, 15)));
```

### Notes

You must specify the minimum and maximum values in the interval as VARBINARY. NULL values are not permitted as minimum or maximum values. The value is stored in the platform-specific endian format.

### See also

- `ttOptEstimateStats`
- `ttOptGetColStats`
ttOptSetColIntvlStats

- ttOptSetColStats
- ttOptSetTblStats
- ttOptUpdateStats
ttOptSetColStats

Description

This procedure modifies the statistics for the specified columns. This procedure enables an application to set statistics manually rather than have TimesTen automatically compute them. This feature is useful for preparing commands before the data has been inserted or for seeing how table characteristics can affect the choice of execution plan. This procedure modifies the relevant row(s) in the COL_STATS system table.

Because this procedure can be used before the table is populated with data, the values specified do not need to bear any relation to the actual values, although some basic validity checking is performed.

Required privilege

This procedure requires no privilege (if owner) or ALTER ANY TABLE privilege (if not owner).

Usage in TimesTen Scaleout

TimesTen Scaleout applications can call this built-in procedure.

This procedure executes on all elements in the grid.

Related views

This procedure has no related views.

Syntax

\texttt{ttOptSetColStats('tblName', 'colName', numUniq, minVal, maxVal, invalidate, numNull)}

Parameters

\texttt{ttOptSetColStats} has these parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tblName</td>
<td>TT_CHAR(61) NOT NULL</td>
<td>Name of an application table. Can include table owner. Using a synonym to specify a table name is not supported.</td>
</tr>
<tr>
<td>colName</td>
<td>TT_CHAR(30) NOT NULL</td>
<td>Name of a column in that table.</td>
</tr>
<tr>
<td>num_Uniq</td>
<td>TT_INTEGER NOT NULL</td>
<td>Number of unique values in the column.</td>
</tr>
<tr>
<td>minVal</td>
<td>VARBINARY(1024) NOT NULL</td>
<td>Minimum value in the column (possibly truncated).</td>
</tr>
<tr>
<td>maxVal</td>
<td>VARBINARY(1024) NOT NULL</td>
<td>Maximum value in the column (possibly truncated).</td>
</tr>
</tbody>
</table>
ttOptSetColStats returns no results.

Examples

CALL ttOptSetColStats ('SALLY.ACCTS, 'BALANCE, 400, 0x00001388, 0x000186A0, 1, 0);

Notes

You must specify the minimum and maximum values as VARBINARY. NULL values are not permitted as minimum or maximum values. The value is stored in the platform-specific endian format.

The statistics are treated as a single interval of column values that are uniformly distributed between the minimum value and the maximum value.

See also

- ttOptEstimateStats
- ttOptGetColStats
- ttOptSetColIntvlStats
- ttOptSetTblStats
- ttOptUpdateStats

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>
ttOptSetFlag

**Description**

This procedure resets all optimizer flags to their default values when the transaction has been committed or rolled back. This alters the generation of execution plans by the TimesTen query optimizer. It sets flags to enable or disable the use of various access methods. The changes made by this call take effect during preparation of statements and affect all subsequent calls to the ODBC functions `SQLPrepare` and `SQLExecDirect` or the JDBC methods `Connection.prepareCall` and `Statement.execute` in the current transaction. If optimizer flags are set while `AutoCommit` is on, they are ignored.

**Required privilege**

This procedure requires no privilege.

**Usage in TimesTen Scaleout**

TimesTen Scaleout applications can call this built-in procedure. This procedure executes locally on the element from which it is called.

**Related views**

This procedure has these related views.

- `SYS.GV$OPT_FLAG`
- `SYS.V$OPT_FLAG`

**Syntax**

```
ttOptSetFlag('optFlag', optVal)
```

**Parameters**

`ttOptSetFlag` has these parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>optFlag</td>
<td><code>TT_CHAR(32)</code></td>
<td>Name of optimizer flag.</td>
</tr>
<tr>
<td>optVal</td>
<td><code>TT_INTEGER</code></td>
<td>The value of the optimizer flag. The value is generally 0 (disable/disallow) or 1 (enable/allow), except as described under &quot;Optimizer flags&quot; below.</td>
</tr>
</tbody>
</table>

**Optimizer flags**

When setting the optimizer flags, use the following character strings, which are not case sensitive:

<table>
<thead>
<tr>
<th>Flag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BranchAndBound</td>
<td>Enables or disables branch and bound optimization. If enabled, TimesTen calculates the maximum cost of the query plan during a “zero phase,” at the very beginning of the optimization process. If disabled, TimesTen does not perform this cost analysis.</td>
</tr>
<tr>
<td>Flag</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DynamicLoadEnable</td>
<td>Enables or disables dynamic load of data from an Oracle database to a TimesTen dynamic cache group. By default, dynamic load of data from an Oracle database is enabled.</td>
</tr>
<tr>
<td>DynamicLoadErrorMode</td>
<td>Enables or disables dynamic load error mode. It controls output of error messages upon failure of a transparent load operation on a TimesTen dynamic cache group. Disabled by default.</td>
</tr>
<tr>
<td>FirstRow</td>
<td>Enables or disables first row optimization in a <code>SELECT</code>, <code>UPDATE</code> or <code>DELETE</code> statement. If the SQL keyword <code>FIRST</code> is used in the SQL statement, it takes precedence over this optimizer hint. The <code>FIRST</code> keyword enables first row optimization.</td>
</tr>
<tr>
<td>ForceCompile</td>
<td>Enables or disables forced compilation. If enabled, TimesTen recompiles the query and regenerates the query plan each time. If disabled, TimesTen does not compile the query plan even if it is available.</td>
</tr>
<tr>
<td>GenPlan</td>
<td>Enables or disables the creation of entries in the <code>PLAN</code> table for the rest of the transaction. For an example, see &quot;Instruct TimesTen to store the plan in the system <code>PLAN</code> table&quot; in Oracle TimesTen In-Memory Database Operations Guide.</td>
</tr>
<tr>
<td>Hash</td>
<td>Enables or disables the use of existing hash indexes in indexed table scans.</td>
</tr>
<tr>
<td>HashGb</td>
<td>Enables or disables the use of hash groups.</td>
</tr>
<tr>
<td>IndexedOR</td>
<td>Enables or disables serialized table scans. If disabled, TimesTen uses serialized table scans for <code>IN...list</code> conditions, else TimesTen uses multiple index scans for an <code>OR</code> condition.</td>
</tr>
<tr>
<td>MergeJoin</td>
<td>Enables or disables the use of merge joins.</td>
</tr>
<tr>
<td>NestedLoop</td>
<td>Refers to a common way of joining two tables.</td>
</tr>
<tr>
<td>NoRemRowIdOpt</td>
<td>Enables or disables internal generation of RowIDs. If enabled, RowIDs are not internally generated for optimization purposes. If disabled, RowIDs may be internally generated, even if the row is not in the <code>SELECT</code> list.</td>
</tr>
</tbody>
</table>
In addition, you can use the string **AllFlags** to refer to all optimizer flags, and the string **Default** to refer to the default flags. **Default** excludes the **GenPlan** flag but includes all other optimizer flags.

<table>
<thead>
<tr>
<th>Flag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PassThrough</td>
<td>Temporarily changes the pass through level for TimesTen Cache applications. The pass through level can be set at any time and takes effect immediately. Legal values for this flag are: 0 - (default) - SQL statements are executed only on TimesTen. 1 - INSERT, UPDATE and DELETE statements are executed on TimesTen unless they reference one or more tables that are not in TimesTen. If they reference one or more tables not in TimesTen, they are passed through to the Oracle database. DDL statements are executed on TimesTen. Other statements are passed through to the Oracle database if they generate a syntax error in TimesTen or if one or more tables referenced within the statement are not in TimesTen. 2 - INSERT, UPDATE and DELETE statements performed on tables in read-only cache groups or user managed cache groups with the <strong>READONLY</strong> cache table attribute are passed through to the Oracle database. Passthrough behavior for other cache group types is the same as <strong>PassThrough=1</strong>. 3 - All statements are passed through to the Oracle database for execution.</td>
</tr>
<tr>
<td>Range</td>
<td>Enables or disables the use of existing range indexes in indexed table scans.</td>
</tr>
<tr>
<td>Rowid</td>
<td>Enables or disables the use of Row IDs.</td>
</tr>
<tr>
<td>RowLock</td>
<td>Allows or disallows the optimizer to consider using row locks.</td>
</tr>
<tr>
<td>Scan</td>
<td>Refers to full table scans.</td>
</tr>
<tr>
<td>ShowJoinOrder</td>
<td>Shows the join order of the tables in an optimizer scan.</td>
</tr>
<tr>
<td>TblLock</td>
<td>Enables or disables the optimizer to consider using table locks.</td>
</tr>
<tr>
<td>TmpHash</td>
<td>Enables or disables the use of a temporary hash scan. This is an index that is created during execution for use in evaluating the statement. Though index creation is time-consuming, it can save time when evaluating join predicates.</td>
</tr>
<tr>
<td>TmpRange</td>
<td>Performs a temporary range scan. Can also be used so that values are sorted for a merge join. Though index creation is time-consuming, it can save time when evaluating join predicates.</td>
</tr>
<tr>
<td>TmpTable</td>
<td>Stores intermediate results into a temporary table. This operation is sometimes chosen to avoid repeated evaluation of predicates in join queries or sometimes just to allow faster scans of intermediate results in joins.</td>
</tr>
<tr>
<td>UseBoyerMooreStringSearch</td>
<td>Enables or disables the Boyer-Moore string search algorithm. If enabled, Boyer-Moore string search algorithm is enabled. This can improve performance of <strong>LIKE</strong> operations.</td>
</tr>
</tbody>
</table>
Flag description

The value of each flag can be 1 or 0:

- If 1, the operation is enabled
- If 0, the operation is disabled unless absolutely necessary

Initially, all the flag values except GenPlan are 1 (all operations are permitted).

For example, an application can prevent the optimizer from choosing a plan that stores intermediate results:

```sql
ttOptSetFlag ('TmpTable', 0 )
```

Similarly, an application can specify a preference for MergeJoin:

```sql
ttOptSetFlag ('MergeJoin', 0 )
```

In the second example, the optimizer may still choose a nested loop join if a merge join is impossible (for example, if there is no merge-join predicate). Similarly, the optimizer may occasionally not be able to satisfy an application request to avoid table scans (when the Scan flag is set to 0).

You cannot specify that a particular operation is prohibited only at a certain step of a plan or that a particular join method always be done between two specific tables. Similarly, there is no way to specify that certain indexes be used or that a hash index be used to evaluate a specific predicate. Each operation is either fully permitted or fully restricted.

When a command is prepared, the current optimizer flags, index hints and join order are maintained in the structure of the compiled form of the command and are used if the command is ever reprepared by the system. See “The TimesTen Query Optimizer” in Oracle TimesTen In-Memory Database Operations Guide for an example of reprepared statements.

If both RowLock and TblLock are disabled, TimesTen uses row-locking. If both RowLock and TblLock are enabled, TimesTen uses the locking scheme that is most likely to have better performance:

<table>
<thead>
<tr>
<th>TblLock status</th>
<th>RowLock status</th>
<th>Effect on the optimizer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disabled</td>
<td>Disabled</td>
<td>Use row-level locking.</td>
</tr>
<tr>
<td>Enabled</td>
<td>Disabled</td>
<td>Use table-level locking.</td>
</tr>
<tr>
<td>Disabled</td>
<td>Enabled</td>
<td>Use row-level locking.</td>
</tr>
<tr>
<td>Enabled</td>
<td>Enabled</td>
<td>Optimizer chooses row-level or table-level locking.</td>
</tr>
</tbody>
</table>

In general, table-level locking is useful when a query accesses a significant portion of the rows of a table or when there are very few concurrent transactions accessing the table.

Result set

`ttOptSetFlag` returns no results.

Examples

```sql
CALL ttOptSetFlag ('TmpHash', 1);
```
Notes

You can also set the join order using statement level optimizer hints in certain SQL statements. For details, see "Statement level optimizer hints" in the Oracle TimesTen In-Memory Database SQL Reference. Specifically, see the table, "Differences between statement level and transaction level optimizer hints" to understand the behavior of each style of hint.

See also

- ttOptEstimateStats
- ttOptGetFlag
- ttOptGetOrder
- ttOptSetColIntvlStats
- ttOptSetOrder
- ttOptSetTblStats
- ttOptUpdateStats
- ttPLSQLMemoryStats
ttOptSetMaxCmdFreeListCnt

Description
This procedure sets the maximum count of the free list of SQL compiled commands for regular tables. To get the current setting use the ttOptGetMaxCmdFreeListCnt procedure.

Required privilege
This procedure requires the ADMIN privilege.

Syntax

```
ttOptSetMaxCmdFreeListCnt(maxCnt)
```

Parameters
ttOptSetMaxCmdFreeListCnt has the required parameter:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>maxCnt</td>
<td>TT_INTEGER NOT NULL</td>
<td>The max number of free SQL compiled commands for regular tables.</td>
</tr>
</tbody>
</table>

Result set

ttOptSetMaxCmdFreeListCnt returns no results.

Examples

```
CALL ttOptSetMaxCmdFreeListCnt(40);
```

See also

ttOptGetMaxCmdFreeListCnt
**ttOptSetMaxPriCmdFreeListCnt**

**Description**

This procedure sets the maximum count of the free list of SQL compiled commands that perform materialized view maintenance.

When this command is set, freeable materialized view compiled commands are counted separately from those of regular tables. If this command is not set, materialized view compiled commands are counted as regular commands.

**Required privilege**

This procedure requires the ADMIN privilege.

**Usage in TimesTen Scaleout**

TimesTen Scaleout applications can call this built-in procedure.

This procedure executes on all elements in the grid.

**Related views**

This procedure has no related views.

**Syntax**

```
ttOptSetMaxPriCmdFreeListCnt(maxCnt)
```

**Parameters**

`ttOptSetMaxPriCmdFreeListCnt` has the required parameter:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>maxCnt</td>
<td>TT_INTEGER NOT NULL</td>
<td>The size of the SQL compiled command cache.</td>
</tr>
</tbody>
</table>

**Result set**

`ttOptSetMaxPriCmdFreeListCnt` returns no results.

**Examples**

```
CALL ttOptSetMaxPriCmdFreeListCnt(40);
```

**See also**

- `ttOptGetMaxCmdFreeListCnt`
- `ttOptSetMaxCmdFreeListCnt`
ttOptSetOrder

Description
This procedure specifies the order in which tables should be joined by the optimizer. The character string is a list of table names or table correlation names referenced in the query or a subquery, separated by spaces (not commas). The table listed first is scanned first by the plan. (It is outermost in a nested loop join, for example.) A correlation name is a shortcut or alias for a qualified table name. AutoCommit must be set to OFF when running this built-in procedure.

Required privilege
This procedure requires no privilege.

Usage in TimesTen Scaleout
TimesTen Scaleout applications can call this built-in procedure. This procedure executes locally on the element from which it is called.

Related views
This procedure has no related views.

Syntax
`ttOptSetOrder('joinOrder')`

Parameters
`ttOptSetOrder` has the required parameter:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>join_Order</td>
<td>TT_VARCHAR(1024)</td>
<td>List of space-separated table or table correlation names. If an owner is required to distinguish the table name, use a table correlation name. If the <code>joinOrder</code> is not specified the query optimizer reverts to its default behavior.</td>
</tr>
</tbody>
</table>

Result set
`ttOptSetOrder` returns no results.

Examples
```
CALL ttOptSetOrder ('EMPS DEPTS ACCTS');
```
If an application makes the call:
```
call ttOptSetOrder('ORDERS CUSTOMERS');
```
The optimizer scans the ORDERS table before scanning the CUSTOMERS when evaluating the following query that lists all the customers who have at least one unshipped order:
```
SELECT CUSTOMERS.NAME
FROM CUSTOMERS
WHERE EXISTS (SELECT 1
```
FROM  ORDERS
WHERE  CUSTOMERS.ID = ORDERS.CUSTID
AND ORDER.STATUS = 'UN-SHIPPED');

Consider an application that makes the following call.

```
ttOptSetOrder('DEPTS EMPS ACCTS');
```

The optimizer is prevented from executing a join between `DEPTS` and `ACCTS` when evaluating the number of employees working on a specific account:

```
SELECT COUNT(DISTINCT EMPS.ID)
FROM ACCTS, DEPTS, EMPS
WHERE ACCTS.DEPTS = DEPTS.ID
AND EMPS.DEPTS = DEPTS.ID
AND ACCTS.NUM = :AcctNum
```

If the application does not reset the join order and tries to prepare a command that does not reference each of the three tables (and no others), the optimizer issues warning number 965. The specified join order is not applicable. TimesTen considers valid join orders and ignores the specified join order when preparing the command.

### Notes

A table alias name for a derived table is not supported in the join order. If you specify a table alias name, TimesTen returns the warning message 965 that indicates the order cannot be honored.

The string length is limited to 1,024 bytes. If a string exceeds this length, it is truncated and a warning is issued.

When correlation names referenced in subqueries are in included in the order, TimesTen may internally change the isolation mode.

When a command is prepared, the current optimizer flags, index hints, and join order are maintained in the structure of the compiled form of the command and are used if the command is ever reprepared by the system. See "The TimesTen Query Optimizer" in *Oracle TimesTen In-Memory Database Operations Guide* for an example of reprepared statements.

The changes made by this call take effect immediately and affect all subsequent calls to the ODBC function `SQLPrepare` or the JDBC method `Connection.prepareStatement` in the current transaction. The query optimizer reverts to its default behavior for subsequent transactions.

The tables referenced by a query must exactly match the names given if the join order is to be used (the comparisons are not case sensitive). A complete ordering must be specified; there is no mechanism for specifying partial orders. If the query has a subquery then the join order should also reference the correlation names in the subquery. In essence, the join order should reference all the correlation names referenced in the query. The TimesTen optimizer internally implements a subquery as a special kind of join query with a `GROUP BY`. For the join order to be applicable it should reference all the correlation names. If there is a discrepancy, TimesTen issues a warning and ignores the specified join order completely.

You can also set the join order using statement level optimizer hints in certain SQL statements. For details, see "Statement level optimizer hints" in the *Oracle TimesTen In-Memory Database SQL Reference*. Specifically, see the section, "Differences between statement level and transaction level optimizer hints" to understand the behavior of each style of hint.
See also

- `ttOptEstimateStats`
- `ttOptGetFlag`
- `ttOptGetOrder`
- `ttOptSetColIntvlStats`
- `ttOptSetFlag`
- `ttOptSetTblStats`
- `ttOptUpdateStats`
- `ttPLSQLMemoryStats`
ttOptSetTblStats

Description
This procedure modifies the statistics for the specified table. This procedure enables an application to set statistics explicitly rather than have TimesTen automatically compute them.

Required privilege
This procedure requires no privilege (if owner) or ALTER ANY TABLE privilege (if not owner).

Usage in TimesTen Scaleout
TimesTen Scaleout applications can call this built-in procedure.
This procedure executes on all elements in the grid.

Related views
This procedure has no related views.

Syntax
```
ttOptSetTblStats('tblName', numRows, invalidate)
```

Parameters
```
ttOptSetTblStats has these parameters:
```
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tblName</td>
<td>TT_CHAR(61) NOT NULL</td>
<td>Name of an application table. Can include table owner. Using a synonym to specify a table name is not supported.</td>
</tr>
<tr>
<td>numRows</td>
<td>TT_INTEGER NOT NULL</td>
<td>Number of rows in the table.</td>
</tr>
<tr>
<td>invalidate</td>
<td>TT_INTEGER</td>
<td>0 (no) or 1 (yes). If invalidate is 1, all commands that reference the affected tables are automatically prepared again when re-executed, including commands prepared by other users. If invalidate is 0, the statistics are not considered to have been modified and existing commands are not reprepared.</td>
</tr>
</tbody>
</table>

Result set
```
ttOptSetTblStats returns no results.
```

Examples
```
CALL ttOptSetTblStats ( 'ACCTS', 10000, 0 );
```
Notes

This feature is useful for preparing commands before the data has been inserted or for seeing how table size can affect the choice of an execution plan. Because the command can be used before any data is in the table, the values specified do not need to bear any relation to the actual values. This procedure modifies the relevant row(s) in the TBL_STATS system table. See "SYS.TBL_STATS" in Oracle TimesTen In-Memory Database System Tables and Views Reference.

See also

- `ttOptEstimateStats`
- `ttOptGetFlag`
- `ttOptGetOrder`
- `ttOptSetColIntvlStats`
- `ttOptSetFlag`
- `ttOptSetOrder`
- `ttOptUpdateStats`
- `ttPLSQLMemoryStats`
ttOptShowJoinOrder

Description
This procedure returns the join order of the last prepared or executed SQL statement (SELECT, UPDATE, DELETE, and INSERT SELECT) in the current transaction. For a join order to be collected, use ttOptSetFlag('ShowJoinOrder', 1) or set the ttIsql ShowJoinOrder command to ON (1) first in the same transaction. AUTOCOMMIT must be off when using either of these commands. The join order is represented by the order of the table names.

Required privilege
This procedure requires no privilege.

Usage in TimesTen Scaleout
TimesTen Scaleout applications can call this built-in procedure.
This procedure executes locally on the element from which it is called.

Related views
This procedure has these related views.
SYS.GV$OPT_JOIN_ORDER
SYS.V$OPT_JOIN_ORDER

Syntax

ttOptShowJoinOrder()

Parameters

ttOptShowJoinOrder has no parameters.

Result set

ttOptShowJoinOrder returns the result:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>joinOrder</td>
<td>TT VARCHAR (4096) NOT NULL</td>
<td>Table names, including owner name quantifiers and correlation name for each table if specified. Table names are returned in parentheses. Using a synonym to specify a table name is not supported.</td>
</tr>
</tbody>
</table>

Examples

Command> AUTOCOMMIT 0;
CALL ttOptSetFlag('ShowJoinOrder', 1);
PREPARE SELECT * FROM t1;
CALL ttOptShowJoinOrder();
{ T1 }
Notes

You must call `ttOptSetFlag('ShowJoinOrder', 1)` or set the `ttIsql ShowJoinOrder` command to `ON (1)` before using this procedure.

This procedure works within one transaction and is not persistent across transactions.

See also

- `ttOptEstimateStats`
- `ttOptGetFlag`
- `ttOptGetOrderBy`
- `ttOptSetColIntvlStats`
- `ttOptSetFlag`
- `ttOptSetOrderBy`
- `ttOptSetTblStats`
- `ttOptUpdateStats`
- `ttPLSQLMemoryStats`
ttOptStatsExport

Description
This procedure returns the set of statements required to restore the table statistics to the current state. If no table is specified, it returns the set of statements required to restore the table statistics for all user tables that the calling user has permission to access.

Required privilege
This procedure requires ADMIN privilege.

Usage in TimesTen Scaleout
TimesTen Grid applications can call this built-in procedure.
This procedure executes locally on the element from which it is called.

Related views
This procedure has these related views.
SYS.GV$OPT_STATS
SYS.V$OPT_STATS

Syntax
ttOptStatsExport('tblName')

Parameters
ttOptStatsExport has the parameter:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tblName</td>
<td>TT_CHAR(61) NOT NULL</td>
<td>Name of the table whose statistics are to be returned. If NULL is passed, then values for all tables are returned. Using a synonym to specify a table name is not supported.</td>
</tr>
</tbody>
</table>

Result set
ttOptStatsExport returns the result set:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stmt</td>
<td>TT_VARCHAR(8300) NOT NULL</td>
<td>The set of statements required to restore the table(s) statistics to the current state.</td>
</tr>
</tbody>
</table>

Examples
CALL ttOptStatsExport('MyTable');
See also

"Create script to regenerate current table statistics" in the Oracle TimesTen In-Memory Database Operations Guide.
ttOptUpdateStats

Description
This procedure updates the statistics for the specified table. TimesTen looks at the data in the table and updates the TBL_STATS and COL_STATS system tables. If the table is large, this process can take some time. Statistics are not computed automatically as rows are updated; an application must compute them explicitly by calling this procedure.

The procedure operates on all tables owned by the current user if tblName is not specified. If the user is the instance administrator, only tables owned by the instance administrator are updated. If the tables are not owned by the user, the user can qualify the table name with their own user name to update stats for the current user.

To determine if your stats are updated, look at the system tables, SYS.COL_STATS and SYS.TBL_STATS, before and after you perform this operation.

Required privilege
This procedure requires no privilege if the user is the table owner, or if tblName is not specified.

This procedure requires the ALTER ANY TABLE privilege if the user is not the table owner.

Usage in TimesTen Scaleout
TimesTen Scaleout applications can call this built-in procedure.
This procedure executes on all elements in the grid.

Related views
This procedure has no related views.

Syntax

ttOptUpdateStats(['tblName'], [invalidate], [option])

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tblName</td>
<td>TT_CHAR(61)</td>
<td>Name of an application table. Can include table owner. If a value of NULL or an empty string is provided, the statistics for all the current user's tables are updated. Using a synonym to specify a table name is not supported.</td>
</tr>
</tbody>
</table>
ttOptUpdateStats

Result set

`ttOptUpdateStats` returns no results.

Examples

CALL `ttOptUpdateStats` ('ACCTS', 1);

Updates the `ACCTS` table and causes all commands that reference the `ACCTS` table to be re-prepared when they are next executed.

CALL `ttOptUpdateStats`('', 1);

Updates all the current user's tables and causes commands on those tables to be reprepared when they are next executed.

CALL `ttOptUpdateStats`('ACCTS', 0, 1);

Forces single interval statistics to be collected.

Notes

If the table name specified is an empty string, statistics are updated for all the current user's tables.

When complete interval statistics are collected, the total number of rows in the table is divided into 20 or less intervals and the distribution of each interval is recorded in the statistics. The new statistics contain the information:

- Number of intervals
- Total number of NULL values in the column
- Total number of NON NULL UNIQUE values in the column
- Total number of rows in the table

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>invalidate</td>
<td>TT_INTEGER</td>
<td>0 (no) or 1 (yes). If <code>invalidate</code> is 1, marks all commands for reprepare on next execution except <code>ALTER TABLE DROP TABLE</code>, and the <code>ALTER TABLE ADD COLUMN FOR SELECT * FROM TABLE</code> statements. These exceptions require manual reprepare. If <code>invalidate</code> is 0, the statistics are not considered to have been modified and existing commands are not reprepared. The <code>invalidate</code> parameter is optional and defaults to 0.</td>
</tr>
<tr>
<td>option</td>
<td>TT_INTEGER</td>
<td>Specifies whether to collect complete interval statistics information. Valid values for this option are: NULL or 0 - Collect complete interval statistics only if a range index exists on the column. If a range index does not exist, only single interval statistics are collected. 1 - Do not collect complete interval statistics. Only single interval statistics are collected. The <code>option</code> parameter is optional and defaults to 0. See the notes below for more information.</td>
</tr>
</tbody>
</table>
Interval information, where each interval contains:

- The minimum value
- The maximum value
- The most frequently occurring value
- The number of times the most frequent value occurred
- The number of rows that have different values than the most frequent value
- The number of unique values besides the most frequent value

Collection of complete interval statistics requires the data to be sorted.

If complete interval statistics are not selected, then statistics are collected by treating the entire distribution as a single interval.

For performance reasons, TimesTen does not hold a lock on tables or rows when computing statistics. However, computing statistics can still slow performance. Estimating statistics generally provides better performance than computing exact statistics. See “ttOptEstimateStats” on page 3-148 for information on estimating statistics.

If you estimate or update statistics with an empty table list, statistics on system tables are updated also, if you have privileges to update the system tables.

See also

- ttOptEstimateStats
- ttOptGetColStats
- ttOptSetColStats
- ttOptSetColIntvlStats
- ttOptSetTblStats
- ttOptUpdateStats
**ttOptUseIndex**

**Description**
This procedure enables applications to alter the generation of execution plans by the TimesTen query optimizer. Applications can call this procedure to disable the use of a set of indexes or enable the consideration of only a set of indexes for each correlation used in a query. Enabling the consideration of an index does not guarantee that the plan generated uses the index. Depending on the estimated cost, the optimizer might choose to use a serialization scan or a materialization scan to access the associated correlation if these scans resulted in a better plan than the ones that use the specified index.

The changes made by this call take effect immediately and affect all subsequent calls to the ODBC functions `SQLPrepare` and `SQLExecDirect` or the JDBC methods `Connection.prepareCall` and `Statement.execute` in the current transaction until the applications explicitly issue a call to clear it. The setting is cleared whenever a new transaction is started.

*AutoCommit* must be set to OFF when running this built-in procedure.

**Required privilege**
This procedure requires no privilege.

**Usage in TimesTen Scaleout**
TimesTen Scaleout applications can call this built-in procedure.

This procedure executes locally on the element from which it is called.

**Related views**
This procedure has no related views.

**Syntax**

```
ttOptUseIndex('IndexName, CorrelationName, 0 | 1 [...}')
```

**Parameters**

`ttOptUseIndex` has a single comma-delimited string parameter, `indOption`, of type `TTVARCHAR(1024)` with these components:

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IndexName</td>
<td>The name of the user-defined index or '_TMPRANGE' for temporary range index or '_TMPHASH' for temporary hash index. If index name is omitted, the setting applies to all indexes of the specified correlation.</td>
</tr>
<tr>
<td>CorrelationName</td>
<td>The correlation name of the table. If a table is defined with a correlation name in the FROM clause, use this correlation name instead of the table name when specifying the index hint for this table. If correlation name is omitted for an entry, the setting affects all tables with the specified index name.</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Result set

`ttOptUseIndex` returns no results.

Examples

CALL `ttOptUseIndex('"3456","1234", t1, 0');`
CALL `ttOptUseIndex('data1.i1, data1.t1, 0');`
CALL `ttOptUseIndex('i1, t1, 0');`

Notes

If `ttOptUseIndex` is called without a parameter or with a NULL value, TimesTen clears the previous index hint.

See also

`ttOptEstimateStats`
`ttOptGetFlag`
`ttOptGetOrder`
`ttOptSetColIntvlStats`
`ttOptSetFlag`
`ttOptSetOrder`
`ttOptSetTblStats`
`ttOptUpdateStats`
`ttPLSQLMemoryStats`
**ttPLSQLMemoryStats**

**Description**

This procedure returns result statistics about PL/SQL library cache performance and activity.

**Required privilege**

This procedure requires no privilege.

**Usage in TimesTen Scaleout**

TimesTen Scaleout applications can call this built-in procedure.

This procedure returns a row for the element from which it was called. To see information about other elements, query the `SYS.GV$PLSQL_MEMORY_STATS` system table.

**Related views**

This procedure has these related views.

- `SYS.GV$PLSQL_MEMORY_STATS`
- `SYS.V$PLSQL_MEMORY_STATS`

**Syntax**

```
ttPLSQLMemoryStats()
```

**Parameters**

`ttPLSQLMemoryStats` takes no parameters.

**Result Set**

`ttPLSQLMemoryStats` returns the results in the following columns:

<table>
<thead>
<tr>
<th>Columns</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>paramName</td>
<td>TT_VARCHAR(30) NOT NULL</td>
<td>The name of the result statistic returned in this row.</td>
</tr>
<tr>
<td>paramValue</td>
<td>BINARY_FLOAT NOT NULL</td>
<td>The value of the result statistic returned in this row.</td>
</tr>
</tbody>
</table>

The following statistics are returned:

- **Gets**: Number of times a lock was requested for a PL/SQL object.
- **GetHits**: Number of times a PL/SQL object’s handle was found in memory.
- **GetHitRatio**: Ratio of GetHits to Gets.
- **Pins**: Number of times a PIN was requested for PL/SQL objects.
- **PinHits**: Number of times all the metadata pieces of the library object were found in memory.
- **PinHitRatio**: Ratio of PinHits to Pins.
- **Reloads**: Any PIN of an object that is not the first PIN performed since the object handle was created, and which requires loading the object from the database.

- **Invalidations**: Total number of times objects in this namespace were marked invalid because a dependent object was modified.

- **CurrentConnectionMemory**: The total amount of heap memory, in MB, allocated to PL/SQL on this database connection.

- **DeferredCleanups**: Total number of times a deferred cleanup occurred.

### Examples

```sql
connect 'DSN=sample';
Connection successful:
DSN=sample;UID=timesten;DataStore=/scratch/timesten/sample;
DatabaseCharacterSet=AL32UTF8;ConnectionCharacterSet=AL32UTF8;
PermSize=128;PLSQL_MEMORY_SIZE=32;
PLSQL_MEMORY_ADDRESS=20000000;PLSQL=1;(Default setting AutoCommit=1)
Command> create procedure hello is begin
dbms_output.put_line('Hello, World!');
end;
  > /
Procedure created.
Command> call ttPlsqlMemoryStats;
< Gets, 485.000000 >
< GetHits, 444.000000 >
< GetHitRatio, .9154639 >
< Pins, 260.000000 >
< PinHits, 178.000000 >
< PinHitRatio, .6846154 >
< Reloads, 4.000000 >
< Invalidations, 0.000000e+00 >
< CurrentConnectionMemory, 56.000000 >
9 rows found.
```
ttRamPolicyAutoReloadGet

Description
This procedure returns the RAM autoreload policy used to determine if a database is reloaded into RAM after an invalidation. The policy can be either autoreload or noautoreload.

Required privilege
This procedure requires no privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax

```
ttRamPolicyAutoReloadGet()
```

Result set

```
ttRamPolicyAutoReloadGet returns the results:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>flag</td>
<td>TT_INTEGER</td>
<td>The policy used to determine if the database is reloaded into RAM after an invalidation. Valid values are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 - The database is not automatically reloaded into memory after an invalidation. This is the equivalent of the command ttAdmin -noAutoReload.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - The database is automatically reloaded into memory after an invalidation. This is the equivalent of the command ttAdmin -autoReload. This is the default autoreload policy.</td>
</tr>
</tbody>
</table>
```

Parameters

```
ttRamPolicyAutoReloadGet has no parameters.
```

Examples

To view the RAM autoreload policy, use:

```
CALL ttRamPolicyAutoReloadGet();
```

See also

```
ttRamPolicyAutoReloadSet
"ttAdmin" on page 5-5
```
ttRamPolicyAutoReloadSet

Description
This procedure determines the RAM autoreload policy if a database is invalidated. The policy can be either autoreload or noautoreload.

Required privilege
This procedure requires the ADMIN privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax
ttRamPolicyAutoReloadSet(flag)

Parameters

ttRamPolicyAutoReloadSet has the parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>flag</td>
<td>TT_INTEGER NOT NULL</td>
<td>The policy used to determine if the database is reloaded into RAM after an invalidation. Valid values are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 - The database is not automatically reloaded into memory after an invalidation. This is the equivalent of the command ttAdmin -noAutoReload.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - The database is automatically reloaded into memory after an invalidation. This is the equivalent of the command ttAdmin -autoReload. This is the default autoreload policy.</td>
</tr>
</tbody>
</table>

Result set

ttRamPolicyAutoReloadSet returns no results.

Examples
To automatically reload a database into RAM after an invalidation, use:

CALL ttRamPolicyAutoReloadSet(1);
ttRamPolicyGet

Description
This procedure returns the RAM policy used to determine when a database is loaded into memory. The policy can be either always, manual, or inUse.

Required privilege
This procedure requires no privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax

```
ttRamPolicyGet()
```

Result set

```
ttRamPolicyGet returns the results:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ramPolicy</td>
<td>TT_VARCHAR (10)</td>
<td>The policy used to determine when the database is loaded into system RAM. Valid values are: always - Specifies that the database should remain in system RAM all the time. manual - Specifies that the database is only to be loaded in system RAM when explicitly loaded by the user, using the <code>ttAdmin -ramLoad</code> command. inUse (default) - Specifies that the database is only loaded in system RAM when in use (when applications are connected). This option cannot be used with temporary databases. TimesTen only allows a temporary database to be loaded into RAM manually. Trying to set the policy generates a warning. This policy is not supported in TimesTen Scaleout.</td>
</tr>
<tr>
<td>ramGrace</td>
<td>TT_INTEGER</td>
<td>If the ramPolicy is inUse, this field reports the number of seconds the database is kept in RAM after the last application has disconnected. Otherwise, this field is NULL.</td>
</tr>
</tbody>
</table>
```

Parameters

```
ttRamPolicyGet has no parameters.
```

Examples

To view the RAM policy, use:

```
CALL ttRamPolicyGet();
```
See also

ttRamPolicySet
"ttAdmin" on page 5-5
“Specifying a RAM policy” in Oracle TimesTen In-Memory Database Operations Guide
ttRamPolicySet

Description
This procedure defines the policy used to determine when a database is loaded into memory. The policy can be either always, manual, or inUse.

Required privilege
This procedure requires the ADMIN privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax
```
ttRamPolicySet('ramPolicy', [ramGrace])
```

Parameters

```
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ramPolicy</td>
<td>TT_VARCHAR (10) NOT NULL</td>
<td>The policy used to determine when the database is loaded into system RAM. Valid values are: always - Specifies that the database should remain in system RAM all the time. manual - Specifies that the database is only to be loaded in system RAM when explicitly loaded by the user, using the <code>ttAdmin -ramLoad</code> command. inUse - Specifies that the database is only loaded in system RAM when in use (when applications are connected). This option cannot be used with temporary databases. TimesTen only allows a temporary database to be loaded into RAM manually. Trying to set the policy generates a warning.</td>
</tr>
<tr>
<td>ramGrace</td>
<td>TT_INTEGER</td>
<td>Sets the number of seconds the database is kept in RAM after the last application has disconnected. This number is only effective if <code>ramPolicy</code> is inUse. This parameter is optional, and when omitted or set to NULL, the existing <code>ramGrace</code> period is left unchanged.</td>
</tr>
</tbody>
</table>
```

Result set
```
ttRamPolicySet returns no results.
```
Examples

To set the policy for loading a database into RAM to be `inUse` and for the database to kept in RAM for 10 seconds after the last application has disconnected, use:

```sql
CALL ttRamPolicySet('inUse', 10);
```

See also

- `ttRamPolicyGet`
- "ttAdmin" on page 5-5
- “Specifying a RAM policy” in Oracle TimesTen In-Memory Database Operations Guide
**ttRedundantIndexCheck**

**Description**

This procedure scans the indicated table (or all the current user's tables) to find redundant indexes. It returns the names of the redundant indexes and a suggestion for which to drop.

**Required privilege**

This procedure requires no privilege.

**Usage in TimesTen Scaleout**

TimesTen Scaleout applications can call this built-in procedure. This procedure executes locally on the element from which it is called.

**Related views**

This procedure has these related views.

- SYS.GV$REDUNDANT_INDEX
- SYS.V$REDUNDANT_INDEX

**Syntax**

```
ttRedundantIndexCheck('tblname')
```

**Parameters**

`ttRedundantIndexCheck` has the parameter:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tblName</td>
<td>TT_CHAR(61)</td>
<td>Name of an application table. Can include table owner. If a value of NULL or an empty string is provided, the redundant indexes for all the current user's tables. Using a synonym to specify a table name is not supported.</td>
</tr>
</tbody>
</table>

**Result Set**

`ttRedundantIndexCheck` returns the result:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>redundancy</td>
<td>TT_VARCHAR (1024) NOT NULL</td>
<td>The names of redundant indexes and a suggestion for which index to drop.</td>
</tr>
</tbody>
</table>

**Examples**

Create table `y` with a primary key. Then create index `i`. TimesTen returns a warning that a redundant index is being created. Create another index, `i1`. The command fails and TimesTen returns an error. Call this procedure to show the warnings.

```
CREATE TABLE y (ID tt_integer primary key);
CREATE INDEX i ON y (id);
```
Warning 2240: New non-unique index I has the same key columns as existing unique index Y; consider dropping index I

CREATE INDEX i1 ON y(id);

2231: New index I1 would be identical to existing index I
The command failed.

CALL ttredundantindexcheck('y');

< Non-unique index SCOTT.Y.I has the same key columns as unique index SCOTT.Y.Y;
consider dropping index SCOTT.Y.I >
1 row found.
ttRepDeactivate

Description
This procedure changes the state of the active database in an active standby pair from ACTIVE to IDLE. Use this procedure when reversing the roles of the master databases in an active standby pair.

Required privilege
This procedure requires the ADMIN privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax
ttRepDeactivate()

Parameters
ttRepDeactivate has no parameters.

Result set
ttRepDeactivate returns no results.

Examples
To deactivate the active database in an active standby pair, use:
CALL ttRepDeactivate();

See also
*ttRepTransmitGet*
*ttRepTransmitSet*
*ttReplicationStatus*
*ttRepPolicySet*
*ttRepStateSave*
*ttRepStateSet*
*ttRepStop*
*ttRepSubscriberStateSet*
*ttRepSubscriberWait*
"ttRepDuplicateEx" in Oracle TimesTen In-Memory Database C Developer’s Guide
ttReplicationStatus

Description
This procedure returns the status of one or more replication peer databases.

Required privilege
This procedure requires no privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax
```
ttReplicationStatus(['subscriber'], ['hostname'])
```

Parameters
```
ttReplicationStatus has the optional parameters:
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>subscriber</td>
<td>TT_VARCHAR(200)</td>
<td>Subscriber of interest or NULL for all subscribers. If the parameter is provided, then it names a replication subscriber about which information is sought. If the parameter is not provided, then information on replication subscribers defined for the current database is returned.</td>
</tr>
<tr>
<td>hostname</td>
<td>TT_VARCHAR(200)</td>
<td>The host name of one or more stores that are configured to receive updates from the executing store; if NULL, then receiving stores are identified by subscriber alone. If both receiver and host name are NULL, then all receiving stores are selected.</td>
</tr>
</tbody>
</table>

Result set
```
ttReplicationStatus returns the result set:
```

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>subscriber</td>
<td>TT_VARCHAR(200) NOT NULL</td>
<td>Subscriber name.</td>
</tr>
<tr>
<td>hostName</td>
<td>TT_VARCHAR(200) NOT NULL</td>
<td>Name of the system that hosts the subscriber.</td>
</tr>
<tr>
<td>port</td>
<td>TT_INTEGER NOT NULL</td>
<td>TCP/IP port used by the subscriber agent to receive updates from the master. A value of 0 indicates replication has automatically assigned the port.</td>
</tr>
</tbody>
</table>
ttReplicationStatus

### Examples

Command>
```sql
CALL ttReplicationStatus();
< MASTER2, HOST1, 0, start     , 1, 257142, _ACTIVESTANDBY, TTREP >
1 row found.

Command>
```sql
CALL ttReplicationStatus('master2', 'host1');
< MASTER2, HOST1, 0, start     , 1, 266439, _ACTIVESTANDBY, TTREP >
1 row found.
```

### Notes

If the `receiver` parameter is not `NULL`, only the status of the given receiver is returned. If the `receiver` parameter is `NULL`, the status of all subscribers is returned.

This procedure is supported only for TimesTen Data Manager ODBC applications. It is not supported for TimesTen Client or JDBC applications.

### See also

- `ttRepDeactivate`
- `ttRepPolicySet`
- `ttRepStop`
- `ttRepSubscriberStateSet`
- `ttRepSyncGet`
- `ttRepSyncSet`
- `ttRepTransmitSet`

"ttRepDuplicateEx" in Oracle TimesTen In-Memory Database C Developer’s Guide
ttRepPolicyGet

Description
This procedure returns the replication restart policy used to determine when the TimesTen for the connected database should run. The policy can be always, manual, or norestart.

Required privilege
This procedure requires no privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax

ttRepPolicyGet()

Parameters

ttRepPolicyGet has no parameters.

Result set

ttRepPolicyGet returns the results:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>repPolicy</td>
<td>TT_VARCHAR (10)</td>
<td>The policy used to determine when the TimesTen replication agent for the database should run. Valid values are: always - Specifies that the replication agent for the database is always running. This option immediately starts the TimesTen replication agent. When the TimesTen daemon restarts, TimesTen automatically restarts the replication agent. manual - Specifies that you must manually start the replication agent using either the ttRepStart built-in procedure or the ttAdmin -repStart command. You must explicitly stop the replication agent using either the ttRepStop built-in procedure or the ttAdmin -repStop command. norestart - Specifies that the replication agent for the database is not to be restarted after a failure.</td>
</tr>
</tbody>
</table>

Examples

To set the policy for TimesTen replication agent to always, use:

CALL ttRepPolicyGet();
See also

- ttRepDeactivate
- ttRepTransmitSet
- ttReplicationStatus
- ttRepPolicySet
- ttRepStart
- ttRepStop
- ttRepSubscriberStateSet
- ttRepSubscriberWait
- ttRepSyncGet
- ttRepSyncSet

"ttRepDuplicateEx" in *Oracle TimesTen In-Memory Database C Developer’s Guide*
ttRepPolicySet

Description
This procedure defines the replication restart policy used to determine when the TimesTen for the connected database should run. The policy can be either always, manual, or norestart.

Required privilege
This procedure requires the ADMIN privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax
```
CALL ttRepPolicySet('repPolicy');
```

Parameters
```
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>repPolicy</td>
<td>TT_VARCHAR</td>
<td>Specifies the policy used to determine when the TimesTen replication agent for the database should run. Valid values are: always - Specifies that the replication agent for the database is always running. This option immediately starts the TimesTen replication agent. When the TimesTen daemon restarts, TimesTen automatically restarts the replication agent. manual - Specifies that you must manually start the using either the ttRepStart built-in procedure or the ttAdmin -repStart command. You must explicitly stop the replication agent using either the ttRepStop built-in procedure or the ttAdmin -repStop command. norestart - Specifies that the replication agent for the database is not to be restarted after a failure.</td>
</tr>
</tbody>
</table>
```

Result set
```
ttRepPolicySet returns no results.
```

Examples
To set the policy for TimesTen replication agent to always, use the following.
```
CALL ttRepPolicySet('always');
```
See also

ttRepDeactivate
ttRepTransmitSet
ttReplicationStatus
ttRepPolicyGet
ttRepStart
ttRepStop
ttRepSubscriberStateSet
ttRepSubscriberWait
ttRepSyncGet
ttRepSyncSet

"ttRepDuplicateEx" in Oracle TimesTen In-Memory Database C Developer’s Guide
ttRepQueryThresholdGet

Description
This procedure returns the number of seconds that was most recently specified as the query threshold for the replication agent. The number of seconds returned may not be the same as the query threshold in effect. Setting a new value for the query threshold takes effect the next time the replication agent is started.

Required privilege
This procedure requires the ADMIN privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax

```
ttRepQueryThresholdGet()
```

Parameters

`ttRepQueryThresholdGet` has no parameters.

Result set

`ttRepQueryThresholdGet` returns the result:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>repQueryThreshold</td>
<td>TT_INTEGER</td>
<td>The number of seconds that a replication query executes before returning an error.</td>
</tr>
</tbody>
</table>

Examples

To get the replication query threshold value, use:

```
CALL ttRepQueryThresholdGet;
< 4 >
1 row found.
```

See also

- `ttRepDeactivate`
- `ttReplicationStatus`
- `ttRepPolicyGet`
- `ttRepQueryThresholdSet`
- `ttRepStart`
- `ttRepStop`
- `ttRepSubscriberStateSet`
- `ttRepSubscriberWait`
- `ttRepSyncGet`
ttRepSyncSet

ttRepTransmitSet

"ttRepDuplicateEx" in Oracle TimesTen In-Memory Database C Developer’s Guide
ttRepQueryThresholdSet

Description
This procedure specifies the number of seconds that a query can be executed by the replication agent before TimesTen writes a warning to the support log. The specified value takes effect the next time the replication agent is started. The query threshold for the replication agent applies to SQL execution on detail tables of materialized views, ON DELETE CASCADE operations and some internal operations that execute SQL statements.

Required privilege
This procedure requires the ADMIN privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax
```sql
ttRepQueryThresholdSet(seconds);
```

Parameters
`ttRepQueryThresholdSet` has the parameter:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>seconds</td>
<td>TT_INTEGER NOT NULL</td>
<td>Number of seconds a SQL statement can be executed by the replication agent before TimesTen writes a warning to the support log. The value must be greater than or equal to 0. Default is 0 and indicates that TimesTen does not write any warnings.</td>
</tr>
</tbody>
</table>

Result set
`ttRepQueryThresholdSet` returns no results.

Examples
To set the replication query threshold value to four seconds, use:
```sql
CALL ttRepQueryThresholdSet(4);
```

See also
- `ttRepDeactivate`
- `ttReplicationStatus`
- `ttRepPolicyGet`
- `ttRepQueryThresholdGet`
- `ttRepStart`
- `ttRepStop`
ttRepQueryThresholdSet

"ttRepDuplicateEx" in Oracle TimesTen In-Memory Database C Developer’s Guide
ttRepStart

Description
This procedure starts the TimesTen replication agent for the connected database.

Required privilege
This procedure requires the \texttt{CACHE\_MANAGER} privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax
\texttt{ttRepStart()}

Parameters
\texttt{ttRepStart} has no parameters.

Result set
\texttt{ttRepStart} returns no results.

Examples
To start the replication agent, use:
\begin{verbatim}
CALL ttRepStart();
\end{verbatim}

Notes
The replication agent does not start if the database does not participate in any replication scheme.

When using this procedure, no application, including the application making the call, can be holding a connection that specifies database-level locking (\texttt{LockLevel}=1).

See also
\begin{itemize}
\item \texttt{ttRepDeactivate}
\item \texttt{ttRepTransmitGet}
\item \texttt{ttRepTransmitSet}
\item \texttt{ttReplicationStatus}
\item \texttt{ttRepPolicySet}
\item \texttt{ttRepStop}
\item \texttt{ttRepSubscriberStateSet}
\item \texttt{ttRepSubscriberWait}
\item \texttt{ttRepSyncSet}
\item \texttt{ttRepSyncGet}
\item "ttRepDuplicateEx" in \textit{Oracle TimesTen In-Memory Database C Developer’s Guide}
\end{itemize}
ttRepStateGet

Description
This procedure returns the current replication state of a database in an active standby pair.

Required privilege
This procedure requires no privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax

ttRepStateGet()

Parameters

ttRepStateGet has no parameters.

Result set

ttRepStateGet returns the result:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>state</td>
<td>TT_VARCHAR (20) NOT NULL</td>
<td>The current replication state of the database. One of: ACTIVE - The database is currently the active master database. Applications may update its replicated tables. STANDBY - The database is the standby master database. Applications may only update its non-replicated tables. FAILED - The database is a failed master database. No updates are replicated to it. IDLE - The database has not yet been assigned its role in the active standby pair. It cannot be updated by applications or replication. Every store comes up in the IDLE state. RECOVERING - The store is in the process of synchronizing updates with the active store after a failure.</td>
</tr>
</tbody>
</table>

Examples
To determine the replication state of the active standby pair, use:

Call ttRepStateGet();
<STANDBY>

Call ttRepStateGet();
<ACTIVE>
Call ttRepStateGet();
<FAILED>

**See also**

- ttRepDeactivate
- ttRepTransmitSet
- ttReplicationStatus
- ttRepPolicySet
- ttRepStateSave
- ttRepStateSet
- ttRepStop
- ttRepSubscriberStateSet
- ttRepSubscriberWait

"ttRepDuplicateEx" in *Oracle TimesTen In-Memory Database C Developer's Guide*
ttRepStateSave

Description
This procedure saves the state of a remote peer database in an active standby pair to the currently connected database. Currently, may only be used to indicate to the active database that the standby database, storeName on hostName, has failed, and that all updates on the active database should be replicated directly to the read-only subscribers.

Required privilege
This procedure requires the ADMIN privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax
`ttRepStateSave('state', 'storeName', 'hostName')`

Parameters
`ttRepStateSave` has these parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>state</code></td>
<td>TT_VARCHAR (20) NOT NULL</td>
<td>The replication state of the indicated database. May only be specified as FAILED in this release. Recording that a standby database has failed indicates that all replicated updates are to be sent directly from the active database to the read-only subscribers.</td>
</tr>
<tr>
<td><code>storeName</code></td>
<td>TT_VARCHAR (200) NOT NULL</td>
<td>Name of the database for which the state is indicated.</td>
</tr>
<tr>
<td><code>hostName</code></td>
<td>TT_VARCHAR (200)</td>
<td>Name of the host where the database resides.</td>
</tr>
</tbody>
</table>

Result set
`ttRepStateSave` returns no results.

Examples
To indicate to the active database that the standby database `standby` on host `backup1` has failed, use:

`ttRepStateSave('FAILED', 'standby', 'backup1');`
See also

ttRepDeactivate
ttRepTransmitSet
ttReplicationStatus
ttRepPolicySet
ttRepStateGet
ttRepStateSet
ttRepStop
ttRepSubscriberStateSet
ttRepSubscriberWait
ttRepSyncGet
ttRepSyncSet
"ttRepDuplicateEx" in Oracle TimesTen In-Memory Database C Developer's Guide
ttRepStateSet

Description
This procedure sets the replication state of a database in an active standby pair replication scheme. Currently, ttRepStateSet may only be used to set the state of a database to ACTIVE, indicating that it is to take the active role in an active standby pair. ttRepStateSet may only be executed in the following situations:

- A database has had a CREATE ACTIVE STANDBY PAIR command executed and no failures have occurred since.
- A database is currently in the STANDBY state, and the other database in the active standby pair has had its state changed from ACTIVE to IDLE using the ttRepDeactivate procedure.
- A database has just recovered from the local transaction log and was in the ACTIVE state before it went down.

Required privilege
This procedure requires the ADMIN privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax

```
ttRepStateSet('state')
```

Parameters

ttRepStateSet has the parameter:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>state</td>
<td>TT_VARCHAR (20) NOT NULL</td>
<td>The replication state of the database. Must be ACTIVE, in this release. Setting a store to ACTIVE designates it as the active database in an active standby pair.</td>
</tr>
</tbody>
</table>

Result set

ttRepStateSet returns no results.

Examples

To set the replication state of the database to ACTIVE, use:

```
CALL ttRepStateSet('ACTIVE');
```

See also

ttRepDeactivate
ttRepTransmitSet
ttReplicationStatus
ttRepPolicySet
ttRepStateGet
ttRepStateSave
ttRepStop
ttRepSubscriberStateSet
ttRepSubscriberWait
ttRepSyncGet
ttRepSyncSet

"ttRepDuplicateEx" in Oracle TimesTen In-Memory Database C Developer’s Guide
ttRepStop

Description
This procedure stops the TimesTen replication agent for the connected database.

Required privilege
This procedure requires the CACHE_MANAGER privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax
```
ttRepStop()
```

Parameters
```
ttRepStop has no parameters.
```

Result set
```
ttRepStop returns no results.
```

Examples
To stop the replication agent, use:
```
CALL ttRepStop();
```

Notes
When using this procedure, no application, including the application making the call, can be holding a connection that specifies database-level locking (LockLevel=1).

See also
```
ttRepDeactivate
ttRepTransmitSet
ttReplicationStatus
ttRepPolicySet
ttRepStart
ttRepSubscriberStateSet
ttRepSubscriberWait
ttRepSyncGet
ttRepSyncSet
"ttRepDuplicateEx" in Oracle TimesTen In-Memory Database C Developer’s Guide
```
ttRepSubscriberStateSet

Description
This procedure changes a replicating subscriber's state with respect to the executing master store.

Required privilege
This procedure requires the ADMIN privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax

```sql
ttRepSubscriberStateSet('replicationName', 'replicationOwner',
'subscriberStoreName', 'subscriberHostName', newStateCode)
```

Parameters

**ttRepSubscriberStateSet** has these parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>replicationName</td>
<td>TT_CHAR (30)</td>
<td>The name of the replication scheme on which to operate. May be NULL to indicate all replication schemes.</td>
</tr>
<tr>
<td>replicationOwner</td>
<td>TT_CHAR (30)</td>
<td>The owner of the replication scheme. May be NULL to indicate all replication scheme owners.</td>
</tr>
<tr>
<td>subscriberStoreName</td>
<td>TT_VARCHAR (200)</td>
<td>The name of the subscribing database whose state is to be set. May be NULL to indicate all stores on host <code>subscriberHostName</code>.</td>
</tr>
<tr>
<td>subscriberHostName</td>
<td>TT_VARCHAR (200)</td>
<td>The subscriber's host. May be NULL to indicate all hosts of subscribing peers.</td>
</tr>
<tr>
<td>newStateCode</td>
<td>TT_INTEGER</td>
<td>An integer code representing the specified subscriber's new state: 0/NULL - Start (default). Starts replication to the subscriber. 1 - Pause. Pauses the replication agent, preserving updates. 2 - Stop. Stops replication to the subscriber, discarding updates. All other state codes are disallowed. (This procedure cannot set a subscriber state to &quot;failed.&quot;) “Set the replication state of subscribers” in the <em>Oracle TimesTen In-Memory Database Replication Guide</em> for more information.</td>
</tr>
</tbody>
</table>

"Set the replication state of subscribers" in the *Oracle TimesTen In-Memory Database Replication Guide* for more information.
Result set

ttRepSubscriberStateSet returns no results.

Examples

For the replication scheme named REPL.REPSCHEDULE, the following directs the master database to set the state of the subscriber database (SUBSCRIBERDS ON SYSTEM1) to Stop (2):

CALL ttRepSubscriberStateSet('REPSCHEDULE', 'REPL', 'SUBSCRIBERDS', 'SYSTEM1', 2);

To direct the master database to set the state of all its subscribers to Pause (1), use:

CALL ttRepSubscriberStateSet( , , , , 1 );

Leaving a parameter empty is equivalent to using NULL.

See also

ttRepDeactivate
ttRepTransmitSet
ttReplicationStatus
ttRepPolicySet
ttRepStart
ttRepStop
ttRepSubscriberWait
ttRepTransmitGet
ttRepTransmitSet
"ttRepDuplicateEx" in Oracle TimesTen In-Memory Database C Developer's Guide
**ttRepSubscriberWait**

**Description**

This procedure causes the caller to wait until all transactions that committed before the call have been transmitted to the subscriber `subscriberStoreName`. It also waits until the subscriber has acknowledged that the updates have been durably committed at the subscriber database.

Call this procedure in a separate transaction, when no other transaction is pending on the active database. This call returns an error if any transactions on the active database are open.

If you set the `waitTime` parameter to -1 and the `subscriberStoreName` parameter to NULL, the `ttRepSubscriberWait` procedure does not return until all updates committed up until the time of the procedure call have been transmitted to all subscribers, and all subscribers have acknowledged that the updates have been durably committed.

The `ttRepSubscriberWait` procedure should not be used when an urgent response is required. Instead, you should use the return receipt service.

Procedure is working expected for transient error scenarios.

---

**Note:** If this procedure is called after all write transaction activity is quiesced at a store (there are no active transactions and no transactions have started), it may take 60 seconds or longer before the subscriber sends the acknowledgment that all updates have been durably committed at the subscriber.

The procedure does not return any failure output (01 value) for permanent error scenarios.

---

**Required privilege**

This procedure requires no privilege.

**Usage in TimesTen Scaleout**

This procedure is not supported in TimesTen Scaleout.

**Related views**

This procedure has no related views.

**Syntax**

```
int ttRepSubscriberWait('replicationName', 'replicationOwner',
    'subscriberStoreName', 'subscriberHostName', waitTime)
```

**Parameters**

`ttRepSubscriberWait` has these parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>replicationName</td>
<td>TT_CHAR (30)</td>
<td>The name of the replication scheme on which to operate. May be NULL to indicate all replication schemes.</td>
</tr>
</tbody>
</table>
ttRepSubscriberWait

**Result Set**

`ttRepSubscriberWait` returns the result set:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>timeOut</code></td>
<td>BINARY(1)</td>
<td>0x00 - The wait succeeded within the allotted <code>waitTime</code>; the specified subscribers are up to date at the time this procedure was called. TimesTen returns 0x01 if not enough time has been granted.</td>
</tr>
</tbody>
</table>

**Examples**

If there is one defined replication scheme `REPOWNER.REPSCHEME`, to direct the transmitting database to wait ten minutes for subscriber `REP2` on `SERVER2` to catch up, use:

```sql
CALL ttRepSubscriberWait('REPSCHEME','REPOWNER', 'REP2', 'SERVER2', 600);
```

**See also**

- `ttRepDeactivate`
- `ttRepTransmitSet`
- `ttReplicationStatus`
- `ttRepPolicySet`
- `ttRepStart`
- `ttRepStop`
- `ttRepSubscriberStateSet`
- `ttRepSyncGet`
- `ttRepSyncSet`
- "ttRepDuplicateEx" in *Oracle TimesTen In-Memory Database C Developer’s Guide*
ttRepSyncGet

Description
This procedure returns static attributes associated with the caller’s use of the replication-based return service. This procedure operates with either the RETURN RECEIPT or RETURN TWOSAFE service.

Required privilege
This procedure requires no privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax
```
ttRepSyncGet();
```

Parameters
`ttRepSyncGet` has no parameters.

Result set
`ttRepSyncGet` returns the result set:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>requestReturn</td>
<td>BINARY(1)</td>
<td>0 (default) - Don’t wait for return notification configured with the RETURN RECEIPT BY REQUEST or RETURN TWOSAFE BY REQUEST option. 1 - Wait for the return notification. Commit resets this attribute to its default value of 0 (“off”).</td>
</tr>
<tr>
<td>returnWait</td>
<td>TT_INTEGER</td>
<td>Specifies the number of seconds to wait for return service acknowledgment. The default value is 10 seconds. A value of '0' means that there is no wait time. This attribute persists across transaction boundaries and applies to all RETURN services independent of the BY REQUEST option.</td>
</tr>
</tbody>
</table>
Examples

To retrieve the caller's requestReturn value, use:

```c
SQLCHAR requestReturn[1];
SQLINTEGER len;
rc = SQLExecDirect ( hstmt , (SQLCHAR *) "CALL ttRepSyncGet( NULL )" , SQL_NTS );
rc = SQLBindCol ( hstmt , /* ColumnNumber */ 1 , /* TargetValuePtr */ requestReturn , /* BufferLength */ sizeof requestReturn , /* StrLen_ */ &len );
rc = SQLFetch( hstmt );
if ( requestReturn[0] ) {
...}
```

Notes

When called within a standalone transaction, `ttRepSyncGet` always returns the default value for `requestReturn`.

Applications can call `ttRepSyncGet` at any point within a transaction in which it is used to request the BY REQUEST return service for that transaction.

If you call `ttRepSyncGet` in a transaction that does not update any RETURN RECEIPT BY REQUEST or RETURN TWOSAFE BY REQUEST replication elements, the call has no external effect.

See also

- `ttRepDeactivate`
- `ttRepTransmitSet`
- `ttReplicationStatus`
- `ttRepPolicySet`
- `ttRepStart`
- `ttRepStop`
- `ttRepSubscriberStateSet`
- `ttRepSubscriberWait`
- `ttRepSyncSet`

"ttRepDuplicateEx" in Oracle TimesTen In-Memory Database C Developer’s Guide
ttRepSyncSet

Description
This procedure sets static attributes associated with the caller’s use of the replication-based return service. This procedure operates with either the RETURN RECEIPT or RETURN TWOSAFE service.

Required privilege
This procedure requires no privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax
```
ttRepSyncSet([requestReturn], [returnWait], [localAction])
```

Parameters
```
ttRepSyncSet has these optional parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>requestReturn</td>
<td>BINARY(1)</td>
<td>0x00 - Turn off the return service for the current transaction.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x01 - Turn on return services for the current transaction. Committing the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>transaction resets this attribute to its default value of 0 (“off”).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>You can use this parameter to turn on or turn off return services only when</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the replication subscribers have been configured with RETURN RECEIPT BY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>REQUEST or RETURN TWOSAFE BY REQUEST.</td>
</tr>
<tr>
<td>returnWait</td>
<td>TT_INTEGER</td>
<td>Specifies the number of seconds to wait for return service acknowledgment.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The default value is 10. A value of 0 means there is no wait time.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This timeout value overrides the value set by the RETURN WAIT TIME attribute</td>
</tr>
<tr>
<td></td>
<td></td>
<td>in the CREATE REPLICATION or ALTER REPLICATION statement.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The timeout set by this parameter persists across transaction boundaries and</td>
</tr>
<tr>
<td></td>
<td></td>
<td>applies to all return services independent of the BY REQUEST option.</td>
</tr>
</tbody>
</table>
```
ttRepSyncSet

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| localAction   | TT_INTEGER| Action to be performed in the event the subscriber cannot acknowledge commit of the transaction within the timeout period specified by returnWait. This parameter can only be used for return twosafe transactions. Set to NULL when using the RETURN service.  
1 (default) - NO ACTION. When a COMMIT times out, it returns the application unblocked, leaving the transaction in the same state it was when the COMMIT began. The application may only reissue the COMMIT.  
2 - COMMIT. When the COMMIT times out, the transaction is committed locally. No more operations are possible on this transaction, and the replicated databases diverge. This attribute persists across transactions and for the life of the connection. |

Result set

ttRepSyncSet has no result set.

Examples

To enable the return receipt service in the current transaction for all the replication elements configured with RETURN RECEIPT BY REQUEST or RETURN TWOSAFE BY REQUEST, use:

```sql
rc = SQLExecDirect ( hstmt,  
(SQLCHAR *)"{CALL ttRepSyncSet( 0x01 )}",  
SQL_NTS )
```

Notes

The call to enable the return receipt service must be part of the transaction (AutoCommit must be off).

See also

- ttRepDeactivate
- ttRepTransmitSet
- ttReplicationStatus
- ttRepPolicySet
- ttRepStart
- ttRepStop
- ttRepSubscriberStateSet
- ttRepSubscriberWait
- ttRepSyncGet
- "ttRepDuplicateEx" in Oracle TimesTen In-Memory Database C Developer’s Guide
ttRepSyncSubscriberStatus

Description

This procedure queries a subscriber database in a replication scheme configured with a return service and a `RETURN DISABLE` failure policy to determine whether return service blocking for the subscriber has been disabled by the failure policy.

The `ttRepSyncSubscriberStatus` procedure returns the failure status of the subscriber database with the specified name on the specified host. You can specify only the `storeName`. However, an error is generated if the replication scheme contains multiple subscribers with the same name on different hosts.

Required privilege

This procedure requires no privilege.

Usage in TimesTen Scaleout

This procedure is not supported in TimesTen Scaleout.

Related views

This procedure has no related views.

Syntax

```
@ttRepSyncSubscriberStatus('subscriber', 'hostName')
```

Parameters

`ttRepSyncSubscriberStatus` has these parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>subscriber</td>
<td>TT_VARCHAR (200) NOT NULL</td>
<td>The name of the subscribing database to be queried.</td>
</tr>
<tr>
<td>hostName</td>
<td>TT_VARCHAR (200)</td>
<td>The host name of one or more stores that are configured to receive updates from the executing store; if NULL, then receiving stores are identified by receiver alone. If both receiver and host name are NULL, then all receiving stores are selected.</td>
</tr>
</tbody>
</table>

Result set

`ttRepSyncSubscriberStatus` returns:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| disabled | TT_INTEGER | Value is either:
| 1 - The return service has been disabled on the subscriber database. |
| 0 - The return service is still enabled on the subscriber database. |
Notes

If the replication scheme specifies `DISABLE RETURN ALL`, then you must use `ttRepSyncSubscriberStatus` to query the status of each individual subscriber in the replication scheme.
ttRepTransmitGet

Description
This procedure returns the status of transmission of updates to subscribers for the current transaction. The corresponding ttRepSyncSet built-in procedure enables you to stop transmission of updates to subscribers for the length of a transaction.

Required privilege
This procedure requires the ADMIN privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax
ttRepTransmitGet()

Parameters
ttRepTransmitGet has no parameters.

Result set
ttRepTransmitGet returns the result:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>transmit</td>
<td>TT_INTEGER</td>
<td>0 - Updates are not being transmitted to any subscribers for the remainder of the transaction on the connection.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 (default) - Updates are being transmitted to subscribers on the connection.</td>
</tr>
</tbody>
</table>

Examples
To return the transmit status on the active database in an active standby pair, use:

CALL ttRepTransmitGet();

See also
ttRepDeactivate
ttReplicationStatus
ttRepPolicySet
ttRepStateSave
ttRepStateSet
ttRepStop
ttRepSubscriberStateSet
ttRepSubscriberWait
ttRepTransmitSet
"ttRepDuplicateEx" in Oracle TimesTen In-Memory Database C Developer’s Guide
ttRepTransmitSet

Description
This procedure stops subsequent updates on the connection it is executed in from being replicated to any subscriber. Use this procedure with care since it could easily lead to transactional inconsistency of remote stores if partial transactions are replicated. If updates are disallowed from getting replicated, the subscriber stores diverge from the master store.

Required privilege
This procedure requires the ADMIN privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax

```sql
ttRepTransmitSet(transmit)
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>transmit</td>
<td>TT_INTEGER NOT NULL</td>
<td>When set to 1, updates are transmitted to subscribers on the connection after the built-in is executed. (This is the default.) When set to 0, updates are not transmitted to any subscribers for the remainder of the transaction in which this call was issued on the connection that issued it.</td>
</tr>
</tbody>
</table>

Result set

`ttRepTransmitSet` returns no results.

Examples

To activate the active database in an active standby pair, use:

```sql
CALL ttRepTransmitSet(1);
```

To deactivate the active database in an active standby pair, use:

```sql
CALL ttRepTransmitSet(0);
```

See also

- `ttRepDeactivate`
- `ttReplicationStatus`
ttRepPolicySet
ttRepStateSave
ttRepStateSet
ttRepStop
ttRepSubscriberStateSet
ttRepSubscriberWait
ttRepTransmitGet

"ttRepDuplicateEx" in Oracle TimesTen In-Memory Database C Developer's Guide
ttRepXactStatus

Description
This procedure checks on the status of a RETURN RECEIPT or RETURN TWOSAFE replication transaction. Using the built-in procedure ttRepXactTokenGet, you can get the token of a RETURN RECEIPT or RETURN TWOSAFE transaction. This is then passed as an input parameter to this built-in procedure. Only a token received from ttRepXactTokenGet may be used. The procedure returns a list of rows each of which have three parameters, a subscriber name, the replication status with respect to the subscriber and an error string that is only returned if a RETURN TWOSAFE replication transaction began but did not complete commit processing.

**Note:** The error parameter is only returned for RETURN TWOSAFE transactions.

Required privilege
This procedure requires no privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax

```sql
ttRepXactStatus(xactID)
```

Parameters

**ttRepXactStatus** has the parameter:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>xactID</td>
<td>VARBINARY (10000)</td>
<td>If no parameter is specified, status is returned for one of the following:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• If called in a transaction that has begun, but not completed, commit processing, it returns the status of the transaction.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• If called at any other time, it returns status for the most recently committed transaction on the connection that was in RETURN RECEIPT or RETURN TWOSAFE mode.</td>
</tr>
</tbody>
</table>

Result set

**ttRepXactStatus** returns the result set:
ttRepXactStatus

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>subscriberName</td>
<td>TT_CHAR (61)</td>
<td>The name of the database that subscribes to tables updated in the transaction. The name returns as: store_name@host_name.</td>
</tr>
<tr>
<td>state</td>
<td>TT_CHAR (2)</td>
<td>The state of the transaction with respect to the subscribing database. The return values are one of the following:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>’NS’ - Transaction not sent to the subscriber.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>’RC’ - Transaction received by the subscriber agent.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>’CT’ - Transaction applied at the subscriber store. (Does not convey whether the transaction ran into an error when being applied.)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>’AP’ - Transaction has been durably applied on the subscriber.</td>
</tr>
<tr>
<td>errorString</td>
<td>TT_VARCHAR (2000)</td>
<td>Error string returned by the subscriber agent describing the error it encountered when applying the twosafe transaction. If no error is encountered, this parameter is NULL. Non-null values are only returned when this procedure is called inside a twosafe replication transaction that has begun, but has not yet completed, processing a commit.</td>
</tr>
</tbody>
</table>

See also

ttRepDeactivate
ttRepTransmitSet
ttReplicationStatus
ttRepPolicySet
ttRepStart
ttRepStop
ttRepSubscriberStateSet
ttRepSubscriberWait
ttRepSyncGet
ttRepSyncSet
ttRepXactTokenGet
"ttRepDuplicateEx" in Oracle TimesTen In-Memory Database C Developer’s Guide
ttRepXactTokenGet

Description

This procedure returns a token for RETURN RECEIPT or RETURN TWOSAFE replication transactions. Depending on the input parameter, type, it returns either:

- A token to the most recently committed RETURN RECEIPT transaction on the connection handle in which it is invoked.
- A token to the most recent transaction on the connection handle in which it is invoked that has begun commit processing on a transaction in RETURN TWOSAFE mode.

This procedure can be executed in any subsequent transaction or in the same transaction after commit processing has begun for a transaction in RETURN TWOSAFE replication.

Required privilege

This procedure requires no privilege.

Usage in TimesTen Scaleout

This procedure is not supported in TimesTen Scaleout.

Related views

This procedure has no related views.

Syntax

```
ttRepXactTokenGet('typ')
```

Parameters

`ttRepXactTokenGet` has these parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>typ</td>
<td>TT_CHAR (2) NOT NULL</td>
<td>The type of transaction desired:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>‘RR’ - Return receipt.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>‘R2’ - Return twosafe.</td>
</tr>
</tbody>
</table>

Result set

`ttRepXactTokenGet` returns the result set:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>token</td>
<td>VARBINARY (10000)</td>
<td>A VARBINARY token used to represent the transaction desired.</td>
</tr>
</tbody>
</table>

See also

- `ttRepDeactivate`
- `ttRepTransmitSet`
- `ttReplicationStatus`
ttRepPolicySet
ttRepStart
ttRepStop
ttRepSubscriberStateSet
ttRepSubscriberWait
ttRepSyncGet
ttRepSyncSet
ttRepXactStatus

"ttRepDuplicateEx" in Oracle TimesTen In-Memory Database C Developer's Guide
ttSetUserColumnID

Description
This procedure explicitly sets the value for the user-specified column ID. Updates presented to the application by the Transaction Log API may contain information about the columns of a table. This column information contains a system-specified column number and a user-specified column identifier. The user-specified column ID has the value 0 until set explicitly by this call.

The system assigns an ID to each column during a CREATE TABLE or ALTER TABLE operation. Setting a user-assigned value for the column ID enables you to have a unique set of column numbers across the entire database or a specific column numbering system for a given table.

Required privilege
This procedure requires the XLA privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax
```
CALL ttSetUserColumnID('tblName', 'colName', repID);
```

Parameters
```
Parameter      Type          Description
------------------------
tblName         TT_CHAR(61) NOT NULL  Table name.
                Using a synonym to specify a table name is not supported.
colName         TT_CHAR(30) NOT NULL  Column name.
repID           TT_INTEGER NOT NULL  Integer identifier.
```

Result set
```
CALL ttSetUserColumnID('APP.SESSION', 'SESSIONID', 15);
```

Examples
```
CALL ttSetUserColumnID('APP.SESSION', 'SESSIONID', 15);
```

See also
```
ttSetUserTableID
```
ttSetUserTableID

Description
This procedure explicitly sets the value of the user table ID. The table that each row is
associated with is expressed with two codes: an application-supplied code called the
user table ID and a system-provided code called the system table ID. Updates are
presented to the application by the Transaction Log API in the form of complete rows.
The user table ID has the value zero until explicitly set with the ttSetUserTableID
procedure.

Required privilege
This procedure requires the XLA privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax

ttSetUserTableID('tblName', repID)

Parameters

ttSetUserTableID has these parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tblName</td>
<td>TT_CHAR (61) NOT NULL</td>
<td>Table name. Using a synonym to specify a table name is not supported.</td>
</tr>
<tr>
<td>repID</td>
<td>BINARY(8) NOT NULL</td>
<td>Integer identifier.</td>
</tr>
</tbody>
</table>

Result set

ttSetUserTableID returns no results.

Examples

CALL ttSetUserTableID('APP.SESSION', 0x123456);

See also

ttSetUserColumnID
**ttSize**

**Description**

This procedure estimates the size of a table or view and the size of indexes. It returns a single row with a single `DOUBLE` column with the estimated number of bytes for the table. The table can be specified as either a table name or a fully qualified table name. A non-NULL `nrows` parameter causes the table size to be estimated assuming the statistics of the current table scaled up to the specified number of rows. If the `nrows` parameter is NULL, the size of the table is estimated with the current number of rows.

The current contents of the table are scanned to determine the average size of each `VARBINARY` and `VARCHAR` column. If the table is empty, the average size of each `VARBINARY` and `VARCHAR` column is estimated to be one-half its declared maximum size. The estimates computed by `ttSize` include storage for the table itself, `VARBINARY` and `VARCHAR` columns and all declared indexes on the table.

The table is scanned when this built-in procedure is called. The scan of the table can be avoided by specifying a non-NULL `frac` value, which should be between 0 and 1. This value estimates the average size of varying-length columns. The maximum size of each varying-length column is multiplied by the `frac` value to compute the estimated average size of `VARBINARY` or `VARCHAR` columns. If the `frac` parameter is not given, the existing rows in the table are scanned and the average length of the varying-length columns in the existing rows is used. If `frac` is omitted and the table has no rows in it, then `frac` is assumed to have the value 0.5.

**Required privilege**

This procedure requires the `SELECT` privilege on the specified table.

**Usage in TimesTen Scaleout**

TimesTen Scaleout applications can call this built-in procedure.

This procedure returns a row for the element from which it was called. To see information about other elements, query the `SYS.GV$TABLE_SIZES` system table.

**Related views**

This procedure has no related views.

**Syntax**

```
ttSize(['tblName'], [nRows], frac)
```

**Parameters**

`ttSize` has these parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>tblName</code></td>
<td><code>TT_CHAR(61)</code></td>
<td>Name of an application table. Can include table owner. This parameter is optional. If not specified all table sizes are returned. Using a synonym to specify a table name is not supported.</td>
</tr>
</tbody>
</table>
Built-In Procedures

**Result set**

`ttSize` returns the following result set.

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>size</code></td>
<td>BINARY_DOUBLE</td>
<td>Estimated size of the table, in bytes.</td>
</tr>
</tbody>
</table>

**Examples**

```
CALL ttSize('ACCTS', 1000000, NULL);
CALL ttSize('ACCTS', 30000, 0.8);
CALL ttSize('SALES.FORECAST', NULL, NULL);
```

When using `ttSize`, you must first execute the command and then fetch the results. For example:

**ODBC**

```c
double size;
SQLLEN len;

rc = SQLExecDirect(hstmt, "call ttSize('SalesData', 250000, 0.75)", SQL_NTS);
rc = SQLBindColumn(hstmt, 1, SQL_C_DOUBLE, &size, sizeof double, &len);
rc = SQLFetch(hstmt);
rc = SQLFreeStmt(hstmt, SQL_CLOSE);
```

**JDBC**

```java
String URL="jdbc:timesten:MyDataStore";
Connection con;
double tblSize=0;

con = DriverManager.getConnection(URL);
CallableStatement cStmt = con.prepareCall("{CALL ttSize('SalesData', 250000, 0.75) }");
if( cStmt.execute() )
{
   rs=cStmt.getResultSet();
   if (rs.next()) {
      tblSize=rs.getDouble(1);
   }
   rs.close();
}
cStmt.close();
con.close();
```
Notes

The \texttt{ttSize} procedure enables you to estimate how large a table will be with its full population of rows based on a small sample. For the best results, populate the table with at least 1,000 typical rows.

See also

\texttt{ttComputeTabSizes}
ttSQLCmdCacheInfo

Description
This procedure returns information about all prepared SQL statements in the TimesTen SQL command cache.

Required privilege
This procedure requires the ADMIN privilege.

Usage in TimesTen Scaleout
TimesTen Scaleout applications can call this built-in procedure.
This procedure executes locally on the element from which it is called.

Related views
This procedure has these related views.
SYS.GV$SQL_CMD_CACHE
SYS.V$SQL_CMD_CACHE

Syntax

```
ttSQLCmdCacheInfo([sqlCmdID])
```

Parameters

`ttSQLCmdCacheInfo` has the optional parameter:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sqlCmdID</td>
<td>TT_BIGINT</td>
<td>The unique identifier of a SQL command in the TimesTen command cache. If no value is supplied, information is displayed for all commands.</td>
</tr>
</tbody>
</table>

Result set

`ttSQLCmdCacheInfo` returns the result set:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sqlCmdID</td>
<td>TT_BIGINT</td>
<td>The unique identifier of a command.</td>
</tr>
<tr>
<td></td>
<td>NOT NULL</td>
<td></td>
</tr>
<tr>
<td>privateCommandConnectionID</td>
<td>TT_INTEGER</td>
<td>If the command is private, this is the connection ID of the connection where it was prepared. If not a private command, this value is -1.</td>
</tr>
<tr>
<td>executions</td>
<td>TT_BIGINT</td>
<td>Counts the number of executions of the command.</td>
</tr>
<tr>
<td></td>
<td>NOT NULL</td>
<td></td>
</tr>
<tr>
<td>prepares</td>
<td>TT_BIGINT</td>
<td>Counts the number of prepares for the command.</td>
</tr>
<tr>
<td></td>
<td>NOT NULL</td>
<td></td>
</tr>
</tbody>
</table>
Example table:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reprepares</td>
<td>TT_BIGINT NOT NULL</td>
<td>Counts the number of reprepares for the command.</td>
</tr>
<tr>
<td>freeable</td>
<td>TT_TINYINT NOT NULL</td>
<td>Indicates whether this command can be garbage collected by the subdaemon.</td>
</tr>
<tr>
<td>size</td>
<td>TT_INTEGER NOT NULL</td>
<td>The total space (bytes) allocated for this command in the command cache.</td>
</tr>
<tr>
<td>owner</td>
<td>TT_CHAR(31) NOT NULL</td>
<td>The user who created the command.</td>
</tr>
<tr>
<td>queryText</td>
<td>TT_VARCHAR (409600) NOT NULL</td>
<td>The full SQL text for the current command.</td>
</tr>
<tr>
<td>fetchCount</td>
<td>TT_BIGINT NOT NULL</td>
<td>The total number of fetch executions done for this statement. The number of fetches depends on TT_PREFETCH_COUNT. The pre-fetch count has a default value of 5 in Read Committed isolation mode and a default of 128 in Serializable mode.</td>
</tr>
<tr>
<td>startTime</td>
<td>TT_TIMESTAMP</td>
<td>The time when the statement was last executed. The value is in the form: YYYY-MM-DD HH:MI:SS.FFF</td>
</tr>
<tr>
<td>maxExecuteTime</td>
<td>NUMBER</td>
<td>The maximum wall clock execute time in seconds for this statement.</td>
</tr>
<tr>
<td>lastExecuteTime</td>
<td>NUMBER</td>
<td>Last measured execution time in seconds of the command.</td>
</tr>
<tr>
<td>minExecuteTime</td>
<td>NUMBER</td>
<td>If SqlCmdSampleFactor &gt; 0, minimum execute time in seconds, otherwise 0.0.</td>
</tr>
<tr>
<td>execloc</td>
<td>TT_TINYINT NOT NULL</td>
<td></td>
</tr>
<tr>
<td>gridCmdId</td>
<td>TT_VARCHAR (64)</td>
<td></td>
</tr>
<tr>
<td>tempSpaceUsage</td>
<td>TT_BIGINT</td>
<td></td>
</tr>
<tr>
<td>maxTempSpaceUsage</td>
<td>TT_BIGINT</td>
<td></td>
</tr>
</tbody>
</table>

Examples

To display command information in ttIsql for all the current valid commands, use:

Command> call ttsqlcmdcacheinfo;
< 43428992, 2048, 5, 5, 0, 1, 2800, SYS , select sys.
objectSequence.nextval from dual >
< 51629120, 2048, 12, 12, 0, 1, 3040, SYS , delete from sys.idl_char$ where obj#:1 and part#:2 >
< 51641192, 2048, 2, 2, 0, 1, 2112, BWAF4EVR , create table tab1 (c1 number not null, c2 number) >
< 43442488, 2048, 5, 5, 0, 1, 4616, SYS , insert into sys.obj$ (owner#, name, namespace, obj#, type#, ctime, mtime, stime, status, flags) values (1, 2, 3, 4, 5, 6, 7, 8, 9, 10) >
< 51632072, 2048, 12, 12, 0, 1, 3040, SYS , delete from sys.idl_ub2$ where obj#:1 and part#:2 >
< 49375216, 2048, 0, 1, 0, 0, 4232, SYS , select 1 fr
To display the information formatted vertically in ttIsql, use:

Command> vertical call ttSQLCmdCacheInfo;
...

To display the information vertically in ttIsql for sqlCmdID 51623232, use:

Command> vertical call ttSQLCmdCacheInfo(51623232);

SQLCMDID:                        51623232
PRIVATE_COMMAND_CONNECTION_ID:   2048
EXECUTIONS:                      4
PREPARES:                        4
REPREPARES:                      0
FREEABLE:                        1
SIZE:                            2704
OWNER:                           SYS
QUERYTEXT:                       delete from sys.source$ where obj#:=
FETCHCOUNT:                      
STARTTIME:                      
MAXEXECUTETIME:                   
LASTEXECUTETIME:                  
MINEXECUTETIME:                   

1 row found.

See also

ttSQLCmdCacheInfo
ttSQLCmdCacheInfoGet
ttSQLCmdCacheInfoGet

Description
This procedure displays information about the commands in the TimesTen SQL command cache.

Required privilege
This procedure requires no privilege.

Usage in TimesTen Scaleout
TimesTen Scaleout applications can call this built-in procedure.
This procedure executes locally on the element from which it is called.

Related views
This procedure has these related views.
SYS.GV$SQL_CMD_CACHE_INFO
SYS.V$SQL_CMD_CACHE_INFO

Syntax
ttSQLCmdCacheInfoGet()

Parameters

Parameters has no parameters.

Result set

**ttSQLCmdCacheInfoGet** returns the result set:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cmdCount</td>
<td>TT_INTEGER NOT NULL</td>
<td>Number of commands in the cache.</td>
</tr>
<tr>
<td>freeableCount</td>
<td>TT_INTEGER NOT NULL</td>
<td>Count of number of freeable commands that can be garbage collected by the subdaemon at that moment. This number is obtained by examining the command information.</td>
</tr>
<tr>
<td>size</td>
<td>TT_BIGINT NOT NULL</td>
<td>The current total space allocated to store all the cached commands, in bytes.</td>
</tr>
</tbody>
</table>

Examples
To display the command count, freeable command count, and total space allocated to the command cache, use:

Command> call ttSQLCmdCacheInfoGet;
< 5,4,12316 >
1 row found
See also

ttSQLCmdCacheInfo
ttSQLCmdQueryPlan

Description
This procedure returns all detailed runtime query plans for SQL statements in the TimesTen SQL command cache. If no argument is supplied, this procedure displays the query plan for all valid commands in the TimesTen cache. For invalid commands, an error is returned that displays the text of the query and the syntax problems.

Required privilege
This procedure requires the ADMIN privilege.

Usage in TimesTen Scaleout
TimesTen Scaleout applications can call this built-in procedure.
This procedure executes locally on the element from which it is called.

Related views
This procedure has these related views.
SYS.GV$SQL_CMD_QUERY_PLAN
SYS.V$SQL_CMD_QUERY_PLAN

Syntax
```
ttSQLCmdQueryPlan([sqlCmdID])
```

Parameters
```
ttSQLCmdQueryPlan has the optional parameter:
```
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sqlCmdID</td>
<td>TT_BIGINT</td>
<td>The unique identifier of a SQL command in the TimesTen command cache. If no value is supplied displays the query plan for all valid commands in the TimesTen cache.</td>
</tr>
</tbody>
</table>

Result set
```
ttSQLCmdQueryPlan returns the result set:
```
<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sqlCmdID</td>
<td>TT_BIGINT NOT NULL</td>
<td>The unique identifier of a command in the TimesTen command cache.</td>
</tr>
<tr>
<td>queryText</td>
<td>TT_VARCHAR(409600)</td>
<td>The first 1024 characters of the SQL text for the current command.</td>
</tr>
<tr>
<td>step</td>
<td>TT_INTEGER</td>
<td>The step number of current operation in this run-time query plan.</td>
</tr>
<tr>
<td>level</td>
<td>TT_INTEGER</td>
<td>The level number of current operation in this run-time query plan.</td>
</tr>
</tbody>
</table>
Examples

To display the query plan for SQLCmdID 528078576:

```
Command> call ttSqlCmdQueryPlan(528078576);
< 528078576, select * from t1 where 1=2 or (x1 in (select x2 from t2, t5 where y2 in (select y3 from t3)) and y1 in (select x4 from t4)), <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL> >
< 528078576, <NULL>, 0, 4, RowLkSerialScan , T1 , TTUSER , , , >
< 528078576, <NULL>, 1, 7, RowLkRangeScan , T2 , TTUSER , I2 , , >
< 528078576, <NULL>, 2, 7, RowLkRangeScan , T5 , TTUSER , I2 , , >
< 528078576, <NULL>, 3, 6, NestedLoop , , , , >
< 528078576, <NULL>, 4, 6, RowLkRangeScan , T3 , TTUSER , I1 , { (Y3=Y2; ) } , , >
< 528078576, <NULL>, 5, 5, NestedLoop , , , , >
< 528078576, <NULL>, 6, 4, Filter , , , , , X1 = X2; >
< 528078576, <NULL>, 7, 3, NestedLoop(Left OuterJoin) , , , , >
< 528078576, <NULL>, 8, 2, Filter , , , , >
```
< 528078576, <NULL>, 9, 2, RowLkRangeScan , T4 , TTUSER , I2 , , Y1 = X4; >
< 528078576, <NULL>, 10, 1, NestedLoop(Left OuterJoin) , , , , , >
< 528078576, <NULL>, 11, 0, Filter , , , , , >
13 rows found.

To display query plans for all valid queries, omit the argument for ttSqlCmdQueryPlan:

< 528079360, select * from t7 where x7 is not null or exists (select 1 from t2,t3 where not 'tuf' like 'abc'), <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL> >
< 528079360, <NULL>, 1, 3, RowLkRangeScan  , T2 , , NOT(LIKE( tuf ,abc ,NULL )) >
< 528079360, <NULL>, 2, 3, RowLkRangeScan  , T3  , TTUSER  , , >
< 528079360, <NULL>, 3, 2, NestedLoop  ,  , , , >
< 528079360, <NULL>, 4, 1, NestedLoop(Left OuterJoin) , , , , , >
< 528079360, <NULL>, 5, 0, Filter , , , , , X7 >
< 527576540, call ttSqlCmdQueryPlan(527973892), <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL> >
< 527576540, <NULL>, 0, 0, Procedure Call , , , , , >
< 528056648, create index i1 on t3(y3), <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL> >
< 528013192, select * from t1 where exists ( select * from t2 where x1=x2) or y1=1, <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL> >
< 528061248, create index i1 on t3(y3), <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL> >
< 528070368, call ttOptSetOrder('t3 t4 t2 t1'), <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL> >
< 528070368, <NULL>, 0, 0, Procedure Call , , , , , >
< 528018856, insert into t2 select * from t1, <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL> >
< 527573452, call ttSqlCmdCacheInfo(527973892), <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL> >
< 527573452, <NULL>, 0, 0, Procedure Call , , , , , >
.... /* more rows here */
The `ttSQLExecutionTimeHistogram` built-in procedure returns a histogram of SQL execution times for either a single SQL command or all SQL commands if command cache sampling is enabled.

**Required privilege**

This procedure requires the `ADMIN` privilege.

**Usage in TimesTen Scaleout**

TimesTen Scaleout applications can call this built-in procedure.

This procedure executes locally on the element from which it is called.

**Related views**

This procedure has these related views.

- `SYS.GV$EXECUTION_TIME_HISTOGRAM`
- `SYS.V$EXECUTION_TIME_HISTOGRAM`

**Syntax**

`ttSQLExecutionTimeHistogram(sqlCmdID)`

**Parameters**

`ttSQLExecutionTimeHistogram` has the optional parameter:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sqlCmdID</td>
<td>TT_BIGINT</td>
<td>The unique identifier of a SQL command in the TimesTen command cache. If no value is supplied displays information about all current commands in the TimesTen command cache.</td>
</tr>
</tbody>
</table>

**Result set**

`ttSQLExecutionTimeHistogram` returns the result set:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>histogramSamples</td>
<td>TT_BIGINT</td>
<td>The number of SQL command execution time operations have been measured since either the database was started or the <code>ttStatsConfig</code> built-in procedure was used to reset the statistics.</td>
</tr>
<tr>
<td>totalExecuteTime</td>
<td>NUMBER</td>
<td>The accumulated wall clock execution time when sampling in seconds.</td>
</tr>
<tr>
<td>bucketUpperBound</td>
<td>NUMBER</td>
<td>The upper limit in seconds of execution time.</td>
</tr>
<tr>
<td>count</td>
<td>TT_BIGINT</td>
<td>The number of SQL commands with time less than or equal to <code>ExecutionTimeLimit</code> and greater than <code>ExecutionTimeLimit</code> from the previous row or 0.</td>
</tr>
</tbody>
</table>
Examples

The following example shows the output for the \texttt{ttSQLExecutionTimeHistogram} built-in procedure:

The following example of the \texttt{ttSQLExecutionTimeHistogram} built-in procedure shows that a total of 1919 statements executed. The total time for all 1919 statements to execute was 1.090751 seconds. This example shows that SQL statements ran in the following time frames:

- 278 statements executed in a time frame that was less than or equal to 0.0001562 seconds.
- 1484 statements executed in a time frame that was greater than 0.0001562 seconds and less than or equal to 0.000125 seconds.
- 35 statements executed in a time frame that was greater than 0.000125 seconds and less than or equal to 0.001 seconds.
- 62 statements executed in a time frame that was greater than 0.001 seconds and less than or equal to 0.008 seconds.
- 60 statements executed in a time frame that was greater than 0.008 seconds and less than or equal to 0.064 seconds.

Command> \texttt{call ttSQLExecutionTimeHistogram;}
< 1919, 1.090751, 0.0001562, 278 >
< 1919, 1.090751, 0.000125, 1484 >
< 1919, 1.090751, 0.001, 35 >
< 1919, 1.090751, 0.008, 62 >
< 1919, 1.090751, 0.064, 60 >
< 1919, 1.090751, 0.512, 0 >
< 1919, 1.090751, 4.096, 0 >
< 1919, 1.090751, 32.768, 0 >
< 1919, 1.090751, 262.144, 0 >
< 1919, 1.090751, 9.999999999E+125, 0 >
10 rows found.

See also

\texttt{ttStatsConfig}
ttStatsConfig

Description
The ttStatsConfig built-in procedure controls statistics collection and parameters for the ttStats utility. This procedure takes a name/value pair as input and outputs a single row result set corresponding to the name/value pair parameters.

Required privilege
This procedure requires the ADMIN privilege.

Usage in TimesTen Scaleout
This procedure is supported in TimesTen Scaleout, but supports different parameter/value pairs than in TimesTen Classic.
This procedure broadcasts changes to all elements.

Related views
This procedure has no related views.

Syntax
```
ttStatsConfig("param", [value], [value])
```

Parameters
ttStatsConfig has the parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>param</td>
<td>VARCHAR2(50)</td>
<td>The name of the parameter to configure.</td>
</tr>
<tr>
<td>value</td>
<td>VARCHAR2(200)</td>
<td>The value of the specified parameter. If no value is supplied, the built-in procedure displays the current value for the specified parameter.</td>
</tr>
<tr>
<td>option</td>
<td>VARCHAR2 (200)</td>
<td></td>
</tr>
</tbody>
</table>

Parameter / Value Pairs
The supported parameter/value pairs in TimesTen Classic and TimesTen Scaleout are different. These are the supported parameter/value pairs:

- **TimesTen**
- **TimesTen Scaleout**

**TimesTen**
These parameter/value pairs can be set with TimesTen:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConnSampleFactor</td>
<td>C,S</td>
<td>The unique identifier of a SQL command in the TimesTen command cache. If you do not supply a value, TimesTen displays the current value of the command.</td>
</tr>
<tr>
<td></td>
<td>0&lt;=C&lt;=Connections</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0&lt;=S&lt;=60000</td>
<td></td>
</tr>
</tbody>
</table>
### LatchStats

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LatchStats</td>
<td>scope, level</td>
<td>Specifies the scope, <code>scope</code>, and the level, <code>level</code>, for collection for latch statistics. The scope value determines at what level TimesTen should collect latch statistics:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <code>conn</code> - Collects latch statistics for your current connection.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <code>db</code> - Collects latch statistics for your database.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <code>con_id</code> - Collects latch statistics for the connection name that you specify.</td>
</tr>
<tr>
<td></td>
<td>level=NONE</td>
<td>TYPICAL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <code>NONE</code> - Disables the collection of latch statistics.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <code>TYPICAL</code> - Ensures the collection of major useful latch statistics.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <code>ALL</code> - Additional statistics are added to the set of statistics collected with the <code>TYPICAL</code> setting. The additional statistics include internal and debugging statistics.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <code>BASIC</code> - Disables the collection of many of the important latch statistics.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If latch statistics are enabled, TimesTen allocates around 100KB from temporary memory to store these statistics.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Once you have configured the <code>LatchStats</code> parameter, you can use the <code>ttLatchStatsGet</code> built-in procedure to view latch statistics. See “ttLatchStatsGet” on page 3-130 for more information.</td>
</tr>
<tr>
<td>SQLCmdHistogramReset</td>
<td>0 or not</td>
<td>The existing SQL execution time statistics are reset if the specified value is nonzero.</td>
</tr>
<tr>
<td>SQLCmdSampleFactor</td>
<td>0 &lt;= value &lt;= 60000</td>
<td>The frequency at which a SQL command sample is taken. The default is 0. A value of 0 indicates that sampling is turned off. A value greater than 0 indicates that a sample is taken at that interval of SQL statements. For example, a value of 10 indicates that for every 10th SQL statement executed, the wall clock time of that execution is captured.</td>
</tr>
</tbody>
</table>
TimesTen Scaleout
These parameter/value pairs can be set with TimesTen Scaleout:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pollSec</td>
<td>0</td>
<td>The polling interval, in seconds, at which the ttStats daemon captures snapshots of the TimesTen Scaleout. A value of 0 disables the ttStats daemon from capturing metrics.</td>
</tr>
<tr>
<td></td>
<td>10 &lt;= value &lt;= 60</td>
<td>The value of the polling interval does not affect the performance of the TimesTen Scaleout. However, a polling interval of 60 seconds tends to use six times more space than a polling interval of 10 seconds. Ensure that you have sufficient PermSize to support the desired polling interval.</td>
</tr>
<tr>
<td>retainMinutes</td>
<td>15 &lt;= value &lt;= 1440</td>
<td>The time, in minutes, that the ttStats daemon waits before aggregating and purging raw metrics. If you use a larger value for retainMinutes, the ttStats daemon stores more metrics in the system tables.</td>
</tr>
<tr>
<td>retentionDays</td>
<td>1 &lt; value &lt; 730</td>
<td>The retention time interval, in days, at which the ttStats daemon drops ttStats snapshots of the TimesTen Scaleout. For example, if the retention time interval is 62 days, the ttStats daemon drops the 1st day's snapshot on the 63rd day. Ensure that you have sufficient PermSize to support the desired retention time interval. In most cases, a day's worth of data takes up around 20 MB of space. These metrics are stored in SYS tables and survive database bounces.</td>
</tr>
</tbody>
</table>

The default value is 30 seconds.

The default value is 120 minutes.

The default value is 62 days.
Result set

`ttStatsConfig` returns the result set:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>param</td>
<td>VARCHAR2(50) NOT NULL</td>
<td>The name of the parameter that was configured.</td>
</tr>
<tr>
<td>value</td>
<td>VARCHAR2(200)</td>
<td>The value of the specified parameter. If no value is supplied, the built-in procedure displays the current value for the specified parameter.</td>
</tr>
</tbody>
</table>

Examples

Since TimesTen and TimesTen Scaleout support different parameter/value pairs, there are also different examples. These are supported examples:

- **TimesTen Classic**
- **TimesTen Scaleout**

**TimesTen Classic**

Sample every command:

```
Command> call ttStatsConfig('SqlCmdSampleFactor',1);
< SQLCMDSAMPLEFACTOR, 1 >
1 row found.
```

Check sampling:

```
Command> call ttStatsConfig('SqlCmdSampleFactor');
< SQLCMDSAMPLEFACTOR, 1 >
1 row found.
```

Sample every fifth statement on connection 1.

```
Command> call ttStatsConfig('ConnSampleFactor', '1,5');
< CONNSAMPLEFACTOR, 1,5 >
1 row found.
```

Turn off sampling on connection 1.

```
Command> call ttStatsConfig('ConnSampleFactor', '1,0');
< CONNSAMPLEFACTOR, 1,0 >
1 row found.
```

Check data store statistics collection level.

```
Command> call ttstatsconfig('StatsLevel');
< STATSLEVEL, TYPICAL >
1 row found.
```

Turn off data store statistics collection.

```
Command> call ttstatsconfig('StatsLevel','None');
< STATSLEVEL, NONE >
1 row found.
```

**TimesTen Scaleout**

Sets the polling interval of statistics to 45 seconds. Therefore, the `ttStats` daemon aggregates statistics every 45 seconds:
Command> call ttStatsConfig('pollsec', 45);
< POLLSEC, 45 >
1 row found.

Sets the time interval when the ttStats daemon purges raw metrics to 60 minutes:

Command> call ttStatsConfig('retainMinutes', 60);
< RETAINMINUTES, 60 >
1 row found.

Sets the retention time interval for statistics to 30 days:

Command> call ttStatsConfig('retentionDays', 30);
< RETENTIONDAYS, 30 >
1 row found.

See also

"ttStats" on page 5-133

ttStatsConfigGet
**ttStatsConfigGet**

**Description**
The `ttStatsConfigGet` built-in procedure returns parameters of the `ttStats` utility that you can set with the `ttStatsConfig` built-in procedure. This procedure does not take any input and outputs a multiple row result set with name/value pair parameters.

**Required privilege**
This procedure requires the `ADMIN` privilege.

**Usage in TimesTen Scaleout**
This procedure is supported in TimesTen Scaleout, but supports different parameter/value pairs than in TimesTen Classic.

This procedure returns a row for the element from which it was called. To see information about other elements, query the `SYS.GV$STATS_CONFIG` system table.

**Related views**
This procedure has these related views.

- `SYS.GV$STATS_CONFIG`
- `SYS.V$STATS_CONFIG`

**Syntax**
```
ttStatsConfigGet()
```

**Parameters**
`ttStatsConfigGet` has no parameters:

**Result set**
`ttStatsConfigGet` returns the result set:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>param</td>
<td>VARCHAR2(50)</td>
<td>The name of the parameter.</td>
</tr>
<tr>
<td>value</td>
<td>VARCHAR2(200)</td>
<td>The current value of the parameter.</td>
</tr>
</tbody>
</table>

**Parameter / Value Pairs**
The supported return parameter/value pairs in TimesTen and TimesTen Scaleout are different. These are the return parameter/value pairs:

- **TimesTen**
- **TimesTen Scaleout**

**TimesTen**
These parameter/value pairs can be returned in the result set in TimesTen:
These parameter/value pairs can be returned in the result set in TimesTen Scaleout:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pollSec</td>
<td>The polling interval, in seconds, at which the ttStats daemon captures snapshots of the TimesTen Scaleout. A value of 0 disables the ttStats daemon from capturing metrics.</td>
</tr>
<tr>
<td>retainMinutes</td>
<td>The time, in minutes, that the ttStats daemon waits before aggregating and purging raw metrics.</td>
</tr>
<tr>
<td>retentionDays</td>
<td>The retention time interval, in days, at which the ttStats daemon drops ttStats snapshots of the TimesTen Scaleout. For example, if the retention time interval is 62 days, the ttStats daemon drops the 1st day's snapshot on the 63rd day.</td>
</tr>
</tbody>
</table>

Examples

Since TimesTen and TimesTen Scaleout support different name/value pair results, there are also different examples. These are supported examples:

- TimesTen Classic
- TimesTen Scaleout

TimesTen

View the configuration settings of ttStatsConfig:

```
Command> call ttStatsConfigGet();
< SQLCMDSAMPLEFACTOR, 1 >
< CONNSAMPLEFACTOR, 2047,0 >;
< STATSLEVEL, TYPICAL >
3 rows found.
```

TimesTen Scaleout

View the configuration settings of ttStatsConfig:
Command> call ttStatsConfigGet();
< POLLSEC, 10 >
< RETAINMINUTES, 120 >
< RETENTIONDAYS, 62 >
3 rows found.

See also
"ttStats" on page 5-133

ttStatsConfig
## ttTableSchemaFromOraQueryGet

### Description
This built-in procedure evaluates a `SELECT` query on a table in an Oracle database and generates a `CREATE TABLE` SQL statement that you can choose to execute. The TimesTen `CREATE TABLE` statement matches the result set column names and types.

This procedure does not create the TimesTen table, it only returns a statement that identifies the table schema.

For more details and usage information, see "Loading data from an Oracle database into a TimesTen table" in the *Oracle TimesTen In-Memory Database Operations Guide*.

### Required privilege
This procedure requires no privileges. The session user must have all required privileges to execute the query on the Oracle database.

### Usage in TimesTen Scaleout
TimesTen Scaleout applications can call this built-in procedure.

This procedure executes locally on the element from which it is called.

### Related views
This procedure has no related views.

### Syntax

```
ttTableSchemaFromOraQueryGet(['tblOwner'], 'tblName', 'selectSQL')
```

### Parameters

`ttTableSchemaFromOraQueryGet` has the parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tblOwner</td>
<td>TT_CHAR (30)</td>
<td>TimesTen table owner (optional). If not provided, the connection ID is used.</td>
</tr>
<tr>
<td>tblName</td>
<td>TT_CHAR (30)</td>
<td><strong>NOT NULL</strong> Table name for the <code>CREATE TABLE</code> statement. The specified TimesTen table cannot be a system table, a synonym, a view, a materialized view or a detail table of a materialized view, a global temporary table or a cache group table.</td>
</tr>
<tr>
<td>selectSQL</td>
<td>TT_VARCHAR (409600)</td>
<td><strong>NOT NULL</strong> A <code>SELECT</code> query on an Oracle database to derive the table column definition. Any expressions in the <code>SELECT</code> list should be provided with a column alias; otherwise, an implementation dependent column name is assumed and the expression is not evaluated.</td>
</tr>
</tbody>
</table>

### Result set

`ttTableSchemaFromOraQueryGet` returns the result set:
This example returns the CREATE TABLE statement to create the TimesTen HR.EMPLOYEES table with all columns found in the Oracle database HR.EMPLOYEES table.

Command> call ttTableSchemaFromOraQueryGet('hr','employees', 'SELECT * FROM hr.employees');
< CREATE TABLE "HR"."EMPLOYEES" ( "EMPLOYEE_ID" number(6,0) NOT NULL, "FIRST_NAME" varchar2(20 byte), "LAST_NAME" varchar2(25 byte) NOT NULL, "EMAIL" varchar2(25 byte) NOT NULL, "PHONE_NUMBER" varchar2(20 byte), "HIRE_DATE" date NOT NULL, "JOB_ID" varchar2(10 byte) NOT NULL, "SALARY" number(8,2), "COMMISSION_PCT" number(2,2), "MANAGER_ID" number(6,0), "DEPARTMENT_ID" number(4,0) ) >
1 row found.

Notes

The query on the Oracle database cannot have any parameter bindings.

TimesTen returns an error if the query cannot be described on the Oracle database, for example, if there is a syntax error.

If an output column type does not have a matching type in TimesTen, TimesTen outputs a warning and the following line for the column definition: >>>>column_name column_type /* reason */

If the query on the Oracle database outputs types not supported by TimesTen, you can add a CAST clause in the SELECT list to explicitly change the output to a TimesTen supported type. Column aliases can be specified for expressions in the SELECT list.

If the query on the Oracle database has LOB output, it is mapped to a VAR type.
ttVersion

Description

The ttVersion built-in procedure returns the five parts of the TimesTen release number.

Required privilege

This procedure requires no privilege.

Usage in TimesTen Scaleout

TimesTen Scaleout applications can call this built-in procedure.

This procedure executes locally on the element from which it is called.

Related views

This procedure has these related views.

SYS.GV$VERSION
SYS.V$VERSION

Syntax

ttVersion()

Parameters

ttVersion has no parameters.

Result set

ttVersion returns the result set:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>major1</td>
<td>TT_INTEGER NOT NULL</td>
<td>The first part of the five-part release number (18 for release 18.1.2.1.0). A change in major1 indicates major infrastructure and functionality changes.</td>
</tr>
<tr>
<td>major2</td>
<td>TT_INTEGER NOT NULL</td>
<td>The second part of the five-part release number (1 for release 18.1.2.1.0). A change in only major2 indicates a version with new functionality changes, but no infrastructure changes.</td>
</tr>
<tr>
<td>patchset</td>
<td>TT_INTEGER NOT NULL</td>
<td>The third part of the five-part release number (2 for release 18.1.2.1.0). A change in only patchset indicates a release that contains all bug fixes since the previous maintenance release.</td>
</tr>
<tr>
<td>patch</td>
<td>TT_INTEGER NOT NULL</td>
<td>The fourth part of the five-part release number (1 for release 18.1.2.1.0). A change in only patch indicates a release with minor bug fixes.</td>
</tr>
</tbody>
</table>
### Examples

Return for release 18.1.2.1.0:

```sql
Command> call ttVersion();
< 18, 1, 2, 1, 0 >
1 row found.
```

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reserved</td>
<td>TT_INTEGER NOT NULL</td>
<td>The fifth part of the five-part release number (0 for release 18.1.2.1.0). Reserved for future use.</td>
</tr>
</tbody>
</table>
**ttWarnOnLowMemory**

**Description**
This procedure enables applications to specify that operations executed on the current connection should return a warning if they allocate memory and find that memory is low. If the value is set, a warning is returned for any operation that does an allocation and finds total memory in use to be above the connection’s threshold value as specified by the `PermWarnThreshold` and `TempWarnThreshold` connection attributes.

**Required privilege**
This procedure requires no privilege.

**Usage in TimesTen Scaleout**
TimesTen Scaleout applications can call this built-in procedure. This procedure executes locally on the element from which it is called.

**Related views**
This procedure has no related views.

**Syntax**

```
ttWarnOnLowMemory(permanent, temporary)
```

**Parameters**

`ttWarnOnLowMemory` has these parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| permanent | TT_INTEGER NOT NULL | 1 - Enable warnings for the permanent data partition  
0 - Disable warnings for the permanent data partition |
| temporary | TT_INTEGER NOT NULL | 1 - Enable warnings for the permanent data partition  
0 - Disable warnings for the permanent data partition |

**Result set**

`ttWarnOnLowMemory` returns no results.

**Examples**

```
CALL ttWarnOnLowMemory(1, 0);
```

Enables low memory warnings for the permanent data partition only.

**Notes**

By default, TimesTen does not issue low memory warnings for either partition. Applications that want to receive these warnings must call this procedure. This
procedure is connection specific, and so you must issue it for each connection upon which warnings are desired. Also, the current setting does not persist to subsequent connections.
ttXactIdGet

Description
This procedure returns transaction ID information for interpreting lock messages. The two result columns of ttXactIdGet are used in combination to uniquely identify a transaction in a database. Taken individually, the columns are not interesting. The result should only be used to correlate with other sources of transaction information. The numbers may not follow a strict pattern.

Required privilege
This procedure requires no privilege.

Usage in TimesTen Scaleout
TimesTen Scaleout applications can call this built-in procedure.
This procedure executes locally on the element from which it is called.

Related views
This procedure has these related views.
SYS.GV$XACT_ID
SYS.V$XACT_ID

Syntax

```
ttXactIdGet()
```

Parameters

ttXactIdGet has no parameters.

Result set

ttXactIdGet returns the result set:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>xactID</td>
<td>TT_INTEGER</td>
<td>Connection ID.</td>
</tr>
<tr>
<td>counter</td>
<td>TT_BIGINT</td>
<td>An increasing number that distinguish successive transactions of the same transaction ID.</td>
</tr>
</tbody>
</table>

Examples

```
Command > automcommit 0;
Command > call ttXactIdGet;
<2,11>
1 row found
Command > commit;
Command > call ttXactIdGet
<3, 12>
1 row found
```
Notes

The output correlates to the values printed in lock error messages and \texttt{ttXactAdmin} lock information output.

See also

\texttt{ttXactAdmin}
"\texttt{txActIdRollback}" in the \textit{Oracle TimesTen In-Memory Database C Developer's Guide}
ttXlaBookmarkCreate

Description
This procedure creates the specified bookmark.

Required privilege
This procedure requires the XLA privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax

ttXlaBookmarkCreate('bookmark', 'replicated')

Parameters

ttXlaBookmarkCreate has the parameter:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bookmark</td>
<td>TT_CHAR (31) NOT NULL</td>
<td>The name of the bookmark to be created.</td>
</tr>
<tr>
<td>replicated</td>
<td>BINARY(1)</td>
<td>0x00 or NULL (equivalent) for non-replicated bookmarks (default setting). 0x01 for replicated bookmarks. If NULL, non-replicated bookmarks are used.</td>
</tr>
</tbody>
</table>

Result set

ttXlaBookmarkCreate returns no results.

Examples

For non-replicated bookmark, execute the following:
Command > call ttXlaBookmarkCreate('mybookmark');

or:
Command> call ttXlabookmarkcreate('mybmk2',0x00);

For a replicated bookmark, execute the following:
Command > call ttXlaBookmarkCreate('mybookmark', 0x01);

For more details on XLA bookmarks, including replicated XLA bookmarks, see "About XLA bookmarks" in the Oracle TimesTen In-Memory Database C Developer’s Guide.
Notes

You can also create a bookmark when you call `ttXlaPersistOpen` function to initialize an XLA handle. See “Creating or reusing a bookmark” in Oracle TimesTen In-Memory Database C Developer’s Guide.

See also

- `ttXlaSubscribe`
- `ttXlaUnsubscribe`
- `ttXlaBookmarkDelete`
ttXlaBookmarkDelete

Description
This procedure deletes the specified bookmark. The bookmark cannot be deleted while it is in use.

Required privilege
This procedure requires the XLA privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax

```
ttXlaBookmarkDelete('bookmark')
```

Parameters

`ttXlaBookmarkDelete` has the parameter:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bookmark</td>
<td>TT_CHAR (31) NOT NULL</td>
<td>The name of the bookmark to be deleted.</td>
</tr>
</tbody>
</table>

Result set

`ttXlaBookmarkDelete` returns no results.

Examples

```
Command > call ttXlaBookmarkDelete('mybookmark');
```

Notes
Before dropping a table that is subscribed to by an XLA bookmark, you must first drop all XLA bookmarks or unsubscribe from XLA tracking.

See also

- `ttXlaBookmarkCreate`
- `ttXlaSubscribe`
- `ttXlaUnsubscribe`
ttXlaSubscribe

Description
This procedure configures persistent XLA tracking of a table. This procedure cannot be executed when the specified bookmark is in use.

Required privilege
This procedure requires the XLA privilege.

Usage in TimesTen Scaleout
This procedure is not supported in TimesTen Scaleout.

Related views
This procedure has no related views.

Syntax
```
ttXlaSubscribe('tblName', 'bookmark')
```

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tblName</td>
<td>TT_CHAR (61) NOT NULL</td>
<td>The name of the table to be tracked. Using a synonym to specify a table name is not supported.</td>
</tr>
<tr>
<td>bookmark</td>
<td>TT_CHAR (31) NOT NULL</td>
<td>The name of the bookmark that the application uses to track this table.</td>
</tr>
</tbody>
</table>

Result set

ttXlaSubscribe returns no results.

Examples

Command > call ttXlaSubscribe ('SALLY.ACCTS', mybookmark);

Notes
Alternatively, the ttXlaTableStatus function subscribes the current bookmark to updates to the specified table, or determines whether the current bookmark is already monitoring DML records associated with the table. See "Specifying which tables to monitor for updates" in Oracle TimesTen In-Memory Database C Developer's Guide

See also

- ttXlaBookmarkCreate
- ttXlaBookmarkDelete
- ttXlaUnsubscribe
**ttXlaUnsubscribe**

**Description**
This procedure stops persistent XLA tracking of a table. This procedure cannot be executed when the specified bookmark is in use.

**Required privilege**
This procedure requires the XLA privilege.

**Usage in TimesTen Scaleout**
This procedure is not supported in TimesTen Scaleout.

**Related views**
This procedure has no related views.

**Syntax**
```sql
ttXlaUnsubscribe('tblName', 'bookmark')
```

**Parameters**
`ttXlaUnsubscribe` has the parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tblName</td>
<td>TT_CHAR (61) NOT NULL</td>
<td>The name of the table on which XLA tracking should be stopped. Using a synonym to specify a table name is not supported.</td>
</tr>
<tr>
<td>bookmark</td>
<td>TT_CHAR (31) NOT NULL</td>
<td>The name of the bookmark that the application uses to track this table.</td>
</tr>
</tbody>
</table>

**Result set**
`ttXlaSubscribe` returns no results.

**Examples**
Command > call ttXlaUnsubscribe ('SALLY.ACCTS', mybookmark);

**Notes**
Before dropping a table that is subscribed to by an XLA bookmark, you must first drop all XLA bookmarks or unsubscribe from XLA tracking.

**See also**
- `ttXlaBookmarkCreate`
- `ttXlaBookmarkDelete`
- `ttXlaSubscribe`
This chapter provides reference information for utilities that are only supported with TimesTen Scaleout, beginning with the following introductory sections:

- Overview
- Required authentication and authorization for utilities
- Utilities list

For information about utilities that are only supported in TimesTen Classic or supported in both TimesTen Classic and TimesTen Scaleout, see Chapter 5, "Utilities".

Overview

The options for TimesTen utilities are generally not case sensitive, except for single character options. You can use `-timeout` or `-TimeOut` interchangeably. However `-v` and `-V` are each unique options.

All utilities return `0` for success and nonzero if an error occurs.

---

**Note:** The utility name and options listed in this chapter are case-insensitive. They appear in mixed case to make the examples and syntax descriptions easier to read.

Required authentication and authorization for utilities

The following sections describe the authentication and authorization required for utilities:

- Required user authentication for utilities
- Required privileges for executing utilities

Required user authentication for utilities

All utilities that require a password prompt for one.

If TimesTen prompts for a password, input is not displayed on the command line.

Generally, when no user is specified, the user is assumed to be the user name identified by the operating system, and TimesTen does not prompt for a password.

When a utility accepts a DSN, connection string or database path as a parameter, specify the value at the end of the command line.
Required privileges for executing utilities

Certain TimesTen command-line utilities require privileges. Each utility in this chapter describes the privilege required for execution. You may receive a database not loaded error if you try to execute any utility with a user other than the instance administrator and the database is not loaded into memory. In this case, TimesTen cannot determine the privileges of the user.

Thus any utilities requiring privileges have to be run either as the instance administrator or executed while the database is loaded.

Utilities list

Utilities listed in Table 4–1 are described in this chapter.

Utilities listed in Table 4–2 are described in Chapter 5, "Utilities”.

Table 4–1  Utilities supported only in TimesTen Scaleout descriptions

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ttGridAdmin</td>
<td>Administers a TimesTen Scaleout grid.</td>
</tr>
<tr>
<td>ttGridRollout</td>
<td>Creates a new grid and database.</td>
</tr>
</tbody>
</table>

Table 4–2  Utilities supported in both TimesTen Scaleout and TimesTen Classic descriptions

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ttInstallationCheck</td>
<td>Examines all files in an installation of TimesTen and generates a signature for the installation.</td>
</tr>
<tr>
<td>ttInstallDSN</td>
<td>Generates a Windows client DSN for one or more entries listed in the provided input file and installs them into the ODBC Panel as a System DSN.</td>
</tr>
<tr>
<td>ttInstanceCreate</td>
<td>Create a new TimesTen instance.</td>
</tr>
<tr>
<td>ttInstanceDestroy</td>
<td>Destroys an existing TimesTen instance.</td>
</tr>
<tr>
<td>ttInstanceModify</td>
<td>Modifies certain attributes of an instance.</td>
</tr>
<tr>
<td>ttIsql</td>
<td>Executes SQL statements interactively.</td>
</tr>
<tr>
<td>ttMigrate</td>
<td>Saves and restores TimesTen objects.</td>
</tr>
<tr>
<td>ttSchema</td>
<td>Prints out the schema, or selected objects, of a database.</td>
</tr>
<tr>
<td>ttSize</td>
<td>Estimates the amount of space that a given table, including any views in the database will consume when the table grows to include a specified number of rows.</td>
</tr>
<tr>
<td>ttStats</td>
<td>Monitors database metrics or takes and compares snapshots of metrics.</td>
</tr>
<tr>
<td>ttSyslogCheck (UNIX/Linux)</td>
<td>Determines if the system's /etc/syslog.conf file is properly configured for TimesTen.</td>
</tr>
<tr>
<td>ttVersion</td>
<td>lists the TimesTen release information.</td>
</tr>
</tbody>
</table>

Note: For security reasons, we do not recommend setting a value for PWD on the command line.
ttGridAdmin

Description

Use the ttGridAdmin utility for all aspects of administering a grid, such as creating a grid, adding or removing data instances or management instances, creating databases, and redistributing data to new data instances. The grid and database configuration resulting from these operations is stored in the grid model, which is distributed to instances of the grid. (See “Grid model” on page 4-4 for an introduction to the model.)

**Important:** In TimesTen Scaleout, do not update configuration files manually (such as timesten.conf, sys.odbc.ini, and tnsnames.ora).

Required privilege

Instance administrator of the management instance from which ttGridAdmin is run. The user then becomes the instance administrator of all instances created with ttGridAdmin.

Usage with TimesTen Scaleout

This utility is specifically for use with TimesTen Scaleout, with commands that perform any operations on a grid.

Syntax

For general syntax (help options and options that apply to all commands), see "Help and general options" on page 4-10. For syntax for individual commands, see the relevant sections under "ttGridAdmin operations" on page 4-9.
Grid model

TimesTen Scaleout maintains a central configuration of the grid within a model that describes the desired structure of the grid. The model represents the desired logical topology of the grid and contains objects that represent components of the grid, such as installations, hosts, instances, and database definitions. Each time you use ttGridAdmin to add a grid component, a corresponding object is added to the model.

When you use ttGridAdmin to apply the model, TimesTen Scaleout attempts to implement it into the operational grid, such as by creating the desired physical installations and instances.
Grid objects and object naming

Each entity in a grid—such as each host, instance, installation, physical group, and repository—is defined as a named object.

Each object type, representing these types of entities, has its own namespace. You can have a host named \textit{xyz} and a physical group named \textit{xyz} without conflict. In addition, instance namespaces and installation namespaces are per host. You can have an instance named \textit{instance1} on host1 and an instance named \textit{instance1} on host2 without conflict.

Object-naming hierarchies such as this can be expressed in \textit{ttGridAdmin} syntax using fully qualified names, \texttt{toplevelobject.nextlevelobject}. For example, host1.instance1 and host2.instance1. To specify an instance or installation in \textit{ttGridAdmin} syntax, you need only specify \texttt{hostname} (instead of \texttt{hostnameinstancename} or \texttt{hostname.installname}) if there is only one instance or installation (as applicable) on the host.

Operations on grid entities through \textit{ttGridAdmin}, such as creating or removing an instance or installation, are managed through the corresponding objects in the model. The physical entities themselves are not created or removed until the model is applied through the \textit{ttGridAdmin modelApply} command. For example, to remove an instance named \texttt{host1.instance1}, the \textit{instanceDelete} command removes the object named \texttt{host1.instance1} from the model, then the \textit{modelApply} command removes the physical instance (the instance home directory and everything under it).

Be aware of these limitations in object naming in the grid model:

- Names must use only ASCII characters.
- Allowed characters are alphabetic, numeric, hyphen (-), and underscore (_).
- Database definition names and connectable names are limited to 32 characters (due to ODBC limitations).
- All other object names are limited to 256 characters.
- Object names are case-insensitive (so you cannot have an instance \texttt{instance} and an instance \texttt{Instance} on the same host), but are represented and shown as specified in \textit{ttGridAdmin} commands. If you specify \texttt{MyInstance} in the \textit{instanceCreate} command, that is how it will be shown.
- You cannot name anything \texttt{All} or \texttt{Default}, which are reserved names.
Address formats

Some ttGridAdmin commands, such as gridCreate and hostCreate, have options to specify the address or addresses used for internal and external communications. You can specify addresses as DNS names or IP addresses, although use of DNS names is more typical. IP addresses can be in either IPv6 or IPv4 format. For example:

- **DNS name**: myhost.example.com
- **IPv6 address**: 2606:b400:2000:834:26:3eff:fe07:5b83
- **IPv4 address**: 92.68.0.1
Database management operations

In a Grid environment, database management operations—dbCreate/dbDestroy, dbLoad/dbUnload, dbOpen/dbClose—are performed asynchronously by default. This is generally advisable, as such operations are not atomic and may take a long time. In a large grid, loading a database may succeed immediately on many hosts, take a little longer on others, and much longer on others. Some hosts may, in fact, be down when the operation was executed, so cannot perform the operation until they are back up.

By default, commands for these database operations return without waiting for completion, but they can optionally wait, with or without a timeout. With wait and a timeout, a command does not return until it has completed on all instances or reaches the timeout. With wait and no timeout, a command will never return if any instances are down. There are advantages and disadvantages to each approach, depending on factors such as how large the grid is. For a large grid, you may choose to proceed before the operation has completed on all instances, while on a small grid it may be more sensible to wait until it has completed on every instance.

TimesTen Scaleout tracks the state of a database, including each element of the database, and it is up to the user to check status of an operation (through the dbStatus command, optionally using the -all option for further details) to see how many instances have completed the operation. In particular, after loading the database, you can use the status information to determine if it has been loaded on enough instances for the database to be opened and users to start accessing it.

No command is provided to cancel an operation on any or all instances.
Command timeouts and waits

Note that as ttGridAdmin executes a command, it may run operating system commands as well. Using the top-level -timeout option, you can specify that ttGridAdmin will wait for the specified number of seconds for such operations to complete. If an operation does not complete within the specified number of seconds, the ttGridAdmin command being executed fails.

In addition, the ttGridAdmin database management commands dbCreate, dbDestroy, dbload, dbUnload, dbOpen and dbClose have a -nowait/-wait option. Each of these commands initiates a state change that is recorded in the active management instance of the grid. With a setting of -nowait (the default), the commands return immediately without waiting for the state change to complete. If -wait with a timeout value \( n \) is specified, ttGridAdmin will wait for up to \( n \) seconds for the state change to complete. If -wait is specified without a timeout, ttGridAdmin will wait without limit for the state change to complete.
ttGridAdmin operations

The listed sections provide detailed information about ttGridAdmin commands and functionality in these areas:

- Help and general options
- Return values
- Backup and restore operations
- Connectable operations
- Data space group operations
- Database definition operations
- Database operations
- Grid operations
- Host operations
- Import and export operations
- Installation operations
- Instance operations
- Management instance operations
- Membership operations
- Model operations
- Oracle Database operations
- Physical group operations
- Repository operations
Help and general options

These options work with any `ttGridAdmin` command or, for help, at the top level without any command.

```
ttGridAdmin [-h | -help | -?] [command]
ttGridAdmin [-o json] [-timeout n] command [-command_option] ...
```

Options

`ttGridAdmin` has the general options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h</td>
<td>Display help information.</td>
</tr>
<tr>
<td>-help</td>
<td>If specified by itself, this option categorizes and lists <code>ttGridAdmin</code> commands. For example: ttGridAdmin -h</td>
</tr>
<tr>
<td>-?</td>
<td>if specified with a command, this option displays syntax and option descriptions for the given command. For example: ttGridAdmin -h dbCreate</td>
</tr>
<tr>
<td>-o json</td>
<td>Choose JSON output.</td>
</tr>
<tr>
<td></td>
<td>Output for the command will be in JSON format. (Otherwise, output is in human-readable format.) Important: There is no guarantee of JSON output compatibility between TimesTen releases.</td>
</tr>
<tr>
<td>-timeout n</td>
<td>Maximum number of seconds to wait for a long-running operation to complete. The default is 600. Note that as <code>ttGridAdmin</code> executes a command, it may run operating system commands as well. It will wait for the specified number of seconds for such operations to complete. If an operation does not complete within the specified number of seconds, the <code>ttGridAdmin</code> command being executed fails.</td>
</tr>
</tbody>
</table>
Return values

This section describes error or status values and JSON data elements returned by ttGridAdmin commands.

Error and status return codes

ttGridAdmin commands returns error codes to note success or failure, including these general codes:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Success</td>
</tr>
<tr>
<td>255</td>
<td>Internal error</td>
</tr>
<tr>
<td>254</td>
<td>Syntax error</td>
</tr>
</tbody>
</table>

JSON data elements returned

When JSON output is specified, the stdout of the command includes at least these name/value pairs. (Refer to [http://www.json.org/](http://www.json.org/) for general information about JSON output.)

<table>
<thead>
<tr>
<th>Return</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;status&quot;</td>
<td>number</td>
<td>Return code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>See the preceding section, &quot;Error and status return codes&quot;.</td>
</tr>
<tr>
<td>&quot;errno&quot;</td>
<td>number</td>
<td>Error number, if an error occurred</td>
</tr>
<tr>
<td>&quot;errmsg&quot;</td>
<td>string</td>
<td>Error message, if an error occurred</td>
</tr>
</tbody>
</table>

Note: Additional, command-specific JSON data elements may also be returned.
Backup and restore operations

Use ttGridAdmin commands in this section to back up and restore databases, display the status of those operations, or delete a backup.

Also see "Migrating, Backing Up and Restoring Data" in Oracle TimesTen In-Memory Database Scaleout User’s Guide.

Back up a database (dbBackup)

The dbBackup command initiates a backup of the specified database.

```
ttGridAdmin dbBackup dbname
   -repository reponame
   [-name backupname]
```

In some cases you must use dbExport instead. This would be the case, for example, if the grid topology at the restore location has fewer replica sets than the backed up database, or the restore location is running a version of TimesTen that is not patch-compatible with the version of the backed up database. See "Migrating, Backing Up and Restoring Data" in Oracle TimesTen In-Memory Database Scaleout User’s Guide for additional information.

Important: Be aware of the following if the specified repository was created with -method scp:

When an element is backed up, the backup file is stored on the local disk where the element is located before being copied to the remote repository.

A backup is stored as a collection under a repository. You first must create the repository. See "Repository operations" on page 4-108.

Options

The dbBackup command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbname</td>
<td>Name of the database to back up.</td>
</tr>
<tr>
<td>-repository reponame</td>
<td>Name of the repository where the backup will be located.</td>
</tr>
<tr>
<td>-name backupname</td>
<td>Specifies a name for the backup. The default is the letter &quot;B&quot; followed by the date and time of the backup, in the format: Byyyymmddhhmmss</td>
</tr>
</tbody>
</table>

Examples

This example backs up database1 into repository repol. It uses the default name for the backup, according to the current timestamp (2/22/17 at 14:55:44).

```
% ttGridAdmin dbBackup database1 -repository repol
dbBackup B20170222145544 started
```
You can then use `dbBackupStatus` to check progress, as shown in the example in "Display the status of a database backup (dbBackupStatus)" on page 4-13. The backup is finished when each element and the database as a whole are indicated as complete.

**Notes**

- The backup is performed asynchronously. Use the `dbBackupStatus` command to check progress.
- One element from each replica set is backed up.
- Each replica set is stored as a sub-collection.
- For disk space requirements, see "Backing up and restoring a TimesTen Scaleout database" in Oracle TimesTen In-Memory Database Scaleout User's Guide.

### Delete a database backup (dbBackupDelete)

The `dbBackupDelete` command deletes the specified database backup.

```
ttGridAdmin dbBackupDelete -repository reponame -name backupname
```

**Options**

The `dbBackupDelete` command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-repository reponame</code></td>
<td>Name of the repository where the backup is located.</td>
</tr>
<tr>
<td><code>-name backupname</code></td>
<td>Name of the backup to delete.</td>
</tr>
</tbody>
</table>

**Examples**

This example deletes the backup created in the example in "Back up a database (dbBackup)" on page 4-12.

```
% ttGridAdmin dbBackupDelete -repository repol -name B20170222145544
Backup B20170222145544 deleted
```

**Notes**

This command is typically used to delete old or failed backups.

### Display the status of a database backup (dbBackupStatus)

The `dbBackupStatus` command shows the status of a database backup or backups previously started.

```
ttGridAdmin dbBackupStatus dbname
                      [-name backupname]
```

**Options**

The `dbBackupStatus` command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>dbname</code></td>
<td>Name of the database being backed up.</td>
</tr>
</tbody>
</table>
Examples

This example shows status upon completion of the backup from the example in "Back up a database (dbBackup)" on page 4-12.

% ttGridAdmin dbBackupStatus database1 -name B20170222145544

Database  Backup          Repository Host  Instance  Elem State     Started                Finished
--------- --------------- ---------  ----- --------- ---- --------- ---------------------- --------
database1 B20170222145544 repo1                           Completed 2017-02-22T14:55:44.000Z Y
host3 instance1    1 Complete
host4 instance1    2 Complete
host5 instance1    3 Complete

Notes

- When you believe the backup is complete, confirm that `dbBackupStatus` shows that the backup as a whole and for each instance is shown as complete. If there were any failures, see "Check the status of a backup" in Oracle TimesTen In-Memory Database Scaleout User’s Guide.

- Y in the Finished column indicates that the command has finished executing, regardless of state—that each instance has succeeded or failed.

- The metadata associated with a database backup is deleted when the database is deleted. After a database is deleted, you can use the `repositoryList` command to see existing backups.

Restore a database (dbRestore)

The `dbRestore` command restores a database backup into a new database.

`ttGridAdmin dbRestore dbname`  
`-repository reponame`  
`-backup backupname`

Options

The `dbRestore` command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>dbname</code></td>
<td>Name of the database to be created, then restored from the backup.</td>
</tr>
<tr>
<td><code>-repository reponame</code></td>
<td>Name of the repository where the backup is located.</td>
</tr>
<tr>
<td><code>-backup backupname</code></td>
<td>Name of the backup to use for the restore.</td>
</tr>
</tbody>
</table>

Examples

This example creates and restores a database `res_db1` from a backup `mybkup`.

% ttGridAdmin dbRestore res_db1 -repository repol -backup mybkup

dbRestore mybkup started

You can then use `dbRestoreStatus` to check progress, as shown in the example in "Display the status of a database restore (dbRestoreStatus)" on page 4-15. The restore is finished when each element and the database as a whole are indicated as complete.
Notes

- This database must already be defined (with dbdefCreate) but not yet created.
- The restore is performed asynchronously. Use the dbRestoreStatus command to check progress.
- The restored database is loaded into memory when dbRestore completes, but not opened.
- You can restore to the original database definition or to a newly created database definition.
- You cannot restore to a database with fewer replica sets than what was backed up. (If the number of data instances on hosts in each data space group is not sufficient to support the number of replica sets in the database that was backed up, you must use dbExport and dbImport instead.)
- If you restore to a database with more replica sets than what was backed up, only the number of replica sets that were backed up will be added to the database distribution map. For example, if you back up a database with two replica sets and restore to a database with four replica sets, only the elements in two replica sets will be added to the distribution map. You would then have to redistribute data with dbDistribute to get four replica sets.
- For disk space requirements, see “Backing up and restoring a TimesTen Scaleout database” in Oracle TimesTen In-Memory Database Scaleout User’s Guide.

Display the status of a database restore (dbRestoreStatus)

The dbRestoreStatus command shows the status of a database restore previously started.

```
ttGridAdmin dbRestoreStatus dbname
```

Options

The dbRestoreStatus command has the option:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbname</td>
<td>Name of the database where the restore is being checked.</td>
</tr>
</tbody>
</table>

Examples

This example shows status upon completion of the restore from the example in "Restore a database (dbRestore)" on page 4-14.

```
% ttGridAdmin dbRestoreStatus res_db
Database Restore Repository Host Instance Elem State Started           Finished
-------- ------- --------- ------ ------- ------------------------------------------------ -
res_db   mybkup  repol     res_db1 Restore_Finale_Complete 2017-03-03T13:19:39.000Z Y
host3    instance1  Restore_Instance_Complete host4    instance1  Restore_Instance_Complete
host5    instance1  Restore_Finale_Complete
```

Notes

- When you believe the restore is complete, confirm that dbRestoreStatus shows Restore_Finale_Complete for the restore as a whole and Restore_Instance_Complete or Restore_Finale_Complete for each instance. If there were any
failures, see "Check the status of a restore" in Oracle TimesTen In-Memory Database Scaleout User’s Guide.

- Y in the Finished column indicates that the command has finished executing, regardless of state—that each instance has succeeded or failed.
Connectable operations

Use `ttGridAdmin` commands in this section to create, delete, modify, export, or list connectable objects, used in connecting to a TimesTen Scaleout database. A connectable specifies a set of connection attribute settings and thereby defines an underlying DSN and `tnsnames.ora` file entry.

There are two types of connectables: direct connectables for direct mode access, and client/server connectables for client/server access.

You can have multiple connectables for a database to use different sets of connection attribute settings. For example, if you have one application designed to receive ASCII data from the database and another designed to receive Unicode data, then create two connectables, each with the appropriate `ConnectionCharacterSet` attribute setting.

---

**Note:** A direct connectable, with the same name as the database, is created automatically when you define a database with the `dbdefCreate` command.

---

See "Connecting to a database" and its subsections in *Oracle TimesTen In-Memory Database Scaleout User’s Guide* for additional information about connectables.

Create a connectable (connectableCreate)

The `connectableCreate` command creates a connectable object in the model, defining connection attribute settings.

```
`ttGridAdmin connectableCreate -dbdef name`  
  [-cs [-only hostname [.instancename]]]
  filepath
```

Options

The `connectableCreate` command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-dbdef name</code></td>
<td>The name of the database definition object that the connectable references (which is the name of the database to connect to).</td>
</tr>
<tr>
<td><code>-cs</code></td>
<td>Specifies that the connectable will be for client/server access to the database. If this is not specified, the connectable will be for direct mode access.</td>
</tr>
<tr>
<td><code>-only hostname [.instancename]</code></td>
<td>For client/server connections, optionally specifies that applications connecting through this connectable should connect to the indicated instance. By default, all data instances in the grid are available and connections are distributed among them. You need to specify the instance name only if there is more than one instance on the host. (See &quot;Grid objects and object naming&quot; on page 4-5.) You can specify multiple <code>-only</code> options, in which case client/server applications can connect to any instance listed in the <code>-only</code> settings.</td>
</tr>
</tbody>
</table>
Examples

The example creates a client/server connectable using this connectable file, database1client.connect:

ConnectionCharacterSet=AL32UTF8
UID=ttclient

Create the connectable:

% ttGridAdmin connectableCreate -dbdef database1 -cs /sw/tten/grid/conndefs/database1client.connect
Connectable database1client created.

Notes

- The connectable file must be in odbc.ini format, as shown in the example, with attribute=value on each line. A DSN name, in [name] format such as [database1client] here, is optional. If provided, it must match the connectable name determined by the connectable file name.
- The default value is used for any connection attribute not set in the connectable.
- When you apply the model after creating a connectable, new versions of all necessary configuration files are written to each data instance. This makes the connectable available for use. (Never edit configuration files manually. They are overwritten each time the model is applied.)
- You cannot set data store (creation time) attributes or first connection attributes in the connectable file. Those must be set in the database definition file for the dbdefCreate or dbdefModify command.
- Any settings for TTC_SERVER, TTC_SERVER_DSN, or TCP_PORT are ignored. Appropriate values for the grid topology are automatically used.
- The connectable that is defined is usable through all APIs supported by TimesTen.

Delete a connectable (connectableDelete)

The connectableDelete command deletes an existing connectable object from the model.

`ttGridAdmin connectableDelete name`

Options

The connectableDelete command has the option:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>The name of the connectable object to delete.</td>
</tr>
</tbody>
</table>
Examples

% ttGridAdmin connectableDelete database1client
Connectable database1client deleted from Model.

Notes

When you apply the model after deleting a connectable, new versions of all necessary configuration files are written to each data instance, with the connectable entry removed. (Never edit configuration files manually. They are overwritten each time the model is applied.)

Export a connectable (connectableExport)

The `connectableExport` command exports a connectable object and its connection attribute settings, typically to a specified file.

```
ttGridAdmin connectableExport name [filepath]
```

Options

The `connectableExport` command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Name of the connectable object to export.</td>
</tr>
<tr>
<td>filepath</td>
<td>Path and name of the connectable file to create, typically a .connect file for use with <code>connectableCreate</code> or <code>connectableModify</code>. If no file is specified, the export goes to stdout. <strong>Important:</strong> If you specify an existing file, it will be overwritten.</td>
</tr>
</tbody>
</table>

Examples

This example exports the connectable created in the `connectableCreate` example to the file `database1client.connect`.

```
% ttGridAdmin connectableExport database1client
/sw/tten/grid/condefs/database1client.connect
Connectable database1client exported

Resulting contents of database1client.connect:

# Connectable GUID 3210288C-DF44-447D-ADB6-BDC8F7CFE17C Exported 2017-11-14 17:53:25
[database1client]
ConnectionCharacterSet=AL32UTF8
UID=ttclient
```

Notes

A typical use case is if you want to modify a connectable, but the original connectable file is no longer available.

List connectables (connectableList)

The `connectableList` command lists the connectable objects that have been created in the specified version of the model.

```
ttGridAdmin connectableList [-latest|current|version n] [-dbdef]
```
Connectable operations

Options

The `connectableList` command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-latest</td>
<td>Lists connectable objects in the latest model—the model being modified and not yet applied to the grid. This is the default.</td>
</tr>
<tr>
<td>-current</td>
<td>Lists connectable objects in the current model—the model most recently applied to the grid.</td>
</tr>
<tr>
<td>-version n</td>
<td>Lists connectable objects in the specified version number of the model.</td>
</tr>
<tr>
<td>-dbdef</td>
<td>Also lists the name of the database definition associated with each connectable object.</td>
</tr>
</tbody>
</table>

Examples

This example lists connectables and the associated database definitions from the model most recently applied to the grid.

```
% ttGridAdmin connectableList -current -dbdef
Connectable   DbDef
-------------- -----------
database1     database1
database1client database1
```

Only `database1client` was created by the user. A direct connectable is also automatically defined for each user-created database definition.

Modify a connectable (connectableModify)

The `connectableModify` command modifies a connectable object.

```
ttGridAdmin connectableModify [-only hostname[.instancename]] filepath
```

Options

The `connectableModify` command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-only hostname[.instancename]</td>
<td>For connectables created for client/server connections (as determined by the <code>-cs</code> option of <code>connectableCreate</code>), optionally specifies that applications connecting through this connectable should connect to the indicated instance. By default, all data instances in the grid are available and connections are distributed among them. You need to specify the instance name only if there is more than one instance on the host. You can specify multiple <code>-only</code> options, in which case client/server applications can connect to any instance listed in the <code>-only</code> settings. <strong>Note:</strong> Instances specified with <code>connectableModify -only</code> replace any instances specified with <code>connectableCreate -only</code>.</td>
</tr>
<tr>
<td>filepath</td>
<td>Path and name of the connectable file, which contains the new set of connection attribute settings. The file name must be of the form <code>connname.connect</code>, where <code>connname</code> is the name of the connectable.</td>
</tr>
</tbody>
</table>
Examples

The example modifies the client/server connectable created in the `connectableCreate` example, editing `database1client.connect` to add a `PermWarnThreshold` setting:

```
ConnectionCharacterSet=AL32UTF8
UID=ttclient
PermWarnThreshold=80
```

Modify the connectable:

```
% ttGridAdmin ttGridAdmin connectableModify
/sw/tten/grid/conndefs/database1client.connect
Connectable database1client modified.
```

Notes

- All connection attribute settings from the previous connectable file are replaced with the connection attribute settings in the specified file.

- You cannot modify a connectable to change from client/server use to direct mode use, or from direct mode use to client/server use. Instead, delete the connectable and create a new one.

- When you apply the model after modifying a connectable, new versions of all necessary configuration files are written to each data instance, with the connectable entry modified according to the new settings. (Never edit configuration files manually. They are overwritten each time the model is applied.)
Data space group operations

Use the ttGridAdmin command in this section to get recommendations for assignments of hosts to data space groups.

Get recommendations for data space group assignments (dataSpaceGroupSuggest)

The dataSpaceGroupSuggest command recommends data space group assignments for hosts, based on the physical topology of the grid as defined in the model. Recommendations in the form of appropriate hostModify commands are output to a shell script (or to stdout).

```
ttGridAdmin dataSpaceGroupSuggest [filepath]
```

Options

The dataSpaceGroupSuggest command has the option:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>filepath</td>
<td>Path and name of the script to contain the recommended actions.</td>
</tr>
<tr>
<td></td>
<td>If no file is specified, the script is written to stdout.</td>
</tr>
</tbody>
</table>

Examples

This example shows a ttGridAdmin dataSpaceGroupSuggest command that writes its recommendations to recommendations.sh. (The output is abbreviated.) If you then execute this file, it will run ttGridAdmin hostModify commands to assign hosts to the recommended data space groups.

```
% ttGridAdmin dataSpaceGroupSuggest /tmp/recommendations.sh
% more /tmp/recommendations.sh
#!/bin/sh
# Recommendations generated by ttGridAdmin -dataSpaceGroupSuggest at Wed Aug 8 17:40:35 PDT 2018
TIMESTEN_HOME=/sw/tten/grid/ttinstances/grid1_mgmt
export TIMESTEN_HOME
. $TIMESTEN_HOME/bin/ttenv.sh > /dev/null 2>/dev/null

# Number of possibilities evaluated: 126
# Number of usable possibilities found: 10
# (A 'usable' possibility is one that is compatible with pre-existing assignments of Hosts to DataSpaceGroups)
# Number of 'ideal' possibilities found: 1
# (An 'ideal' possibility is one where no PhysicalGroups span multiple DataSpaceGroups)
# Possibilities evaluated (best 10 displayed):
# ...
# This script, if executed, would implement the only 'ideal' configuration found.
# Even though this recommendation was 'ideal', you should carefully evaluate it prior to running this script.
# Host mysys1host is already in DataSpaceGroup 1
ttGridAdmin hostModify mysys7host -dataSpaceGroup 1
ttGridAdmin hostModify mysys8host -dataSpaceGroup 1
```
If you decide to accept these recommendations, execute the provided shell script, `recommendations.sh`. Once the script is executed, all hosts are assigned to the designated data space groups.

```
% sh /scratch/gridsetup/misc/recommendations.sh
Host mysys7host modified in Model
Host mysys8host modified in Model
Host mysys9host modified in Model
Host mysys10host modified in Model
Host mysys2host modified in Model
```

### Notes

- Run the command after you have created physical groups, hosts, and installations for the grid and defined how hosts will correspond to physical groups (which you can accomplish through the `hostCreate -physicalgroup` option or the `hostModify -physicalgroup` or `-addphysicalgroup` option).

- This command is for hosts not yet assigned to a data space group, and returns an error if all hosts are already assigned. (Once a host is assigned to a data space group and `modelApply` is executed, you cannot change the assignment.)

- Hosts that do not have any instances are not included in the output.

- Recommendations are based on separating hosts among data space groups in an effort to avoid single points of failure, according to shared physical resources. In addition to the `hostModify` commands, the script has comments explaining the rationale behind the recommendations.

- If there is not enough difference in physical locations for `dataSpaceGroupSuggest` to make meaningful suggestions, it will output a message to that effect:

```
Error 33: Not enough differentiation to make suggestions
```

- For additional information, see "Assigning hosts to data space groups" and "Propose data space group assignments" in Oracle TimesTen In-Memory Database Scaleout User’s Guide.
Database definition operations

Use `ttGridAdmin` commands in this section to create, delete, export, or modify a database definition, or to display a list of existing database definitions.

A database definition specifies the characteristics of a database, consisting of settings for data store attributes (set at database creation) and first connect attributes.

Create a database definition (dbdefCreate)

The `dbdefCreate` command creates a database definition object in the model, defining characteristics of the database according to attribute settings in the specified file. It also creates a direct connectable object with the same name as the database, for direct connections.

```
$ ttGridAdmin dbdefCreate filepath
```

Once a database definition is added to the model, it can be used to create a database.

Options

The `dbdefCreate` command has the option:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>filepath</code></td>
<td>Path and name of the database definition file, which contains the attribute settings for the database definition. The file name must be of the form <code>dbname.dbdef</code>, where <code>dbname</code> defines the name of the database.</td>
</tr>
</tbody>
</table>

Examples

The example uses database definition file `database1.dbdef`:

```
$DataStore=/disk/databases/database1
$LogDir=/disk2/logs
$DatabaseCharacterSet=AL32UTF8
$ConnectionCharacterSet=US7ASCII
$PermSize=256
$TempSize=128
```

Typical settings in a database definition file include the following, as shown in the example.

- **Data store attributes**: `DataStore` (required) `LogDir`, and `DatabaseCharacterSet` (required)
  
  Directories are created on each host as necessary for the `DataStore` and `LogDir` locations.

- **First connection attributes**: `PermSize` (required) and `TempSize`

- **General connection attribute**: `ConnectionCharacterSet`

- As necessary, PL/SQL first connection attributes and server connection attributes

Create the database definition object:

```
$ ttGridAdmin dbdefCreate /sw/tten/grid/dbdefs/database1.dbdef
```

Database Definition `database1` created.
Data store attributes and first connection attributes go in the resulting database definition. From the example, this consists of DataStore, LogDir, DatabaseCharacterSet, PermSize, and TempSize. In addition, a default connections setting and a default durability setting are added to the definition automatically.

```
[database1]
DataStore=/disk/databases/database1
DatabaseCharacterSet=AL32UTF8
LogDir=/disk2/logs
PermSize=256
TempSize=128
connections=100
durability=0
```

General connection attributes go in the resulting connectable definition that is automatically created. In the example, this consists of ConnectionCharacterSet:

```
[database1]
ConnectionCharacterSet=US7ASCII
```

For additional information, see "Creating a database definition file" in Oracle TimesTen In-Memory Database Scaleout User’s Guide.

Notes

- The database name cannot be the same as any existing database or connectable name.
- The database definition file must be in odbc.ini format, as shown in the example, with attribute=value on each line.
- A dbdef file supports the following substitution strings for DataStore and LogDir entries. They are replaced as appropriate when the model is applied:
  - !!HOST_NAME!! is replaced on each host by the host name as specified in the model.
  - !!INSTANCE_NAME!! is replaced in each instance by the instance name as specified in the model.

A scenario for using !!HOST_NAME!! and !!INSTANCE_NAME!!, for example, would be if you use a Storage Area Network device shared between the hosts of the grid. Setting DataStore to /shared/datastores/!!HOST_NAME!!/!!INSTANCE_NAME!! gives each host its own data storage area.

- It is best practice to specify LogDir and have it be on a different file system from DataStore. The DataStore and LogDir paths and directories will be created wherever necessary.
- You can create additional connectables as described in "Create a connectable (connectableCreate)" on page 4-17.
- Some connection attributes cannot be set in a dbdef file, although they could be set for additional connectables that you create. For example, because the initial connectable that is created during database definition must be usable by the instance administrator, the UID and PWD connection attributes cannot be specified in a dbdef file.
- When you apply the model after creating a database definition, new versions of all necessary configuration files are written to each data instance, with an entry added according to the dbdef settings. (Never edit configuration files manually. They are overwritten each time the model is applied.)
Delete a database definition (dbdefDelete)

The `dbdefDelete` command removes a database definition object from the model.

```
ttGridAdmin dbdefDelete name
   [-cascade|-nocascade]
```

**Options**

The `dbdefDelete` command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>name</code></td>
<td>The name of the database definition object to delete.</td>
</tr>
<tr>
<td><code>-cascade</code></td>
<td>Also remove any additional connectable objects that were created for this database. This is the default.</td>
</tr>
<tr>
<td><code>-nocascade</code></td>
<td>Do not remove connectable objects.</td>
</tr>
</tbody>
</table>

**Examples**

This example deletes the database definition object `database2`, showing database definition object listings before and after.

```
% ttGridAdmin dbdefList
database1
database2

% ttGridAdmin dbdefDelete database2
Database Definition database2 deleted.

% ttGridAdmin dbdefList
database1
```

**Notes**

- The database itself must have already been destroyed (or not yet been created).
- When you apply the model after deleting a database definition, new versions of all necessary configuration files are written to each data instance, with the database definition removed. (Never edit configuration files manually. They are overwritten each time the model is applied.)
- The connectable object that was automatically created when the database definition object was created is also removed, regardless of the `-cascade/-nocascade` setting.

Export a database definition (dbdefExport)

The `dbdefExport` command exports an existing database definition object from the model, typically to a specified file.

```
ttGridAdmin dbdefExport name
   [filepath]
```

**Options**

The `dbdefExport` command has the options:
Examples

This example exports the database definition created in the dbdefCreate example above to the file database1.dbdef.

```bash
% ttGridAdmin dbdefExport database1 /sw/tten/grid/dbdefs/database1.dbdef
```

Resulting contents of database1.dbdef:

```bash
# DbDef GUID BCC6AB97-FDC2-4453-AEBC-5BFCAA57EA52 Exported 2017-12-06 19:05:03
[database1]
DataStore=/disk/databases/database1
DatabaseCharacterSet=AL32UTF8
LogDir=/disk2/logs
PermSize=256
TempSize=128
connections=100
durability=0
```

Notes

- The database definition is exported in odbc.ini format, as shown in the example, with attribute=value on each line.
- A typical use case is if you want to modify a database definition, but the original database definition file is no longer available.

List database definitions (dbdefList)

The dbdefList command lists the database definition objects that exist in the specified version of the model.

```
% ttGridAdmin dbdefList [-latest|-current|-version n]
```

Options

The dbdefList command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-latest</td>
<td>Lists database definition objects in the latest model—the model being modified and not yet applied to the grid. This is the default.</td>
</tr>
<tr>
<td>-current</td>
<td>Lists database definition objects in the current model—the model most recently applied to the grid.</td>
</tr>
<tr>
<td>-version n</td>
<td>Lists database definition objects in the specified version number of the model.</td>
</tr>
</tbody>
</table>

Examples

List database definition objects in the latest model (default) after the database1 database definition object was created (as shown in "Create a database definition")
Modify a database definition (dbdefModify)

The dbdefModify command modifies an existing database definition object in the model, defining characteristics of the database according to attribute settings in the specified file.

```
ttGridAdmin dbdefModify filepath
```

Options

The dbdefModify command has the option:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>filepath</td>
<td>Path and name of the file containing the database definition that will modify the database definition object. The file name must be of the form <code>dbname.dbdef</code>, where <code>dbname</code> is the name of the database.</td>
</tr>
</tbody>
</table>

Examples

This example modifies database1, created in "Create a database definition (dbdefCreate)" on page 4-24. The database definition file `database1.dbdef` was updated to change the PermSize and TempSize:

```
% cd /sw/tten/grid/dbdefs
% more database1.dbdef
DataStore=/disk/databases/database1
LogDir=/disk2/logs
DatabaseCharacterSet=AL32UTF8
ConnectionCharacterSet=US7ASCII
PermSize=512
TempSize=256
```

Modify the database definition object:

```
% ttGridAdmin dbdefModify /sw/tten/grid/dbdefs/database1.dbdef
Database Definition database1 modified.
```

Notes

- Database definition files are in `odbc.ini` format, as shown in the example, with `attribute=value` on each line. See "Create a database definition (dbdefCreate)" on page 4-24 for additional discussion.
- Data store attributes, as listed in Table 2–1 (for example, DataStore, DatabaseCharacterSet, LogDir, and Durability), are frozen once a database is created. Trying to change them using dbdefModify will have no effect on the database.
- If the database exists and is loaded, changes by the dbdefModify command to first connection attributes do not take effect until you unload (`dbUnload`) and load (`dbLoad`) the database.
- This command does not modify the database itself, only the database definition object.
- The connectable object that was automatically created when the database definition object was created is also modified appropriately.

- When you apply the model after modifying a database definition, new versions of all necessary configuration files are written to each data instance, with the applicable entry modified according to the `dbdef` settings. (Never edit configuration files manually. They are overwritten each time the model is applied.)

- The specified definition completely replaces the previous definition, deleting previous attribute settings for the database definition and connectable definition. Attributes that were set previously but are not set in the new definition will take their default settings.
Database operations

Use `ttGridAdmin` commands in this section to perform operations on databases, including creating, destroying, loading, unloading, opening, closing, importing, and exporting a database; setting the distribution scheme of a database; determining status of a database; and forcing the termination of connections to the database.

Close a database (dbClose)

The `dbClose` command closes the database so that applications can no longer connect to it.

```
ttGridAdmin dbClose name [-nowait | -wait [timeout]]
```

Options

The `dbClose` command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Name of the database to close.</td>
</tr>
<tr>
<td>-nowait</td>
<td>-wait [timeout]</td>
</tr>
</tbody>
</table>

Examples

This example closes a database without waiting for the elements to be closed on all instances, then checks status (after the database was successfully closed):

```
% ttGridAdmin dbClose database1
Database database1 close started
...
% ttGridAdmin dbStatus database1
Database database1 summary status as of Mon Nov 13 19:27:48 PST 2017

created,loaded-complete,closed
Completely created elements: 4 (of 4)
Completely loaded elements: 4 (of 4)
Completely created replica sets: 2 (of 2)
Completely loaded replica sets: 2 (of 2)

Open elements: 0 (of 4)
```
Notes

- After the command has completed, the database is still loaded but is closed to connections. Only the instance administrator can connect to a closed database.
- If you run $dbClose$ asynchronously (without waiting), you can use the $dbStatus$ command to see when the database is closed.
- The command does not close existing database connections. Any previously open connections must be terminated independently.
- If a database has been closed with $dbClose$, attempting to close it again typically results in an error. However, if any element is in "close failed" state, you can retry $dbClose$. Doing so will change any element in "close failed" state to "opened" state, which will result in TimesTen Scaleout trying to close it again.

Create a database ($dbCreate$)

The $dbCreate$ command creates a database in the grid according to the specified database definition.

```
$ttGridAdmin dbCreate name
    [-instance hostname[.instancename]]
    [-nowait | -wait [timeout]]
```

Options

The $dbCreate$ command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Name of the database definition to use in creating the database. This becomes the name of the database.</td>
</tr>
</tbody>
</table>
| -instance hostname[.instancename] | If specified, database element(s) will be created only on the specified instance(s), instead of on all instances of the grid. Any element previously created successfully on any of the specified instances must first be destroyed.  
  This is typically used to recover after a failure in the grid or after database elements were not successfully created on one or more instances in a previous execution of $dbCreate$.  
  The $hostname$ is required. The $instancename$ is required only if there is more than one instance on the host. (See "Grid objects and object naming" on page 4-5.)  
  You can use this option only once, specifying a single instance, in a single command. |
Examples

This example creates a database without waiting for the elements to be created on all instances, then checks the status, first while database creation is still in progress, then after it is complete.

```
% ttGridAdmin dbCreate database1
Database database1 creation started
...
% ttGridAdmin dbStatus database1
Database database1 summary status as of Mon Nov 13 18:38:39 PST 2017
creating,loading-partial,closed
Completely created elements: 1 (of 4) (3 in progress)
Completely loaded elements: 1 (of 4) (3 in progress)
Completely created replica sets: 0 (of 0)
Completely loaded replica sets: 0 (of 0)
Open elements: 0 (of 4)
...
```

In the following example, element creation on one instance (assume msys5host.griddata3) fails. The example tries again to create the element on that instance after the problem is resolved.

```
% ttGridAdmin dbCreate database1
Database database1 creation started
...
% ttGridAdmin dbStatus database1
Database database1 summary status as of Sat Nov 11 14:23:05 PST 2017
created-partial,loaded,closed
Completely created elements: 3 (of 4) (1 failed)
```

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
Completely loaded elements: 3 (of 4)
Completely created replica sets: 0 (of 0)
Completely loaded replica sets: 0 (of 0)

Open elements: 0 (of 4)

(Resolve the problem with mysys5host.griddata3.)

% ttGridAdmin dbCreate database1 -instance mysys5host.griddata3
Database database1 creation started
...
% ttGridAdmin dbStatus database1
Database database1 summary status as of Mon Nov 13 13:44:12 PST 2017

created,loaded,closed
Completely created elements: 4 (of 4)
Completely loaded elements: 4 (of 4)
Completely created replica sets: 0 (of 0)
Completely loaded replica sets: 0 (of 0)

Open elements: 0 (of 4)

**Notes**

- Each instance creates its element of the database, loads the element into memory, and records the state of the element.
- If you run dbCreate asynchronously (without waiting), you can use the dbStatus command to see when the database is created.
- The database is marked as "existing" as soon as the dbCreate command returns. If you run the command in the default -nowait mode, you can unload the database while its creation is still in progress.
- The database is not available for connections from users other than the instance administrator until you define the database distribution map with dbDistribute and open the database with dbOpen.
- A typical use case for the -instance option is when an element of the database had previously failed, been evicted or removed from the database distribution map, and been destroyed. (Then also use dbDistribute to add the element to the distribution map.)

**Destroy a database (dbDestroy)**

The dbDestroy command destroys the specified database. All data and schema contained in the database are irretrievably lost.

```
ttGridAdmin dbDestroy name
    [-instance hostname[.instancename]]
    [-nowait | -wait [timeout]]
```

**Options**

The dbDestroy command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Name of the database to destroy.</td>
</tr>
</tbody>
</table>
Examples

This example destroys a database without waiting for the elements to be destroyed on all instances. A subsequent attempt to check status indicates that the database was successfully destroyed.

```bash
% ttGridAdmin dbDestroy database1
Database database1 destroy started
...
% ttGridAdmin dbStatus database1
Error 2: Database database1 does not exist
```

This example destroys two of the four elements in the database. Both elements are from the same replica set and had previously been evicted.

```bash
% ttGridAdmin dbDestroy database1 -instance mysys3host.griddatal
Database database1 instance mysys3host.griddatal destroy started
% ttGridAdmin dbDestroy database1 -instance mysys4host.griddatal2
Database database1 instance mysys4host.griddatal2 destroy started
% ttGridAdmin dbStatus database1
Database database1 summary status as of Tue Jan 9 16:04:16 PST 2018
created,unloaded,closed
Completely created elements: 2 (of 4)
Completely loaded elements: 0 (of 4)
Completely created replica sets: 1 (of 1)
Completely loaded replica sets: 0 (of 1)
Open elements: 0 (of 4)
```
Notes

- The database must be unloaded or unloading.
- If you run `dbDestroy` asynchronously (without waiting), you can use the `dbStatus` command to see when the database is removed.
- A typical use case for the `-instance` option is after an element of the database failed and was evicted or removed from the database distribution map. Then using `dbDestroy` with `-instance` recovers the disk space of the failed element.

Force all connections to disconnect (dbDisconnect)

The `dbDisconnect` command forces all user connections to the specified database to be disconnected. This is useful, for example, prior to maintenance operations. Closing connections is mandatory to ensure a smooth shutdown and no data loss.

```
ttGridAdmin dbDisconnect name
-transactional|-immediate|-abort
[-nowait | -wait [timeout]]
```

No new transactions are allowed before the command executes.

A disconnection request is sent to each data instance in the grid.


---

**Note:** The `dbDisconnect` command does not affect subdaemon connections.

---

Options

The `dbDisconnect` command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Name of the database.</td>
</tr>
</tbody>
</table>
Examples

This example:

1. Uses dbStatus to show existing connections.
2. Closes the database and confirms.
3. Disconnects in transactional mode (without wait).

4. Checks status of the dbDisconnect command with dbDisconnectStatus and the status of the connections with dbStatus. (The dbDisconnect command is in progress and the connections still exist.)

5. Disconnects in immediate mode (without wait), to be sure connections are closed.

6. Again checks status of the dbDisconnect command with dbDisconnectStatus and the status of the connections with dbStatus. (The dbDisconnect command has completed and the connections are gone.)

```
% ttGridAdmin dbStatus database1 -connections
+---------+----------+------+-+-------+--------+--------+
<table>
<thead>
<tr>
<th>Host</th>
<th>Instance</th>
<th>ConnId</th>
<th>Name</th>
<th>Pid</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>mysys5host</td>
<td>instance3</td>
<td></td>
<td>database1</td>
<td>20233</td>
<td>Direct</td>
</tr>
<tr>
<td>mysys4host</td>
<td>instance2</td>
<td></td>
<td>database1</td>
<td>26529</td>
<td>Direct</td>
</tr>
<tr>
<td>mysys3host</td>
<td>instance1</td>
<td></td>
<td>database1</td>
<td>1600</td>
<td>Direct</td>
</tr>
<tr>
<td>mysys6host</td>
<td>instance4</td>
<td></td>
<td>database1</td>
<td>1678</td>
<td>Direct</td>
</tr>
</tbody>
</table>
% ttGridAdmin dbClose database1
Database database1 close started

% ttGridAdmin dbStatus database1
Database database1 summary status as of Tue Nov 27 16:12:16 PST 2018

created,loaded-complete,closed
Completely created elements: 4 (of 4)
Completely loaded elements: 4 (of 4)
Completely created replica sets: 2 (of 2)
Completely loaded replica sets: 2 (of 2)
Open elements: 0 (of 4)

First try disconnecting in transactional mode:

% ttGridAdmin dbDisconnect database1 -transactional
Database database1 dbDisconnect started

Let some time pass, then check status—connections still exist:

% ttGridAdmin dbDisconnectStatus database1

```
+---------+----------+-------+-+-------+--------+--------+
<table>
<thead>
<tr>
<th>Database</th>
<th>Host</th>
<th>Instance</th>
<th>Elem</th>
<th>State</th>
<th>Started</th>
</tr>
</thead>
<tbody>
<tr>
<td>database1</td>
<td>mysys5host</td>
<td>instance3</td>
<td>1</td>
<td>Disconnecting</td>
<td>2018-11-27T16:12:55.000Z</td>
</tr>
<tr>
<td></td>
<td>mysys4host</td>
<td>instance2</td>
<td>2</td>
<td>Disconnecting</td>
<td></td>
</tr>
<tr>
<td></td>
<td>mysys3host</td>
<td>instance1</td>
<td>3</td>
<td>Disconnecting</td>
<td></td>
</tr>
<tr>
<td></td>
<td>mysys6host</td>
<td>instance4</td>
<td>4</td>
<td>Disconnecting</td>
<td></td>
</tr>
</tbody>
</table>
% ttGridAdmin dbStatus -connections

Database database1:

```
+---------+----------+------+-+-------+--------+--------+
<table>
<thead>
<tr>
<th>Host</th>
<th>Instance</th>
<th>ConnId</th>
<th>Name</th>
<th>Pid</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>mysys5host</td>
<td>instance3</td>
<td></td>
<td>database1</td>
<td>20233</td>
<td>Direct</td>
</tr>
<tr>
<td>mysys4host</td>
<td>instance2</td>
<td></td>
<td>database1</td>
<td>26529</td>
<td>Direct</td>
</tr>
<tr>
<td>mysys3host</td>
<td>instance1</td>
<td></td>
<td>database1</td>
<td>1600</td>
<td>Direct</td>
</tr>
<tr>
<td>mysys4host</td>
<td>instance4</td>
<td></td>
<td>database1</td>
<td>1678</td>
<td>Direct</td>
</tr>
</tbody>
</table>

Try again in immediate mode:
% ttGridAdmin dbDisconnect database1 -immediate
Database database1 dbDisconnect started

Check status again—now the connections are gone.:
% ttGridAdmin dbDisconnectStatus database1
Database Host Instance Elem State Started
--------- ---------- --------- ---- ------------ ------------------------
database1                           Complete     2018-11-27T16:14:03.000Z
    mysys5host instance3 1 Disconnected
    mysys4host instance2 2 Disconnected
    mysys3host instance1 3 Disconnected
    mysys6host instance4 4 Disconnected

% ttGridAdmin dbStatus database1 -connections
Host Instance ConnId Name Pid Type CHost CAddr CPid
---- -------- ------ ---- --- ---- ----- ----- ----
%

Notes

■ The database must be in a closed state before you execute this command. (Closing a database does not affect existing connections, but does prevent new connections.)

■ In TimesTen Scaleout, the capability to force disconnections is always enabled and the forceDisconnectEnabled connection attribute is ignored.

Check status of forced disconnection (dbDisconnectStatus)
The dbDisconnectStatus command reports the status of the executing or most recently executed dbDisconnect command.

ttGridAdmin dbDisconnectStatus name

Any of these states may be reported for the overall status of the dbDisconnect command:

■ Defined: Disconnect has been requested but not yet initiated.
■ Disconnecting: Disconnect is still in progress on at least one element.
■ Failed: Disconnect failed on at least one element.
■ Complete: Disconnect completed successfully on all elements.

Any of these states may be reported for the status of the disconnect on any given instance:

■ Disconnecting: Disconnect is in progress on the instance.
■ Failed: Disconnect failed on the instance.
■ Disconnected: Disconnect completed successfully on the instance.

Options

The dbDisconnectStatus command has the option:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Name of the database.</td>
</tr>
</tbody>
</table>
Examples

A `dbDisconnectStatus` example is included in the `dbDisconnect` example in the preceding section.

Set or modify the distribution scheme of a database (dbDistribute)

The `dbDistribute` command can add, remove, evict, and replace elements of a database in the distribution map of the database, then distribute or redistribute data among elements. You must always use `-apply` to apply changes and redistribute data. You can do this either in the same command or in a separate command.

```
ttGridAdmin dbDistribute name
   [-list]
   [-add all | hostname[.instancename]]
   [-remove hostname[.instancename] [-replaceWith hostname[.instancename]]]
   [-evict hostname[.instancename] [-replaceWith hostname[.instancename]]]
   [-apply|-reset|-resync]
```

See "Define the distribution map of the database" in *Oracle TimesTen In-Memory Database Scaleout User’s Guide* for additional information.

Options

The `dbDistribute` command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>name</code></td>
<td>Name of the database for data distribution changes.</td>
</tr>
<tr>
<td>`-add all</td>
<td>hostname[.instancename]`</td>
</tr>
<tr>
<td><code>-remove hostname[.instancename]</code></td>
<td>Use this option in any circumstance where you want to remove, and optionally replace, an element, such as to replace an older host system with a newer one. Also see Notes below. Specify one instance per usage, but you can use <code>-remove</code> more than once on a command line. It is typical to use <code>-replaceWith</code> to replace the element. The <code>-remove</code> option without <code>-replaceWith</code> results in redistribution of data. If you have a grid with $k=2$ and you remove one element of a replica set, you must either replace it or also remove the other element of the replica set. Also see &quot;Redistributing data in a database&quot; in <em>Oracle TimesTen In-Memory Database Scaleout User’s Guide</em>.</td>
</tr>
</tbody>
</table>
Database operations

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-evict hostname[.instancename]</td>
<td>Use this option if all elements of a replica set (one element if k=1, two elements if k=2) have unrecoverable failures and you cannot repair them. Specify one instance per usage, but you can use -evict more than once on a command line. If you use -evict, you must evict all elements in the replica set. You can use -replaceWith to replace the element. <strong>Important</strong>: Using the -evict option inevitably results in data loss. Use this only as a last resort. See Notes below for additional considerations. Also see &quot;Redistributing data in a database&quot; in Oracle TimesTen In-Memory Database Scaleout User’s Guide.</td>
</tr>
<tr>
<td>-replaceWith hostname[.instancename]</td>
<td>Optionally use this with -evict or -remove to have the specified replacement contain the same data. The element on the replacing instance must not have previously been added to the distribution. The -replaceWith option must immediately follow the corresponding -remove or -evict option on the command line.</td>
</tr>
<tr>
<td>-list</td>
<td>Displays the current and pending distribution map of the database (&quot;Holds Data&quot; and &quot;Will Hold Data&quot;, respectively).</td>
</tr>
<tr>
<td>-apply</td>
<td>Applies the new distribution to the database. You can use this option by itself to apply settings from previous commands, or in the same command line with the settings.</td>
</tr>
<tr>
<td>-reset</td>
<td>Discards all distribution settings that have not yet been applied. This option cannot be combined with any other option.</td>
</tr>
<tr>
<td>-resync</td>
<td>Attempts to resynchronize metadata in the user database with metadata in the active management instance in case the state of a dbDistribute -apply command is unknown. For example, the user database and management instance may not have matching states due to some failure or loss of communication. In some cases, the management instance may not know about the success or failure of a dbDistribute operation on the data instances and is left in an intermediate state. This option cannot be used with any other dbDistribute options. See &quot;Recovering from a data distribution error&quot; in Oracle TimesTen In-Memory Database Scaleout User’s Guide for related information. <strong>Note</strong>: The -resync option results in metadata in the management instance being read to see if there is a dbDistribute operation that is in progress but was neither committed nor rolled back. Resynchronizing may involve committing or rolling back the metadata changes of the dbDistribute operation (which are intended to be recorded in the management instance).</td>
</tr>
</tbody>
</table>
Examples

This example adds all elements in the grid to the distribution map then distributes data among the elements:

```plaintext
% ttGridAdmin dbDistribute database1 -add all -apply
Distribution map updated
```

You can then use the `-list` option to show the distribution map of elements in the grid (elements able to hold data):

```plaintext
% ttGridAdmin dbDistribute database1 -list
Distribution Map version: 1
RS Host        Instance  Holds Data Will Hold Data Removed Evicted
-- ----------- --------- ---------- -------------- ------- -------
1  mysys3host griddata1          Y              Y       N       N
1  mysys4host griddata2          Y              Y       N       N
2  mysys5host griddata3          Y              Y       N       N
2  mysys6host griddata4          Y              Y       N       N
```

Now remove both elements in replica set 1, then look at the `-list` output again, which indicates the two elements removed from the grid and therefore unable to hold data:

```plaintext
% ttGridAdmin dbDistribute database1 -remove mysys3host.griddata1
Element mysys3host.griddata1 is removed
Distribution map change enqueued
% ttGridAdmin dbDistribute database1 -remove mysys4host.griddata2
Element mysys4host.griddata2 is removed
Distribution map change enqueued
% ttGridAdmin dbDistribute database1 -apply
Distribution map updated
% ttGridAdmin dbDistribute database1 -list
Distribution Map version: 3
RS   Host       Instance  Holds Data Will Hold Data Removed Evicted
---- ---------- --------- ---------- -------------- ------- -------
NULL mysys3host griddata1          N              N       N       Y
NULL mysys4host griddata2          N              N       N       Y
1 mysys5host griddata3          Y              Y       N       N
1 mysys6host griddata4          Y              Y       N       N
```

The following is a new example that evicts two elements (from the same replica set) then looks at the `-list` output, which shows the two elements evicted from the grid and therefore unable to hold data.

```plaintext
% ttGridAdmin dbDistribute database1 -evict mysys3host.griddata1 -evict mysys4host.griddata2 -apply
Distribution map updated
% ttGridAdmin dbDistribute database1 -list
Distribution Map version: 2
RS Host       Instance  Holds Data Will Hold Data Removed Evicted
---- ---------- --------- ---------- -------------- ------- -------
NULL mysys3host griddata1          N              N       N       Y
NULL mysys4host griddata2          N              N       N       Y
1 mysys5host griddata3          Y              Y       N       N
1 mysys6host griddata4          Y              Y       N       N
```

This example shows where the `-resync` option successfully completed a data distribution operation:

```plaintext
% ttGridAdmin dbDistribute database1 -apply
```

TimesTen Scaleout Utilities 4-41
... (Process fails or is interrupted.)

% ttGridAdmin dbDistribute database1 -resync
Distribution map updated

And this example shows where the -resync option rolled back a data distribution operation:

% ttGridAdmin dbDistribute database1 -apply
...

(Process fails or is interrupted.)

% ttGridAdmin dbDistribute database1 -resync
Distribution map Rolled Back

Notes

- You cannot use -reset while distribution (-apply) is in progress. You can try -resync instead, as appropriate.
- You can use -list and -resync while distribution is in progress. Other operations will fail if distribution is in progress.
- Wait until the elements are loaded on all instances on which you will perform operations before using dbDistribute. You can use the dbStatus command to confirm this.
- If you use -add all, you must use -apply in the same command.
- To specify an element, express its instance as hostname[.instancename]. The host name is required. The instance name is required only if there are multiple instances on the host. (See "Grid objects and object naming" on page 4-5.)
- If you need to confirm which elements are in each replica set, use the dbStatus command with the -replicaSet option.
- Once an element has been removed or evicted from the distribution, the only possibility is to eliminate it with dbDestroy -instance. It is advisable to do that as soon as possible to reclaim the disk space that it used. If you are able to use the instance again later, you must recreate the element with dbCreate -instance, then add it to the distribution.
- If k=2 and one element of a replica set has an irrecoverable failure, use -remove and -replaceWith to make the replica set fully operational again. You cannot use -evict when an active replica is available.
- If all elements of any replica set are down, you cannot perform global operations. If you cannot recover any element of the replica set, evicting the elements of the replica set will allow you to perform global operations again, but you will suffer permanent loss of data.
- It is valid to use -add instead of -replaceWith to replace the elements of an evicted replica set, but in either case data on the evicted replica set is lost. Also note that -add results in redistribution of data while -replaceWith (used with either -evict or -remove) does not. See "Recovering when the replica set has a permanently failed element" in Oracle TimesTen In-Memory Database Scaleout User’s Guide for additional information.
List databases (dbList)

The dbList command lists the databases that have been created in the grid and indicates whether they have been loaded or opened.

```
ttGridAdmin dbList
```

Examples

```
% ttGridAdmin dbList
Database                     Loaded Opened
database1                    Y      Y
testdb                       Y      N
```

Load a database into memory (dbLoad)

The ttGridAdmin dbLoad command loads the specified database into memory. A database must be loaded and opened before it is used by applications.

```
ttGridAdmin dbLoad name [-nowait | -wait [timeout]]
```

Options

The dbLoad command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Name of the database to load.</td>
</tr>
<tr>
<td>-nowait</td>
<td>-wait [timeout]</td>
</tr>
</tbody>
</table>

Examples

This example loads a database without waiting for the elements to be loaded on all instances, then checks status (after the database was successfully loaded):

```
% ttGridAdmin dbLoad database1
Database database1 load started ...
% ttGridAdmin dbStatus database1
Database database1 summary status as of Mon Nov 13 18:58:53 PST 2017
created,loaded,closed
Completely created elements: 4 (of 4)
```
Notes

- Before loading a database, it is advisable to run `dbStatus` with the `-loadReadiness` option to confirm all replica sets can be loaded.
- After the command has completed, the database is loaded but closed. (Use `dbOpen` to open it.)
- It is not necessary to run `dbLoad` after `dbCreate`, because `dbCreate` loads the database automatically.
- If you run `dbLoad` asynchronously (without waiting), you can use the `dbStatus` command to see when the database is loaded.

Open a database (`dbOpen`)

The `dbOpen` command opens the database so that applications can connect to it.

```
ttGridAdmin dbOpen name [-nowait | -wait [timeout]]
```

Options

The `dbOpen` command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>name</code></td>
<td>Name of the database to open.</td>
</tr>
</tbody>
</table>
| `-nowait | -wait [timeout]`             | The command initiates a state change that is recorded in the active management instance of the grid. The `-nowait` option causes the command to return immediately without waiting for the state change. This is the default behavior. The `-wait` option causes the command to wait for the state change to complete, when the database element has been opened on each instance in the grid. You can optionally subject the wait to a limit of `timeout` seconds. Otherwise, or if `timeout` is set to 0, there is no limit. In a large grid, it is not typical or generally advisable to use `-wait`. If you do, it is advisable to set a timeout. (See "Database management operations" on page 4-7.)

Examples

This example opens a database without waiting for the elements to be opened on all instances, then checks status (after the database was opened successfully):

```
% ttGridAdmin dbOpen database1
Database database1 open started
...
% ttGridAdmin dbStatus database1
Database database1 summary status as of Mon Nov 13 19:24:39 PST 2017

  created,loaded-complete,open
Completely created elements: 4 (of 4)
```
Completely loaded elements: 4 (of 4)
Completely created replica sets: 2 (of 2)
Completely loaded replica sets: 2 (of 2)
Open elements: 4 (of 4)

Notes

- The database must be loaded or loading (performed automatically by \texttt{dbCreate}).
- The database must have a distribution map (\texttt{dbDistribute -apply}).
- If you run \texttt{dbOpen} asynchronously (without waiting), you can use the \texttt{dbStatus} command to see when the database is open.
- If a database has been opened with \texttt{dbOpen}, attempting to open it again typically results in an error. However, if any element is in "open failed" state, you can retry \texttt{dbOpen}. Doing so will change any element in "open failed" state to "loaded" state, which will result in TimesTen Scaleout trying to open it again.

Monitor the status of a database (\texttt{dbStatus})

The \texttt{dbStatus} command reports the status of a database or databases or the status of specified components of the database or databases, using information from the active management instance. This includes the status of any pending command to create, destroy, load, unload, open, or close the database. You can also request additional details, or request information about the state of each instance regarding whether its element can be loaded.

\begin{verbatim}
\end{verbatim}

You can also refer to \texttt{dbStatus} discussion and examples in "Recovering from Failure" in \textit{Oracle TimesTen In-Memory Database Scaleout User’s Guide}.

Options

The \texttt{dbStatus} command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Name of the database for which to display status. The default is to display status of all databases in the grid.</td>
</tr>
<tr>
<td>-summary</td>
<td>Shows an overall summary of database status. (This is the default mode.)</td>
</tr>
<tr>
<td>-element</td>
<td>Shows the status of each element of the database.</td>
</tr>
<tr>
<td>-replicaSet</td>
<td>Shows the status of each replica set of the database.</td>
</tr>
<tr>
<td>-dataSpaceGroup</td>
<td>Shows the status of each data space group of the database.</td>
</tr>
<tr>
<td>-all</td>
<td>Shows summary, element, replica set, and data space group status (equivalent to -summary -element -replicaSet -dataSpaceGroup).</td>
</tr>
</tbody>
</table>
The `dbStatus` command indicates the status of the database as a whole with a line showing overall created/destroyed, loaded/unloaded, and opened/closed states. (For example: “created, loaded-complete, closed”.)

The states of created, creating, destroyed, loading, loaded, unloaded, unloading, opening, opened, closing, and closed indicate that the corresponding database management command is in progress or has finished, as stated.

In addition:

- **created-partial or creating-partial**: Some elements of the database are in the process of being created or have successfully been created, but others could not be created.

- **createFailed**: Creation of the database failed. This occurs when no elements could be created, such as when every TimesTen instance is down.

- **loaded-partial or loading-partial**: The `dbDistribute` command has not yet been executed on the database (so no replica sets have been defined) and at least one element could not be created or loaded.

- **loaded-incomplete or loading-incomplete**: At least one replica has no elements that finished loading successfully.

- **loaded-functional or loading-functional**: At least one element from each replica set is loaded.

- **loaded-complete or loading-complete**: Every element loaded successfully.

- **notLoaded**: Loading of the database failed—none of the elements is loaded or loading.

### Overall database status

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-details</td>
<td>Shows daemon state information in addition to the status information from the active management instance. You can use this option in addition to any of the preceding options.</td>
</tr>
<tr>
<td>-loadReadiness</td>
<td>Shows information, including up/down status, indicating whether instances in each replica set are in a state where their elements can be loaded. It is advisable to use this option before trying to load a database. You can also use it while a load is in progress.</td>
</tr>
<tr>
<td>-epochs</td>
<td>Shows the most recent epochs available for each element of the grid, and the most recent epoch that could be used for recovery. An epoch is a transaction that marks a globally consistent point in time across all elements of the database. See “Epoch transactions” in Oracle TimesTen In-Memory Database Scaleout User’s Guide.</td>
</tr>
<tr>
<td>-connections</td>
<td>Displays information for existing connections to the specified database. Without the -proxy or -system suboptions, only application connections are shown.</td>
</tr>
<tr>
<td>-proxy</td>
<td>Used with the -connections option, this also displays information for all proxy connections associated with existing application connections. <strong>Note</strong>: This option cannot be used without the -connections option.</td>
</tr>
<tr>
<td>-system</td>
<td>Used with the -connections option, this also displays TimesTen internal connections, such as those used by subdaemons and TimesTen utilities. <strong>Note</strong>: This option cannot be used without the -connections option.</td>
</tr>
</tbody>
</table>
These states can help you determine if the grid is usable even if it is not fully operational. For example, you can execute `dbOpen` before all the elements have been loaded.

**Element status values**

The `dbStatus` command returns these database element status values:

---

**Note:** See "Troubleshooting distributed transactions" in *Oracle TimesTen In-Memory Database Scaleout User’s Guide* for recommendations regarding these status values.

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>close failed</td>
<td>The attempt to close the element failed.</td>
</tr>
<tr>
<td>closing</td>
<td>The element is in the process of closing.</td>
</tr>
<tr>
<td>create failed</td>
<td>The attempt to create the element failed.</td>
</tr>
<tr>
<td>creating</td>
<td>The element is being created.</td>
</tr>
<tr>
<td>destroy failed</td>
<td>The attempt to destroy the element failed.</td>
</tr>
<tr>
<td>destroyed</td>
<td>The element has been destroyed.</td>
</tr>
<tr>
<td>destroying</td>
<td>The element is being destroyed.</td>
</tr>
<tr>
<td>down</td>
<td>The data instance where this element is located is not running.</td>
</tr>
<tr>
<td>evicted</td>
<td>The element was evicted or removed through <code>dbDistribute</code> and has been unloaded from RAM.</td>
</tr>
<tr>
<td>evicted (loaded)</td>
<td>The element was evicted or removed through <code>dbDistribute</code> but unloading it from RAM has not yet begun.</td>
</tr>
<tr>
<td>evicted (unloading)</td>
<td>The element was evicted or removed through <code>dbDistribute</code> and is being unloaded from RAM.</td>
</tr>
<tr>
<td>load failed</td>
<td>The attempt to load the element into RAM failed.</td>
</tr>
<tr>
<td>loaded</td>
<td>The element is loaded into RAM.</td>
</tr>
<tr>
<td>loading</td>
<td>The element is being loaded into RAM.</td>
</tr>
<tr>
<td>opened</td>
<td>The element is open.</td>
</tr>
<tr>
<td>open failed</td>
<td>The attempt to open the element failed.</td>
</tr>
<tr>
<td>opening</td>
<td>The element is in the process of opening.</td>
</tr>
<tr>
<td>uncreated</td>
<td>The element should be created, but creation has not yet started.</td>
</tr>
<tr>
<td>unloaded</td>
<td>The element has been unloaded from RAM.</td>
</tr>
<tr>
<td>unloading</td>
<td>The element is being unloaded from RAM.</td>
</tr>
<tr>
<td>waiting for seed</td>
<td>The element will be loaded into RAM, but not until after the other element in its replica set is loaded.</td>
</tr>
</tbody>
</table>

**Connections status**

This section describes information displayed by the `-connections`, `-proxy`, and `-system` options that show existing connections.
### Database status examples

#### Examples

**Database status examples**

Key for these examples:
RS: Identifying number of the replica set that each element belongs to.

DS: Identifying number of the data space group that each element belongs to.

Elem: Element number for each element.

Status: Status of the operation on each element. See "Status values" above for the list of element states that can be returned.

This example shows complete dbStatus output after a database has had its distribution specified, but the database is closed.

% ttGridAdmin dbStatus database1 -all
Database database1 summary status as of Thu Nov 17 13:28:16 PST 2016
created,loaded-complete,closed
Completely created elements: 4 (of 4)
Completely loaded elements: 4 (of 4)
Completely created replica sets: 2 (of 2)
Completely loaded replica sets: 2 (of 2)
Open elements: 0 (of 4)

Database database1 element level status as of Thu Nov 17 13:28:16 PST 2016

<table>
<thead>
<tr>
<th>Host</th>
<th>Instance</th>
<th>Elem</th>
<th>Status</th>
<th>Date/Time of Event</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>mysys3host</td>
<td>griddata1</td>
<td>3</td>
<td>loaded</td>
<td>2016-11-16 17:36:39</td>
<td></td>
</tr>
<tr>
<td>mysys4host</td>
<td>griddata2</td>
<td>1</td>
<td>loaded</td>
<td>2016-11-16 17:36:40</td>
<td></td>
</tr>
<tr>
<td>mysys5host</td>
<td>griddata3</td>
<td>4</td>
<td>loaded</td>
<td>2016-11-16 17:36:39</td>
<td></td>
</tr>
<tr>
<td>mysys6host</td>
<td>griddata4</td>
<td>2</td>
<td>loaded</td>
<td>2016-11-16 17:36:41</td>
<td></td>
</tr>
</tbody>
</table>

Database database1 Replica Set status as of Thu Nov 17 13:28:16 PST 2016

<table>
<thead>
<tr>
<th>RS DS Elem Host</th>
<th>Instance</th>
<th>Status</th>
<th>Date/Time of Event</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1</td>
<td>mysys3host</td>
<td>loaded</td>
<td>2016-11-16 17:36:39</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 1</td>
<td>mysys5host</td>
<td>loaded</td>
<td>2016-11-16 17:36:39</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Database database1 Data Space Group status as of Thu Nov 17 13:28:16 PST 2016

<table>
<thead>
<tr>
<th>DS RS Elem Host</th>
<th>Instance</th>
<th>Status</th>
<th>Date/Time of Event</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1</td>
<td>mysys3host</td>
<td>loaded</td>
<td>2016-11-16 17:36:39</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 1</td>
<td>mysys5host</td>
<td>loaded</td>
<td>2016-11-16 17:36:39</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This example shows load readiness with all instances up, then with one instance in a replica set down, then with both instances in a replica set down. If all instances in a replica set are down, the database cannot be loaded.

% ttGridAdmin dbStatus database1 -loadReadiness
Data Elements:

<table>
<thead>
<tr>
<th>RS DS Instance</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Unloaded</td>
</tr>
<tr>
<td>2</td>
<td>Unloaded</td>
</tr>
<tr>
<td>1</td>
<td>Loadable</td>
</tr>
<tr>
<td>2</td>
<td>Unloaded</td>
</tr>
</tbody>
</table>
Connection status examples

Examples are shown for the -connections option by itself, -connections with -proxy, -connections with -system, and -connections with both -proxy and -system.

% ttgridadmin dbstatus database1 -connections

Connection status examples

Examples are shown for the -connections option by itself, -connections with -proxy, -connections with -system, and -connections with both -proxy and -system.
<table>
<thead>
<tr>
<th>Host</th>
<th>Instance</th>
<th>ConnId</th>
<th>Name</th>
<th>Pid</th>
<th>Type</th>
<th>PHost</th>
<th>PInstance</th>
<th>PPid</th>
<th>PConnId</th>
</tr>
</thead>
<tbody>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>1</td>
<td>database1</td>
<td>8631</td>
<td>Direct</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>2</td>
<td>con1</td>
<td>8631</td>
<td>Direct</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>3</td>
<td>con2</td>
<td>8631</td>
<td>Direct</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>128</td>
<td>Grid Epoch Generator(TM=2)</td>
<td>31183</td>
<td>GCW</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>129</td>
<td>ttStats Collector</td>
<td>31183</td>
<td>GCW</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>130</td>
<td>ttStats Collector</td>
<td>31871</td>
<td>TTStats</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>131</td>
<td>Garbage Collector</td>
<td>30876</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>132</td>
<td>Grid Watch Remote TM</td>
<td>30876</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>133</td>
<td>Grid Rem Elem Mon</td>
<td>30876</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>134</td>
<td>XactId Rollback</td>
<td>30876</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>135</td>
<td>Grid Epoch Generator</td>
<td>30876</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>136</td>
<td>Grid Seq Batch</td>
<td>30876</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>137</td>
<td>GOW Watcher</td>
<td>30876</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>138</td>
<td>HistGC</td>
<td>30876</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>139</td>
<td>Log Marker</td>
<td>30876</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>140</td>
<td>IndexGC</td>
<td>30876</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>141</td>
<td>Grid Task</td>
<td>30876</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>142</td>
<td>Deadlock Detector</td>
<td>30876</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>143</td>
<td>Flusher</td>
<td>30876</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>144</td>
<td>Monitor</td>
<td>30876</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>145</td>
<td>Checkpoint</td>
<td>30876</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>146</td>
<td>Rollback</td>
<td>30876</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>147</td>
<td>Manager</td>
<td>30876</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>1</td>
<td>database1cs</td>
<td>8653</td>
<td>C/S</td>
<td>mysys2 10.90.137.240</td>
<td>8637</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>2</td>
<td>con1</td>
<td>8666</td>
<td>C/S</td>
<td>mysys2 10.90.137.240</td>
<td>8637</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>3</td>
<td>con2</td>
<td>31210</td>
<td>GCW</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>4</td>
<td>con1</td>
<td>31210</td>
<td>GCW</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>128</td>
<td>Grid Epoch Generator(TM=1)</td>
<td>31210</td>
<td>GCW</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>129</td>
<td>ttStats Collector</td>
<td>31210</td>
<td>GCW</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>130</td>
<td>ttStats Collector</td>
<td>31878</td>
<td>TTStats</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>131</td>
<td>Grid Watch Remote TM</td>
<td>30950</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>132</td>
<td>Grid Seq Batch</td>
<td>30950</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>133</td>
<td>GOW Watcher</td>
<td>30950</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>134</td>
<td>Grid Epoch Generator</td>
<td>30950</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>135</td>
<td>Grid Rem Elem Mon</td>
<td>30950</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>136</td>
<td>XactId Rollback</td>
<td>30950</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>137</td>
<td>Garbage Collector</td>
<td>30950</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>138</td>
<td>Grid Task</td>
<td>30950</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>139</td>
<td>Log Marker</td>
<td>30950</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>140</td>
<td>IndexGC</td>
<td>30950</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>141</td>
<td>Grid Task</td>
<td>30950</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>142</td>
<td>Deadlock Detector</td>
<td>30950</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>143</td>
<td>Flusher</td>
<td>30950</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>144</td>
<td>Monitor</td>
<td>30950</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>145</td>
<td>Checkpoint</td>
<td>30950</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>146</td>
<td>Rollback</td>
<td>30950</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>147</td>
<td>Manager</td>
<td>30950</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Database operations

```
% ttgridadmin dbstatus database1 -connections -proxy -system

<table>
<thead>
<tr>
<th>Host</th>
<th>Instance</th>
<th>ConnId</th>
<th>Name</th>
<th>Pid</th>
<th>Type</th>
<th>PHost</th>
<th>PInstance</th>
<th>PPid</th>
<th>PConnId</th>
</tr>
</thead>
<tbody>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>1</td>
<td>database1</td>
<td>8631</td>
<td>Direct</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>2</td>
<td>con1</td>
<td>8631</td>
<td>Direct</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>2</td>
<td>con1</td>
<td>8631</td>
<td>Proxy</td>
<td>mysys2</td>
<td>instance2</td>
<td>31210</td>
<td>4</td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>3</td>
<td>con2</td>
<td>8631</td>
<td>Direct</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>3</td>
<td>con2</td>
<td>8631</td>
<td>Proxy</td>
<td>mysys2</td>
<td>instance2</td>
<td>31210</td>
<td>3</td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>128</td>
<td>Grid Epoch Generator(TM=2)</td>
<td>31183</td>
<td>GCW</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>129</td>
<td>ttStats Collector</td>
<td>31183</td>
<td>GCW</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>130</td>
<td>ttStats Collector</td>
<td>31871</td>
<td>TTStats</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>131</td>
<td>Garbage Collector</td>
<td>30876</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>132</td>
<td>Grid Watch Remote TM</td>
<td>30876</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>133</td>
<td>Grid Rem Elem Mon</td>
<td>30876</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>134</td>
<td>XactId Rollback</td>
<td>30876</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>135</td>
<td>Grid Epoch Generator</td>
<td>30876</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>136</td>
<td>Grid Seq Batch</td>
<td>30876</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>137</td>
<td>GCW Watcher</td>
<td>30876</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>138</td>
<td>HistGC</td>
<td>30876</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>139</td>
<td>Log Marker</td>
<td>30876</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>140</td>
<td>IndexGC</td>
<td>30876</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>141</td>
<td>Grid Task</td>
<td>30876</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>142</td>
<td>Deadlock Detector</td>
<td>30876</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>143</td>
<td>Flusher</td>
<td>30876</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>144</td>
<td>Monitor</td>
<td>30876</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>145</td>
<td>Checkpoint</td>
<td>30876</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>146</td>
<td>Rollback</td>
<td>30876</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys1</td>
<td>instance1</td>
<td>147</td>
<td>Manager</td>
<td>30876</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>1</td>
<td>database1cs</td>
<td>8653</td>
<td>C/S</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>2</td>
<td>con1</td>
<td>8666</td>
<td>C/S</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>3</td>
<td>con2</td>
<td>31210</td>
<td>GCW</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>4</td>
<td>con1</td>
<td>31210</td>
<td>GCW</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>128</td>
<td>Grid Epoch Generator(TM=1)</td>
<td>31210</td>
<td>GCW</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>129</td>
<td>ttStats Collector</td>
<td>31210</td>
<td>GCW</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>130</td>
<td>ttStats Collector</td>
<td>31878</td>
<td>TTStats</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>131</td>
<td>Garbage Collector</td>
<td>30950</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>132</td>
<td>Grid Watch Remote TM</td>
<td>30950</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>133</td>
<td>Grid Epoch Generator</td>
<td>30950</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>134</td>
<td>Grid Epoch Generator</td>
<td>30950</td>
<td>Proxy</td>
<td>mysys1</td>
<td>instance1</td>
<td>31183</td>
<td>128</td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>135</td>
<td>Grid Seq Batch</td>
<td>30950</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>136</td>
<td>XactId Rollback</td>
<td>30950</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>137</td>
<td>Garbage Collector</td>
<td>30950</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>138</td>
<td>Grid Task</td>
<td>30950</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>139</td>
<td>Log Marker</td>
<td>30950</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>140</td>
<td>Grid Rem Elem Mon</td>
<td>30950</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>141</td>
<td>Flusher</td>
<td>30950</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>142</td>
<td>IndexGC</td>
<td>30950</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>143</td>
<td>Checkpoint</td>
<td>30950</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>144</td>
<td>Deadlock Detector</td>
<td>30950</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mysys2</td>
<td>instance2</td>
<td>145</td>
<td>Monitor</td>
<td>30950</td>
<td>Subdaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
Unload a database (dbUnload)

The dbUnload command unloads the specified database from memory.

```
Usage:

ttGridAdmin dbUnload name

[-nowait | -wait [timeout]]
[-force]
```

### Options

The `dbUnload` command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>name</code></td>
<td>Name of the database to unload.</td>
</tr>
<tr>
<td><code>-nowait</code></td>
<td>The command initiates a state change that is recorded in the active management instance of the grid.</td>
</tr>
<tr>
<td><code>-wait [timeout]</code></td>
<td>The <code>-nowait</code> option causes the command to return immediately without waiting for the state change. This is the default behavior.</td>
</tr>
<tr>
<td><code>-force</code></td>
<td>If Durability=0 and at least one replica set is completely down, this option allows the unload to proceed anyway.</td>
</tr>
</tbody>
</table>

**Important:** Using this option will likely result in data loss. (Normally, to prevent data loss, a database with Durability=0 cannot be unloaded unless at least one element from every replica set is loaded.)

### Examples

This example unloads a database without waiting for the elements to be unloaded on all instances, then checks status (after the database was successfully unloaded).

```
% ttGridAdmin dbUnload database1
Database database1 unload started
...
% ttGridAdmin dbStatus database1
Database database1 summary status as of Mon Nov 13 18:52:47 PST 2017

created, unloaded, closed
Completely created elements: 4 (of 4)
Completely loaded elements: 0 (of 4)
Completely created replica sets: 0 (of 0)
Completely loaded replica sets: 0 (of 0)
```
Notes

- Do not begin any transactions after issuing a `dbUnload` command.
- All connections to the database must be closed.
- The database must be closed.
- If you run `dbUnload` asynchronously (without waiting), you can use the `dbStatus` command to see when the database is loaded.
Grid operations

Use the ttGridAdmin commands in this section to create a grid in the model, configure passwordless SSH for the grid, gather information about the grid, and make changes to the grid. There is also a command to produce a sys.odbc.ini file for use by clients outside of the grid.

Export sys.odbc.ini for client/server connections outside grid (gridClientExport)

The gridClientExport command produces a sys.odbc.ini file that can be used by TimesTen instances that are not part of the grid to access databases in the grid.

```
ttGridAdmin gridClientExport [filepath]
```

The resulting file contains definitions of all client/server connectables defined in the grid. You must manually copy this file to any TimesTen client instances outside of the grid from which you want to connect to databases in the grid.

Options

The gridClientExport command has the option:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>filepath</td>
<td>Path and name of the file where the sys.odbc.ini entries are written.</td>
</tr>
<tr>
<td></td>
<td>If no file is specified, the entries are output to stdout.</td>
</tr>
</tbody>
</table>

Examples

This example exports the sys.odbc.ini entries to the file sys_export.odbc.ini, then shows the contents of that file.

```
% ttGridAdmin gridClientExport /sw/tten/grid/clients/sys_export.odbc.ini
% cd /sw/tten/grid/clients
% more sys_export.odbc.ini

[ODBC Data Sources]
database1client=TimesTen 18.1 Client Driver

[database1client]
TTC_SERVER_DSN=databasel
TTC_SERVER1=mysys3.example.com/21000
TTC_SERVER2=mysys4.example.com/21000
UID=ttclient
```

Notes

This command uses the external address of the host.

Create a grid (gridCreate)

The gridCreate command creates a grid and the initial version of the model.

```
ttGridAdmin gridCreate name
   -k n
   -membershipConfig filepath
```
Grid operations

[-address addr]
[-internalAddress addr]
[-externalAddress addr]
[-mgmtPort n]
[-host name]
[-retainDays numdays]
[-retainVersions n]
[-warnThresh percent]
[-noDataSpaceGroup]
[-walletDir path]

The instance from which the command is run becomes the initial management instance of the new grid. Additional instances (data instances and a second management instance) can then be created and joined to the grid later.

Options

The gridCreate command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Specifies the name for the grid in the model.</td>
</tr>
<tr>
<td>-k n</td>
<td>Specifies the degree of K-safety that this grid provides. Specify a value of 1 or 2.</td>
</tr>
<tr>
<td>-membershipConfig filepath</td>
<td>Path and name of the membership client configuration file, which contains the host name and port of each membership server. The contents of this file will be automatically provisioned in every instance in the grid. Sample contents: Servers zk1.example.com!2181,zk2.example.com!2181, zk3.example.com!2181 Note: Either colons or exclamation marks can be used between host and port. (Always use exclamation marks with IPv6 addresses, which themselves include colons.) Also see &quot;Membership operations&quot; on page 4-95 for information about exporting or importing the membership client configuration file and &quot;Configure a grid as a membership service client&quot; in Oracle TimesTen In-Memory Database Scaleout User's Guide for additional information.</td>
</tr>
<tr>
<td>-internalAddress addr</td>
<td>DNS name or IP address of the local system for internal communications, inside the grid. Use this together with -externalAddress. This option takes one name or address only, and a specified name must resolve to one IP address or to multiple IP addresses on the same network segment. If host names from /etc/hosts are being used, the /etc/hosts files on all instances in the grid must contain identical entries for all hosts in the grid. Also see Notes below and &quot;Address formats&quot; on page 4-6.</td>
</tr>
</tbody>
</table>
## Option Description

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-externalAddress addr</code></td>
<td>DNS name or IP address of the local system for external communications, outside the grid, for client/server connections. Use this together with <code>-internalAddress</code>. This option takes one name or address only, but a name may resolve to one or more IP addresses. If host names from <code>/etc/hosts</code> are being used, the <code>/etc/hosts</code> files on all instances in the grid must contain identical entries for all hosts in the grid. Also see Notes below and &quot;Address formats&quot; on page 4-6.</td>
</tr>
<tr>
<td><code>-address addr</code></td>
<td>DNS name or IP address of the local system for both external and internal communications, if a single address is used. Setting <code>-address xxx</code> is exactly equivalent to setting <code>-internalAddress xxx</code> and <code>-externalAddress xxx</code>. This option takes one name or address only, and a specified name must resolve to one IP address or to multiple IP addresses on the same network segment. If host names from <code>/etc/hosts</code> are being used, the <code>/etc/hosts</code> files on all instances in the grid must contain identical entries for all hosts in the grid. <strong>Note:</strong> Using a single address is not recommended for production environments. Also see Notes below and &quot;Address formats&quot; on page 4-6.</td>
</tr>
<tr>
<td><code>-mgmtPort n</code></td>
<td>Port number used by the initial management instance for replication when management data on the active management instance is replicated. This is required if there will be two management instances. The default is 3754.</td>
</tr>
<tr>
<td><code>-host name</code></td>
<td>Specifies the name that will be given to the host object in the model for the initial host in the grid. If not specified, the first component of the operating system host name is used (the host name up to but not including the first &quot;.&quot;, such as <code>myhost</code>).</td>
</tr>
<tr>
<td><code>-retainDays numdays</code></td>
<td>Specifies that old versions of the model should be retained for <code>numdays</code> days, then automatically deleted. If <code>numdays</code> is 0, then old versions of the model are not automatically deleted based on their age. The default is 30. Also see Notes below.</td>
</tr>
<tr>
<td><code>-retainVersions n</code></td>
<td>Specifies that <code>n</code> old versions of the model should be retained. Anything older than the newest <code>n</code> versions are deleted. If <code>n</code> is 0, then old versions of the model are not automatically deleted based on the number of versions. The default is 10. Also see Notes below.</td>
</tr>
<tr>
<td><code>-warnThresh percent</code></td>
<td>Management instances store metadata for the grid and model. If the metadata on the active management instance fills beyond this percentage of capacity, <code>ttGridAdmin</code> commands result in warnings. The default is 90% full.</td>
</tr>
<tr>
<td><code>-noDataSpaceGroup</code></td>
<td>Specifies that the initial host in the grid is not assigned to a data space group. If this option is not specified, the first host is assigned to data space group 1. Do not set this option if the first host will contain a data instance as well as the management instance.</td>
</tr>
</tbody>
</table>
Examples

% ttGridAdmin gridCreate grid1 -k 2 -membershipConfig
/sw/tten/grid/zkcfg/membership.conf -internalAddress intmysys1.example.com
-externalAddress extmysys1.example.com -host mysys1host
Grid grid1 created

Notes

- You cannot execute this command from an instance that is or has ever been part of another grid.
- You cannot retry gridCreate if it fails. You must remove and recreate the management instance with ttInstanceDestroy and ttInstanceCreate. See "Destroying a grid" and "Creating the initial management instance" in Oracle TimesTen In-Memory Database Scaleout User’s Guide for examples. See "ttInstanceDestroy" on page 5-61 and "ttInstanceCreate" on page 5-59 for reference information.
- Hosts in the grid may be configured with either one or two network addresses, depending on system topology. If configured with two addresses, one is used for communications with systems inside the grid (internal) and one is used for client/server access to databases inside the grid from systems outside the grid (external). If configured with one address, which is not recommended for production environments, it is used for both internal and external communications. You must either set -address or set -internalAddress and -externalAddress.
- You can specify both -retainDays and -retainVersions, in which case old versions of the model are automatically deleted if they are older than numdays days old and there are more than n old versions. If one option is specified as zero, then only the other option takes effect. If both are zero, old versions of the model are not automatically deleted.
- Creating a grid creates version 1 of the grid model.

Display information about the grid (gridDisplay)

Use the gridDisplay command to display information about the grid.

ttGridAdmin gridDisplay

Examples

| Grid name: | grid1 |
| Grid GUID: | 9D049059-1BF2-47E4-AEFA-D3ABA03F609E |
| Created: | 2017-10-30 19:05:47.000000 |
| Major Release: | 18.1 |
| Created Release: | 18.1.2.1.0 |
| K: | 2 |
| Admin Userid: | ttuser1 |
| Admin UID: | 126 |
Get diagnostic information about the grid (gridDump)

The `gridDump` command outputs diagnostic information about the grid to the specified file. This command outputs a very large amount of information and is intended for use by Oracle Support.

```
ttGridAdmin gridDump [filepath]
```

## Options

The `gridDump` command has the option:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>filepath</code></td>
<td>Path and name of the file where diagnostic information is written. If no file is specified, the information is written to <code>stdout</code>.</td>
</tr>
</tbody>
</table>

## Examples

This example outputs to the file `griddumpout`. (When the dump goes to a file, the command has no visible output.)

```
% ttGridAdmin gridDump /sw/tten/grid/misc/griddumpout
```

Collect log information about the grid (gridLogCollect)

The `gridLogCollect` command collects daemon logs and other diagnostic information along with TimesTen configuration files from all instances in the grid. The aggregation of all of this is a collection.

```
ttGridAdmin gridLogCollect -repository reponame [collection]
```

## Options

The `gridLogCollect` command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-repository reponame</code></td>
<td>Name of the repository where the collection of logs, diagnostic information, and configuration files is stored. See &quot;Create a repository (repositoryCreate)&quot; on page 4-109.</td>
</tr>
<tr>
<td><code>collection</code></td>
<td>Name of the collection created to store the logs, diagnostic information, and configuration files. If not specified, the name will be a timestamp in the format <code>Lyyyymmddhhmss</code>.</td>
</tr>
</tbody>
</table>

## Examples

This example creates a repository then creates a collection of logs, diagnostic information, and configuration files in that repository. (See "Create a repository"
Grid operations

(repositoryCreate) on page 4-109 for information about the repositoryCreate command.)

% ttGridAdmin repositoryCreate repocollection -path /repositories
-method scp -address mysys1.example.com
Repository repocollection created

% ttGridAdmin gridLogCollect -repository repocollection mycollection
Logs copied to collection mycollection in repository repocollection

In the repocollection directory, the repository.json file has information about the repository.

The mycollection directory contains logs and configuration files for each instance. (See "Collecting grid logs" in Oracle TimesTen In-Memory Database Scaleout User's Guide for information about the log files.)

Notes

These are automatically included in the collection:

- Contents of the diag directory on each instance (or other diagnostics directory according to the supportlog setting in timesten.conf), such as daemon logs and core files
- TimesTen configuration files from the conf directory on each instance.
- Any .inval and .trace files from the DataStore directory of each element, as specified in the database definition

Modify grid settings (gridModify)

The gridModify command modifies properties of the grid, such as how long previous models of the grid will be retained or how many previous models of the grid will be retained.

ttGridAdmin gridModify [-retainDays numdays]
[-retainVersions n]
[-warnThresh percent]

Options

The gridModify command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-retainDays numdays</td>
<td>Specifies that old versions of the model should be retained for numdays days, then automatically deleted. If numdays is 0, then old versions of the model are not automatically deleted based on their age. The default is 30. Also see Notes below.</td>
</tr>
<tr>
<td>-retainVersions n</td>
<td>Specifies that n old versions of the model should be retained. Anything older than the newest n versions are deleted. If n is 0, then old versions of the model are not automatically deleted based on the number of versions. The default is 10. Also see Notes below.</td>
</tr>
<tr>
<td>-warnThresh percent</td>
<td>Management instances store metadata for the grid and model. If the metadata on the active management instance fills beyond this percentage of capacity, ttGridAdmin commands result in warnings. The default is 90% full.</td>
</tr>
</tbody>
</table>
Examples

This example shows selected output from gridDisplay before and after executing gridModify to change the number of days to retain old versions of the model.

% ttGridAdmin gridDisplay
Grid name: grid1
... Retain Days: 30 Retain Versions: 10 ...

% ttGridAdmin gridModify -retainDays 20
Grid Definition modified.

% ttGridAdmin gridDisplay
Grid name: grid1
... Retain Days: 20 Retain Versions: 10 ...

Notes

You can specify both -retainDays and -retainVersions, in which case old versions of the model are automatically deleted if they are older than numdays days old and there are more than n old versions. If one option is specified as zero, then only the other option takes effect. If both are zero, old versions of the model are not automatically deleted.

Configure SSH (gridSshConfig)

The gridSshConfig command configures a set of TimesTen Scaleout hosts for passwordless SSH connection.

ttGridAdmin gridSshConfig [-internalAddress addr1 [addr2 [addr3...]]]

It prompts for the operating system password of the operating system user executing the command. That user must exist with the same password, UID, and group membership on every host to be configured.

The command can be used either in a TimesTen management instance (with TIMESTEN_HOME set) or outside of any TimesTen instances (where TIMESTEN_HOME is not set), executing ttGridAdmin from the installation bin directory in the latter case. If you use it in a management instance, then a grid and model must already exist.

Users on all of these hosts are configured to enable passwordless SSH connection:

- Any host specified in the -internalAddress option
- If the command is executed in a management instance, any host in the latest version of the grid model
- If the command is executed in a management instance and any repositories have been created or attached in the grid, any host specified in the -address option of a repositoryCreate or repositoryAttach command that was executed with -method scp

After the gridSshConfig command is executed by a user, that user should be able to connect from any host to any other host through SSH without specifying a password. The ttGridAdmin utility will confirm this as part of the execution of the command.
Grid operations

Options

The `gridSshConfig` command has the option:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| `-internalAddress addr1 [addr2 [addr3...]]` | Addresses of hosts to configure for passwordless SSH access.  
Each entry can be an IPv4 address, an IPv6 address, or (typically) a DNS name. Also see "Address formats" on page 4-6.  
This is optional if you run the command in a management instance. |

Examples

This example is run on `mysys1.example.com`, outside of any TimesTen instances, from the installation `bin` directory.

```
% ./ttGridAdmin gridSshConfig -internalAddress mysys1.example.com
mysys2.example.com mysys3.example.com
Enter password:
Setup ssh configuration on local system............................OK
Setup ssh configuration on mysys1.example.com.....................OK
Setup ssh configuration on mysys3.example.com.....................OK
Setup ssh configuration on mysys2.example.com.....................OK
Setup passwordless ssh from local system to mysys1.example.com....OK
Setup passwordless ssh from local system to mysys3.example.com....OK
Setup passwordless ssh from local system to mysys2.example.com....OK
Setup passwordless ssh from mysys1.example.com to mysys1.example.com...OK
Setup passwordless ssh from mysys1.example.com to mysys3.example.com...OK
Setup passwordless ssh from mysys1.example.com to mysys2.example.com...OK
Passwordless ssh working between hosts:

<table>
<thead>
<tr>
<th>From</th>
<th>mysys2.example.com</th>
<th>mysys3.example.com</th>
<th>mysys1.example.com</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>us</em></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>mysys2.example.com</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>mysys3.example.com</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>mysys1.example.com</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
```

This example is run on the same system, but from a TimesTen management instance, executed from the instance `bin` directory. By default, it sets up or confirms passwordless SSH between hosts in the model. The result is the same as in the first example.

Note: You may choose to manually configure passwordless SSH between the hosts of your grid, without using `gridSshConfig`. Doing it manually may allow you more control and more ability to verify that the proper hosts are being configured.
% ttGridAdmin gridSshConfig
Enter password:
Setup ssh configuration on local system.................................OK
Setup ssh configuration on mysys1.example.com.........................OK
Setup ssh configuration on mysys3.example.com.........................OK
Setup ssh configuration on mysys2.example.com.........................OK
Setup passwordless ssh from local system to mysys1.example.com....OK
Setup passwordless ssh from local system to mysys3.example.com....OK
Setup passwordless ssh from local system to mysys2.example.com....OK
Setup passwordless ssh from mysys1.example.com to mysys1.example.com...OK
Setup passwordless ssh from mysys1.example.com to mysys3.example.com...OK
Setup passwordless ssh from mysys1.example.com to mysys2.example.com...OK
Setup passwordless ssh from mysys3.example.com to mysys1.example.com...OK
Setup passwordless ssh from mysys3.example.com to mysys3.example.com...OK
Setup passwordless ssh from mysys3.example.com to mysys2.example.com...OK
Setup passwordless ssh from mysys2.example.com to mysys1.example.com...OK
Setup passwordless ssh from mysys2.example.com to mysys3.example.com...OK
Setup passwordless ssh from mysys2.example.com to mysys2.example.com...OK

Passwordless ssh working between hosts:

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>mysys2.example.com</th>
<th>mysys3.example.com</th>
<th>mysys1.example.com</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>us</em></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>mysys2.example.com</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>mysys3.example.com</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>mysys1.example.com</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>

Notes

- In specifying host addresses, for each host use the same format—fully qualified domain name, simple host name, or IP address—that is used in the -internalAddress or -address option of the hostCreate (or gridCreate) command. For example, do not specify mysys1 for gridSshConfig then mysys1.example.com for hostCreate.

- You can run gridSshConfig multiple times without harm. If you want to enable passwordless SSH on additional hosts later, you can run the command again for those hosts without impacting the hosts already configured.

- In the event of any failure during execution, the command will continue to complete the configuration on as many hosts as it can.

- "Permission denied" errors in the error logs may indicate the password you provided was incorrect or that there is another permissions issue that prevents the command from completing successfully (for example, inappropriate permissions for the user home directory, where the .ssh directory is placed).
Host operations

Use `ttGridAdmin` commands in this section to define a host in the model, modify a host, delete a host, execute commands on all hosts, or list all hosts.

Create a host (hostCreate)

The `hostCreate` command defines a host in the model.

```
hostCreate [name]
  [-address addr]
  [-internalAddress addr]
  [-externalAddress addr]
  [-dataspacegroup n]
  [-nodataspacegroup]
  [-physicalgroup group1 [group2 [group3 [...]]]]
  [-nophysicalgroup]
  [-like name [-cascade]]
  [-comment comment]
```

Options

The `hostCreate` command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>name</code></td>
<td>Specifies the name for the host object in the model. The default is the first component of the operating system host name (the host name up to but not including the first &quot;.&quot;, such as <code>myhost</code>). If this option is omitted, the host system must be accessible through passwordless SSH at the time <code>hostCreate</code> is executed.</td>
</tr>
<tr>
<td><code>-internalAddress addr</code></td>
<td>DNS name or IP address of the host for internal communications, inside the grid. Use this together with <code>-externalAddress</code>. The host must be accessible by passwordless SSH at the specified address. This option takes one name or address only, and a specified name must resolve to one IP address or to multiple IP addresses on the same network segment. If host names from <code>/etc/hosts</code> are being used, the <code>/etc/hosts</code> files on all instances in the grid must contain identical entries for all hosts in the grid. Also see Notes below and &quot;Address formats&quot; on page 4-6.</td>
</tr>
<tr>
<td><code>-externalAddress addr</code></td>
<td>DNS name or IP address of the host for external communications, outside the grid, for client/server connections. Use this together with <code>-internalAddress</code>. The host must be accessible by passwordless SSH at the specified address. This option takes one name or address only, but a name may resolve to one or more IP addresses. If host names from <code>/etc/hosts</code> are being used, the <code>/etc/hosts</code> files on all instances in the grid must contain identical entries for all hosts in the grid. Also see Notes below and &quot;Address formats&quot; on page 4-6.</td>
</tr>
</tbody>
</table>
### Option Description

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-address addr</code></td>
<td>DNS name or IP address of the host for both external and internal communications, if a single address is used. The host must be accessible by passwordless SSH at the specified address. This option takes one name or address only, and a specified name must resolve to one IP address or to multiple IP addresses on the same network segment. If host names from <code>/etc/hosts</code> are being used, the <code>/etc/hosts</code> files on all instances in the grid must contain identical entries for all hosts in the grid. <strong>Note:</strong> Using a single address is not recommended for production environments. Also see Notes below and &quot;Address formats&quot; on page 4-6.</td>
</tr>
<tr>
<td><code>-dataspacegroup n</code></td>
<td>Specifies that this host will belong to data space group number n. The number of data space groups a grid has is determined by the k value set for the grid. A host with a data instance must belong to a data space group. Also see <code>-nodataspacegroup</code> and Notes below. <strong>Note:</strong> Once a host is assigned to a data space group and <code>modelApply</code> is executed, you cannot change the assignment.</td>
</tr>
<tr>
<td><code>-nodataspacegroup</code></td>
<td>Specifies that the host will not be assigned to a data space group. This is the default. A host with a data instance must belong to a data space group. Also see <code>-dataspacegroup</code> and Notes below.</td>
</tr>
<tr>
<td><code>-physicalgroup group1</code></td>
<td>Specifies the set of physical groups that this host will be associated with. It is advisable to spread data instances between different physical groups so that there is redundancy in case of failure. See &quot;Assigning hosts to data space groups&quot; in <em>Oracle TimesTen In-Memory Database Scaleout User's Guide</em>. Also see <code>-nophysicalgroup</code>. <strong>Note:</strong> Physical group assignments are considered by the <code>dataSpaceGroupSuggest</code> command. See Notes below.</td>
</tr>
<tr>
<td><code>-nophysicalgroup</code></td>
<td>Specifies that the host will be associated with no physical groups. This is the default. Also see <code>-physicalgroup</code>.</td>
</tr>
<tr>
<td><code>-like name</code></td>
<td>Specifies that this new host should be created with the same attributes as the named existing host, except where other options that you specify override settings from the existing host. Also see <code>-cascade</code>.</td>
</tr>
<tr>
<td><code>-cascade</code></td>
<td>Use this option with the <code>-like</code> option to specify that installations and instances associated with the <code>-like</code> host also be defined for the host being created. (These objects will be defined for the new host, but not actually created until you run <code>modelApply</code>.)</td>
</tr>
<tr>
<td><code>-comment comment</code></td>
<td>Associates a comment with the host object. Put the comment in quotes if there are any spaces. The comment is stored and included in output of the <code>hostList</code> command.</td>
</tr>
</tbody>
</table>

## Examples

Create a second management instance by adding a new host to the model with a set of installations and instances identical to those on the existing host (specified in the `-like`
option). This command is run from the first management instance, which is on the first host, mysys1host (defined earlier, in the example for “Create a grid (gridCreate)” on page 4-55):

```bash
% ttGridAdmin hostCreate mysys2host -internalAddress intmysys2.example.com -externalAddress extmysys2.example.com -like mysys1host -cascade
Host mysys2host created in Model
Installation installation1 created in Model
Instance gridmgmt1 created in Model
```

This defines gridmgmt1 on mysys2host, duplicating gridmgmt1 on mysys1host.

Create a host for a data instance, specifying the data space group:

```bash
% ttGridAdmin hostCreate mysyshost3 -internalAddress intmysys3.example.com -externalAddress extmysys3.example.com -dataSpaceGroup 1
Host mysyshost3 created in Model
```

### Notes

- In specifying host addresses, for each host use the same format—fully qualified domain name, simple host name, or IP address—as was used in the gridSshConfig command for that host. For example, do not specify mysys1 for hostCreate if mysys1.example.com was specified for gridSshConfig.

- You can use hostModify to change some settings later.

- If you do not assign the host to a data space group during host creation, you can later use the dataSpaceGroupSuggest command to determine optimal assignments based on which physical groups are associated with each host, or you can use the hostModify command to choose a data space group manually.

- Hosts on the grid may be configured with either one or two network addresses, depending on system topology. If configured with two addresses, one is used for communications with systems inside the grid (internal) and one is used for client/server access to databases inside the grid from systems outside the grid (external). If configured with one address, which is not recommended for production environments, it is used for both internal and external communications. You must either set -address or set -internalAddress and -externalAddress.

### Delete a host (hostDelete)

The hostDelete command removes a host from the model.

```bash
ttGridAdmin hostDelete name
    [-cascade]
```

### Options

The hostDelete command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Name of the host object to remove from the model.</td>
</tr>
<tr>
<td>-cascade</td>
<td>Specifies that installation and instance objects associated with the host should also be removed from the model.</td>
</tr>
</tbody>
</table>
Examples

This deletes a host that was created in an example in "Create a host (hostCreate)" on page 4-64.

% ttGridAdmin hostDelete mysys2host -cascade
Instance gridmgmt1 on Host mysys2host deleted from Model
Installation installation1 on Host mysys2host deleted from Model
Host mysys2host deleted from Model

Notes

■ If the host has any installations or instances, you must either use -cascade or use installationDelete and instanceDelete.

■ Deleting instances and installations removes the objects from the model but does not remove the physical instances and installations.

Execute a command or script on grid hosts (hostExec)

The hostExec command executes a command (such as a system command or TimesTen command) or a script on hosts in the grid, as specified.

ttGridAdmin hostExec [-only hostname]
[-exclude hostname]
[-parallel n]
command | -script filepath

Options

The hostExec command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-only hostname</td>
<td>The command or script is executed only on the specified hosts. Specify just one host with -only, but you can use -only multiple times on the command line. Use host names as defined in the model. Without -only or -exclude, the command or script is executed on all hosts in the model.</td>
</tr>
<tr>
<td>-exclude hostname</td>
<td>The command or script is executed on all hosts in the grid except for the specified hosts. Specify just one host with -exclude, but you can use -exclude multiple times on the command line. Use host names as defined in the model. Without -only or -exclude, the command or script is executed on all hosts in the model.</td>
</tr>
<tr>
<td>-parallel n</td>
<td>Specifies that the command or script executes on no more than n hosts simultaneously. The default is 10. A value of 1 results in serial execution.</td>
</tr>
<tr>
<td>command</td>
<td>-script filepath</td>
</tr>
</tbody>
</table>
Examples

This example first shows the existing hosts in the grid, then uses hostExec to run the df / command (to show disk space) on each host, excluding mysys3host and mysys4host. So the command is executed on mysys1host and mysys2host.

% ttGridAdmin hostList
Name        IntAddress             ExtAddress             DSG Comment
----------- ---------------------- ---------------------- --- -------
mysys1host  intmysys1.example.com  extmysys1.example.com    1
mysys2host  intmysys2.example.com  extmysys2.example.com    2
mysys3host  intmysys3.example.com  extmysys3.example.com    1
mysys4host  intmysys4.example.com  extmysys4.example.com    2

% ttGridAdmin hostExec -exclude mysys3host -exclude mysys4host df /
Commands executed on:
   mysys1host rc 0
   mysys2host rc 0
Return code from mysys1host: 0
Output from mysys1host:
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/xvda2           173483816  28416336 136254988  18% /

Return code from mysys2host: 0
Output from mysys2host:
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/xvda2           117144964  35319512  75874836  32% /

Notes

■ The command or script is executed on each host as the instance administrator, through passwordless SSH.

■ No environment variables are set on the hosts, other than those set by SSH by default.

■ The command returns 2000 if execution did not complete prior to the timeout.

■ During execution, stdout and stderr output is displayed as part of the stdout and stderr output from the hostExec command. Because output is buffered, the output from different commands is not intermingled.

List all hosts in the model (hostList)

The hostList command lists information about hosts in the specified version of the model.

```
ttGridAdmin hostList [-latest|-current|-version n]
```

Options

The hostList command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-latest</td>
<td>Lists hosts in the latest model—the model being modified and not yet applied to the grid. This is the default.</td>
</tr>
<tr>
<td>-current</td>
<td>Lists hosts in the current model—the model most recently applied to the grid.</td>
</tr>
<tr>
<td>-version n</td>
<td>Lists hosts in the specified version number of the model.</td>
</tr>
</tbody>
</table>
Examples

The following two examples, relating to examples shown in "Modify a host (hostModify)" on page 4-69 and "List model versions (modelList)" on page 4-102, show identical output, indicating that version 4 is the latest version (the version not yet applied to the model).

For each host, the host name, internal address, external address, and associated data space group are listed (optionally with a comment).

% ttGridAdmin hostlist
Name   IntAddress             ExtAddress             DSG Comment
----------- ---------------------- ------------------     --- --------------------
mysys1host intmysys1.example.com extmysys1.example.com 1
mysys2host intmysys2.example.com extmysys2.example.com 1
mysys3host intmysys3.example.com extmysys3.example.com 1 Move from location1.
mysys4host intmysys4.example.com extmysys4.example.com 2

% ttGridAdmin hostlist -version 4
Name   IntAddress             ExtAddress             DSG Comment
----------- ---------------------- ------------------     --- --------------------
mysys1host intmysys1.example.com extmysys1.example.com 1
mysys2host intmysys2.example.com extmysys2.example.com 1
mysys3host intmysys3.example.com extmysys3.example.com 1 Move from location1.
mysys4host intmysys4.example.com extmysys4.example.com 2

Modify a host (hostModify)

The ttGridAdmin hostModify command modifies a host object in the model.

% ttGridAdmin hostModify name
[-physicalgroup group1 [group2 [group3 [...]]]]
[-addphysicalgroup group1 [group2 [group3 [...]]]]
[-removephysicalgroup group1 [group2 [group3 [...]]]]
[-nophysicalgroup]
[-dataspacegroup n]
[-nodataspacegroup]
[-comment comment]

Options

The hostModify command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Name of the existing host object to modify.</td>
</tr>
<tr>
<td>-physicalgroup group1 [group2 [group3 [...]]]</td>
<td>Specifies a new set of physical groups that this host will be associated with. All physical groups previously associated with the host will be replaced with the specified groups. Also see -addphysicalgroup, -removephysicalgroup, and -nophysicalgroup.</td>
</tr>
<tr>
<td>-addphysicalgroup group1 [group2 [group3 [...]]]</td>
<td>Adds the specified physical groups to the groups the host is associated with. Also see -physicalgroup, -removephysicalgroup, and -nophysicalgroup.</td>
</tr>
<tr>
<td>-removephysicalgroup group1 [group2 [group3 [...]]]</td>
<td>Removes the specified physical groups from the groups the host is associated with. Also see -physicalgroup, -addphysicalgroup, and -nophysicalgroup.</td>
</tr>
</tbody>
</table>
Examples

% ttGridAdmin hostModify mysyshost2 -physicalGroup location3
-physicalGroup location3
-hostModify mysyshost2 -comment "Move from location1."
Host mysyshost2 modified in Model

Notes

- The host system must be accessible through passwordless SSH at the time
  hostModify is executed.
- If modelApply has already been executed for a model including this host, you
  cannot change the data space group assignment.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-nophysicalgroup</td>
<td>Specifies that the host will not be associated with any physical groups, removing any prior associations. Also see -physicalgroup, -addphysicalgroup, and -removephysicalgroup.</td>
</tr>
<tr>
<td>-dataspacegroup n</td>
<td>Specifies the number of the data space group that this host will belong to. The number of data space groups a grid will have is determined by the k value set for the grid. A host with a data instance should always belong to a data space group.</td>
</tr>
<tr>
<td>-nodataspacegroup</td>
<td>Specifies that this host will not be part of any data space group (the default)</td>
</tr>
<tr>
<td>-comment comment</td>
<td>Associates a comment with the host object or modifies an existing comment. Put the comment in quotes if there are any spaces. The comment is stored and included in output of the hostList command.</td>
</tr>
</tbody>
</table>
Import and export operations

Use ttGridAdmin commands in this section to import and export databases, display the status of those operations, or delete an export.

Also see "Migrating, Backing Up and Restoring Data" in Oracle TimesTen In-Memory Database Scaleout User’s Guide.

Export a database (dbExport)

The dbExport command exports data from the specified database into a specified repository. The dbExport and dbImport commands are used, for example, to migrate a database between two grids or between versions of TimesTen that are not patch-compatible. See "Migrating, Backing Up and Restoring Data" in Oracle TimesTen In-Memory Database Scaleout User’s Guide for additional information.

ttGridAdmin dbExport dbname
       -repository reponame
       [-name exportname]

An export is stored as a collection under a repository. You first must create the repository. See "Repository operations" on page 4-108.

Options

The dbExport command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbname</td>
<td>Name of the database to export.</td>
</tr>
<tr>
<td>-repository</td>
<td>reponame</td>
</tr>
<tr>
<td>-name</td>
<td>exportname</td>
</tr>
</tbody>
</table>

Examples

% ttGridAdmin dbExport database1 -repository repol -name exp_db1
...
dbExport exp_db1 started

You can then use dbExportStatus to check progress, as shown in the example in "Display the status of a database export (dbExportStatus)" on page 4-72. The export is finished when each element and the database as a whole are indicated as complete.

Notes

- The export is performed asynchronously. Use the dbExportStatus command to check progress.
- Each replica set of the database is stored as a sub-collection.
- The database must be in a closed state with all connections closed when you run dbExport.
- Only one dbExport command can be run for a database at any given time, and dbExport cannot run concurrently with dbImport.
For disk space requirements, see "Exporting and importing a TimesTen Scaleout database" in Oracle TimesTen In-Memory Database Scaleout User’s Guide.

Delete a database export (dbExportDelete)

The dbExportDelete command deletes the specified database export.

```bash
ttGridAdmin dbExportDelete -repository reponame -name exportname
```

Options

The dbExportDelete command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-repository</td>
<td>Name of the repository where the export is stored.</td>
</tr>
<tr>
<td>-name</td>
<td>Name of the export to delete.</td>
</tr>
</tbody>
</table>

Examples

This example deletes the export created in "Export a database (dbExport)" on page 4-71.

```bash
% ttGridAdmin dbExportDelete -repository repol -name exp_db1
Export exp_db1 deleted
```

Notes

This command is typically used to delete old or failed exports.

Display the status of a database export (dbExportStatus)

The dbExportStatus command shows the status of a database export or exports previously started.

```bash
ttGridAdmin dbExportStatus dbname [-name exportname]
```

Options

The dbExportStatus command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbname</td>
<td>Name of the database being exported.</td>
</tr>
<tr>
<td>-name</td>
<td>Name of the export to check. The default is all exports of the specified database.</td>
</tr>
</tbody>
</table>

Examples

This example shows status upon completion of the export from the example in "Export a database (dbExport)" on page 4-71. (That is the only export for database1 in the repository.)

```bash
% ttGridAdmin dbExportStatus database1
```

<table>
<thead>
<tr>
<th>Database</th>
<th>Export</th>
<th>Repository</th>
<th>Host</th>
<th>Instance</th>
<th>Elem</th>
<th>State</th>
<th>Started</th>
</tr>
</thead>
<tbody>
<tr>
<td>database1</td>
<td>exp_db1</td>
<td>repol</td>
<td>host3</td>
<td>instance1</td>
<td></td>
<td>Completed 2017-03-02T14:42:24.0002</td>
<td></td>
</tr>
</tbody>
</table>

4-72 Oracle TimesTen In-Memory Database Reference
host4 instance1  2 Complete
host5 instance1  3 Complete

Notes

When you believe the export is complete, confirm that dbExportStatus shows Complete for the export as a whole and for every instance. If there were any failures, see "Check the status of a database export" in Oracle TimesTen In-Memory Database Scaleout User’s Guide.

Import a database (dbImport)

The dbImport command imports data from a specified previous export into the specified database. The dbExport and dbImport commands are used, for example, to migrate a database between two grids or between versions of TimesTen that are not patch-compatible.

ttGridAdmin dbImport dbname
   -repository reponame
   -name exportname
   [-ckptFreq mb]
   [-updateStats]
   [-estimateStats pct]

Options

The dbImport command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbname</td>
<td>Name of the database where the data is to be imported.</td>
</tr>
<tr>
<td>-repository reponame</td>
<td>Name of the repository where the export is located.</td>
</tr>
<tr>
<td>-name exportname</td>
<td>Name of the export to use for the import.</td>
</tr>
<tr>
<td>-ckptFreq mb</td>
<td>Checkpoint frequency, in terms of how many megabytes have been imported. A checkpoint is written each time that many megabytes have been imported. The default is to write no checkpoints during the import.</td>
</tr>
<tr>
<td>-updateStats</td>
<td>Update statistics on each table as it is imported.</td>
</tr>
<tr>
<td></td>
<td>Also see Notes below.</td>
</tr>
<tr>
<td>-estimateStats pct</td>
<td>Estimate statistics on each table as it is imported, by reading the specified percentage of rows of each table.</td>
</tr>
<tr>
<td></td>
<td>Also see Notes below.</td>
</tr>
</tbody>
</table>

Examples

This example imports the export created in the example in "Export a database (dbExport)" on page 4-71, into a database imp_db1.

% ttGridAdmin dbImport imp_db1 -repository repol -name exp_db1
   dbImport exp_db1 started

You can then use dbImportStatus to check progress, as shown in the example in "Display the status of a database import (dbImportStatus)" on page 4-74. The import is finished when each element and the database as a whole are indicated as complete.
Notes

- The database must already be created and loaded and must have a distribution map, but must be closed, with all connections closed, when you run `dbImport`.
- If you specify both `-estimateStats` and `-updateStats`, statistics on imported tables are updated, not estimated.
- The import is performed asynchronously. Use the `dbImportStatus` command to check progress.
- Only one `dbImport` command can run for a database at any given time, and `dbImport` cannot run concurrently with `dbExport`.
- Functionality of the `-ckptFreq`, `-updateStats`, and `-estimateStats` options is the same as for equivalent options of the `ttMigrate` utility. See "ttMigrate" on page 5-97.
- For disk space requirements, see "Exporting and importing a TimesTen Scaleout database" in Oracle TimesTen In-Memory Database Scaleout User’s Guide.

Display the status of a database import (dbImportStatus)

The `dbImportStatus` command shows the status of a database import previously started.

```
ttGridAdmin dbImportStatus dbname [-name exportname]
```

Options

The `dbImportStatus` command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>dbname</code></td>
<td>Name of the database where the import is being checked.</td>
</tr>
<tr>
<td><code>-name exportname</code></td>
<td>Name of the export from which the data is being imported. You can use this option in the atypical scenario where there are multiple imports into the same database (otherwise, the status of all the imports would be shown).</td>
</tr>
</tbody>
</table>

Examples

This example shows status upon completion of the import from the example in "Import a database (dbImport)" on page 4-73.

```
% ttGridAdmin dbImportStatus imp_db1 -name exp_db1
Database       Import   Repository Host  Instance  Elem  State                  Started
------------  --------  ---------- ----  ------ ----- --------------------- ------------------------
imp_db1       exp_db1   repol          1     3    Import_Finale_Complete 2016-07-25T17:53:27.000Z
host1 instance1     1 Import_Rows_Complete
host3 instance1     3 Import_Rows_Complete
```

Notes

When you believe the import is complete, confirm that `dbImportStatus` shows **Complete** for the import as a whole and for every instance. If there were any failures, see "Check the status of a database import" in Oracle TimesTen In-Memory Database Scaleout User’s Guide.
Installation operations

Use ttGridAdmin commands in this section to define a TimesTen installation in the model, list all installations in the grid, show status of all installations, delete an installation, or execute a command on all installations.

Create an installation (installationCreate)

The installationCreate command defines a TimesTen installation in the model.

```
ttGridAdmin installationCreate hostname[.installname]
    -location path
    [-source where]
    [-comment comment]
```

Options

The installationCreate command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hostname[.installname]</td>
<td>The hostname is the name of the host where the installation is to be created, optionally with a specified installname for the name of the installation in the model. The default is installation1.</td>
</tr>
<tr>
<td>-location path</td>
<td>Path, on the specified host, to the directory where the installation is to be created. The specified directory does not have to exist, but if it exists it must be empty.</td>
</tr>
</tbody>
</table>
| -source where        | Location that the installation will be copied from. The location does not have to be on a system that is part of the grid. You can specify it in any of the following formats, as applicable:  

```
/path
address:/path
address!/path
[address]!/:/path
```

If path is a directory, it must be the top-level tt18.1.2.1.0 directory of an existing TimesTen installation. If it is a file, it must be a .zip file that expands into a TimesTen installation. The address is a DNS name or IP address.

If address is specified, passwordless SSH is used to fetch the installation source from the system with that address. You must use the fourth format if there is a colon in the address itself, such as for IPv6 addresses.

The default is the location of the installation associated with the active management instance, from which ttGridAdmin is executed.

Also see "Address formats" on page 4-6.

| -comment comment     | Associates a comment with the installation object. Put the comment in quotes if there are any spaces. The comment is stored and included in output of the installationList command. |

Examples

Create an installation for host mysys4host, using the default source location. (This example was run from mysys1.)

```
% ttGridAdmin installationCreate mysys4host.installcreate
```
Installation operations

```
-location /sw/tten/grid/ttinstls/installcreate
Installation installcreate on Host mysys4host created in Model

This time, specify a source location:
% ttGridAdmin installationCreate mysys4host.installcreate2
-location /sw/tten/grid/ttinstls/installcreate2 -source
mysys1:/sw/tten/grid/ttinstls/myinstl/tt18.1.2.1.0
Installation installcreate2 on Host mysys4host created in Model
```

**Notes**

- This command does not create a physical installation. It defines an installation
  object in the model. (The `modelApply` command creates the installation.)
- Multiple installation objects for the same TimesTen release can point to the same
  physical installation; however, you cannot specify the same location on the same
  host for installations from different releases.

**Delete an installation (installationDelete)**

The `installationDelete` command deletes an installation from the model. It deletes the
specified installation (or the only installation, as applicable) on the specified host.

```
ttGridAdmin installationDelete hostname[.installname]
```

**Options**

The `installationDelete` command has the option:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hostname[.installname]</td>
<td>The <code>hostname</code> is the name of the host where the installation is to be deleted. The <code>installname</code> is the name of the installation to be deleted and is required only if there is more than one installation on the host.</td>
</tr>
</tbody>
</table>

**Examples**

In this example, `installcreate2` is the only installation on the host.

```
% ttGridAdmin installationDelete mysys4host
Installation installcreate2 on Host mysys4host deleted from Model
```

**Notes**

- You cannot remove an installation that is still used by instances on the specified host.
- This command removes the installation object from the model but does not remove the physical installation. Remove the files manually when you are certain they are no longer used.

**Execute a command or script on grid installations (installationExec)**

The `installationExec` command executes a command (such as a system command or TimesTen command) or a script on installations in the grid, as specified.

```
ttGridAdmin installationExec [-only hostname[.installname]]
[-exclude hostname[.installname]]
[-parallel n]
command | -script filepath
```
As the command or script executes, the `TIMESTEN_INSTALL` environment variable is set to contain the fully qualified path name of the installation as defined in the model.

**Options**

The `installtionExec` command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-only hostname[.installname]</code></td>
<td>The command or script is executed only on the specified installations. Specify just one installation with <code>-only</code>, but you can use <code>-only</code> multiple times on the command line. Use host names and installation names as defined in the model. You do not have to include the installation name if it is the only installation on the host.</td>
</tr>
<tr>
<td><code>-exclude hostname[.installname]</code></td>
<td>The command or script is executed on all installations in the grid except for those specified. Specify just one installation with <code>-exclude</code>, but you can use <code>-exclude</code> multiple times on the command line. Use host names and installation names as defined in the model. You do not have to include the installation name if it is the only installation on the host.</td>
</tr>
<tr>
<td><code>-parallel n</code></td>
<td>Specifies that the command or script executes on no more than <code>n</code> installations simultaneously. The default is 10. A value of 1 results in serial execution.</td>
</tr>
<tr>
<td>`command</td>
<td>-script filepath`</td>
</tr>
</tbody>
</table>

**Examples**

This example checks the disk space usage on the file system of each installation.

```
% ttGridAdmin installationExec df `"$TIMESTEN_INSTALL"
Commands executed on:
  mysys2host.installation1 rc 0
  mysys4host.installalladc rc 0
  mysys1host.installation1 rc 0
  mysys3host.installslc rc 0
Return code from mysys2host.installation1: 0
Output from mysys2host.installation1:
Filesystem   1K-blocks   Used   Available   Use%   Mounted on
/dev/xvda2   117144964 42660228  68534120   39%   /
Return code from mysys4host.installalladc: 0
Output from mysys4host.installalladc:
Filesystem   1K-blocks   Used   Available   Use%   Mounted on
/dev/xvda2   117144964 42660228  68534120   39%   /
Return code from mysys1host1.installation1: 0
Output from mysys1host1.installation1:
Filesystem   1K-blocks   Used   Available   Use%   Mounted on
/dev/xvda2   173483816 57971312 106700012   36%   /
Return code from mysys3host.installslc: 0
Output from mysys3host.installslc:
Filesystem   1K-blocks   Used   Available   Use%   Mounted on
/dev/xvda2   173483816 57971312 106700012   36%   /
```
Notes

- The command or script is executed as the instance administrator on each installation, through passwordless SSH.
- The command returns 2000 if execution did not complete prior to the timeout.
- During execution, stdout and stderr output is displayed as part of the stdout and stderr output from the installationExec command. Because output is buffered, the output from different commands is not intermingled.

List installations (installationList)

The installationList command lists all TimesTen installations in the model.

```
ttGridAdmin installationList [-latest|-current|-version n] [-instance]
```

Options

The installationList command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-latest</td>
<td>List the installations in the latest (in progress) model, which has not yet been applied to the grid. This is the default.</td>
</tr>
<tr>
<td>-current</td>
<td>List the installations in the current model—the model currently applied to the grid.</td>
</tr>
<tr>
<td>-version n</td>
<td>List the installations in the specified version number of the model.</td>
</tr>
<tr>
<td>-instance</td>
<td>Show the instances that are using each installation. Installations not yet associated with an instance are not displayed.</td>
</tr>
</tbody>
</table>

Examples

This example lists the installations in the latest (in progress) model.

```
% ttGridAdmin installationList
```

<table>
<thead>
<tr>
<th>Host</th>
<th>Install</th>
<th>Location</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>mysys1host</td>
<td>installation1</td>
<td>/sw/tten/grid/ttinstls/myinstl/tt18.1.2.1.0/</td>
<td></td>
</tr>
<tr>
<td>mysys2host</td>
<td>installation1</td>
<td>/sw/tten/grid/ttinstls/myinstl/tt18.1.2.1.0/</td>
<td></td>
</tr>
<tr>
<td>mysys3host</td>
<td>installcreate1</td>
<td>/sw/tten/grid/ttinstls/installcreate1/</td>
<td></td>
</tr>
<tr>
<td>mysys4host</td>
<td>installcreate1</td>
<td>/sw/tten/grid/ttinstls/installcreate1/</td>
<td></td>
</tr>
</tbody>
</table>

This example lists installations in the latest model that are associated with an instance.

```
% ttGridAdmin installationList -instance
```

<table>
<thead>
<tr>
<th>Host</th>
<th>Install</th>
<th>Instance</th>
<th>Location</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>mysys1host</td>
<td>installation1</td>
<td>gridmgmt1</td>
<td>/sw/tten/grid/ttinstls/myinstl/tt18.1.2.1.0/</td>
<td></td>
</tr>
<tr>
<td>mysys2host</td>
<td>installation1</td>
<td>gridmgmt1</td>
<td>/sw/tten/grid/ttinstls/myinstl/tt18.1.2.1.0/</td>
<td></td>
</tr>
</tbody>
</table>

Display status of installations (installationStatus)

The installationStatus command shows the status of all installations that are associated with the grid. This is status of the physical installations, not status of installations in the model.

```
ttGridAdmin installationStatus
```
Examples

% ttGridAdmin installationStatus

<table>
<thead>
<tr>
<th>Host</th>
<th>Install</th>
<th>Usable</th>
<th>DelPend</th>
<th>Message</th>
<th>When</th>
</tr>
</thead>
<tbody>
<tr>
<td>mysys1host</td>
<td>installation1</td>
<td>Yes</td>
<td>N</td>
<td>2016-11-01 14:49:31</td>
<td></td>
</tr>
<tr>
<td>mysys2host</td>
<td>installation1</td>
<td>Yes</td>
<td>N</td>
<td>2016-11-01 14:49:31</td>
<td></td>
</tr>
<tr>
<td>mysys3host</td>
<td>installcreate2slc</td>
<td>Yes</td>
<td>N</td>
<td>2016-11-01 14:49:31</td>
<td></td>
</tr>
<tr>
<td>mysys4host</td>
<td>installcreate2adc</td>
<td>Yes</td>
<td>N</td>
<td>2016-11-01 14:49:31</td>
<td></td>
</tr>
</tbody>
</table>

The DelPend entry indicates whether a deletion is pending, where installationDelete was executed but the updated model has not yet been applied to remove the physical installation.
Instance operations

Use ttGridAdmin commands in this section to define a TimesTen Scaleout instance in the model, modify an instance, delete an instance, list instances in the grid, display status of instances in the grid, import or export an instance configuration file, or execute a command on instances in the grid.

Export instance configuration attributes (instanceConfigExport)

The instanceConfigExport command exports configuration attribute settings, previously imported using instanceConfigImport, from the specified version of the model.

`ttGridAdmin instanceConfigExport [-latest|-current|-version n] [filepath]`

Options

The instanceConfigExport command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-latest</td>
<td>Export configuration attribute settings that were imported into the latest model, which has not yet been applied to the grid. This is the default.</td>
</tr>
<tr>
<td>-current</td>
<td>Export configuration attribute settings that were imported into the current model—the model currently applied to the grid.</td>
</tr>
<tr>
<td>-version n</td>
<td>Export configuration attribute settings that were imported into the specified version of the model.</td>
</tr>
<tr>
<td>filepath</td>
<td>The path and name of the file to export configuration attribute settings into. If no file is specified, the information is written to stdout.</td>
</tr>
</tbody>
</table>

Examples

This example exports a configuration attribute setting from the current version of the model and from the latest (default) version of the model after the imports shown in the next section, “Import instance configuration attributes (instanceConfigImport)” on page 4-80. Contents of the export files are also shown.

```
% ttGridAdmin instanceConfigExport -current /tmp/instanceconfigexp1
% more /tmp/instanceconfigexp1
max_conns_per_server=500
% ttGridAdmin instanceConfigExport /tmp/instanceconfigexp2
% more /tmp/instanceconfigexp2
max_conns_per_server=1000
```

Notes

This command exports only settings that were previously imported, not any other settings from the timesten.conf files.

Import instance configuration attributes (instanceConfigImport)

The instanceConfigImport command imports configuration attribute settings into the latest version of the model, to be used by every instance in the grid.

`ttGridAdmin instanceConfigImport [filepath]`
After you execute `modelApply`, the configuration file for each instance is updated to include the imported attributes. You must restart the TimesTen daemon on each instance for the changes to take effect.

See Notes below for a list of attributes you cannot import.

### Options

The `instanceConfigImport` command has the option:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>filepath</td>
<td>The path and name of the file to import configuration attribute settings from. If no file is specified, the information is read from <code>stdin</code>.</td>
</tr>
</tbody>
</table>

### Examples

Import from this file.

```bash
% more /tmp/instanceconfigimpl
# Set maximum number of connections.
max_conns_per_server=500
% ttGridAdmin instanceConfigImport /tmp/instanceconfigimpl
Instance configuration file /tmp/instanceconfigimpl imported
```

Apply the model (output is not shown):

```bash
% ttGridAdmin modelApply
...
ttGridAdmin modelApply complete
```

Now import from this file:

```bash
% more /tmp/instanceconfigimp2
# Set maximum number of connections.
max_conns_per_server=1000
% ttGridAdmin instanceConfigImport /tmp/instanceconfigimp2
Instance configuration file /tmp/instanceconfigimp2 imported
```

After these steps, the latest version of the model will have a maximum connections setting of 500 and the current version of the model will have a setting of 1000. This is shown in the examples in the previous section, "Export instance configuration attributes (instanceConfigExport)" on page 4-80.

### Notes

- As shown in the example, each entry that is imported is of the form `name=value`. You can also include comments, indicated by `#`.
- The `timesten.conf` files are updated when you execute `modelApply`.
- The following attributes are set automatically when the `modelApply` command creates or configures instances and cannot be imported:
  
  - `admin_uid`
  - `admin_user`
  - `client_only`
  - `daemon_port`
  - `grid_external_addr`
  - `grid_guid`
  - `grid_host`
Refer to Chapter 1, "TimesTen Instance Configuration File" for information about TimesTen configuration attributes.

Create an instance (instanceCreate)

The instanceCreate command defines an instance in the model.

```
    ttGridAdmin instanceCreate hostname[.instancename]    
       -location path                              
       [-type management|data]                    
       [-installation name]                       
       [-daemonport n]                            
       [-csport n]                                 
       [-mgmtport n]                               
       [-comment comment]                         
       [-walletDir path]
```

Options

The instanceCreate command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hostname[.instancename]</td>
<td>The hostname is the name of the host where the instance is to be created, optionally with a specified instancename for the name of the instance in the model. The default is instance1.</td>
</tr>
<tr>
<td>-location path</td>
<td>Path, on the specified host, to the directory where the instance is to be created. The specified directory does not have to exist.</td>
</tr>
<tr>
<td>-type management</td>
<td>data</td>
</tr>
<tr>
<td>-installation name</td>
<td>Name of the installation that the instance will use. This option is not necessary if there is only one installation on the host.</td>
</tr>
<tr>
<td>-daemonport n</td>
<td>Port number where the TimesTen main daemon for the instance will listen. The default is 6624.</td>
</tr>
<tr>
<td></td>
<td><strong>Important</strong>: If you create more than one instance on a system (such as a management instance and a data instance), you must specify unique port numbers.</td>
</tr>
<tr>
<td>-csport n</td>
<td>Port number where the server for TimesTen client/server will listen. The default is 6625.</td>
</tr>
<tr>
<td></td>
<td><strong>Important</strong>: If you create more than one instance on a system (such as a management instance and a data instance), you must specify unique port numbers.</td>
</tr>
<tr>
<td>-mgmtport n</td>
<td>For management instances, the port number that will be used by replication when management data on the active management instance is replicated. The default is 3754.</td>
</tr>
</tbody>
</table>
Examples

% ttGridAdmin instanceCreate mysys3host.griddata1
   -location /sw/tten/grid/ttinstances -daemonPort 20000 -csPort 21000
Instance griddata1 on Host mysys3host created in Model

Notes

■ This command does not create a physical instance. It defines an instance object in
  the model. The modelApply command creates the physical instance.

■ Be aware of these prerequisites:
  – The host must have an associated installation object. Use the
    installationCreate command.
  – For a data instance, the host must be in a data space group. If that is not the
    case, the physical instance cannot be created when you apply the model.
    You can use the hostList command to confirm whether a host is in a data
    space group, and the hostModify command to assign a data space group if
    needed.

■ The timesten_home directory will be location/name. In the example, where the
  location is /sw/tten/ttinstances and the instance name is griddata1, timesten_
  home will be /sw/tten/ttinstances/griddata1.

■ Some instance settings can be changed later through the instanceModify
  command, as desired.

Delete an instance (instanceDelete)

The instanceDelete command deletes an instance from the model.

ttGridAdmin instanceDelete hostname[.instancename]

Options

The instanceDelete command has the option:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hostname[.instancename]</td>
<td>The hostname is the name of the host where the instance is to be deleted. The instancename is the name of the instance to be deleted and is required only if there is more than one instance on the host.</td>
</tr>
</tbody>
</table>

Examples

In this example, griddata1 is the only instance on the host.
Instance operations

% ttGridAdmin instanceDelete mysys3host
Instance griddatal on Host mysys3host deleted from Model

Notes

- This command first stops the instance if it has not already been stopped.
- The command removes the instance object from the model. It does not remove the physical instance. (The modelApply command removes the instance.)
- You cannot remove an instance that is still used by other objects in the model.
- You cannot remove an instance that contains a database element.

Execute a command or script on grid instances (instanceExec)

The instanceExec command executes a command (such as a system command or TimesTen command) or a script on instances in the grid, as specified.

```
ttGridAdmin instanceExec [-only hostname[.instancename]]
  [-exclude hostname[.instancename]]
  [-parallel n]
  [-type all|management|data]
  [-up] command | -script filepath
```

Options

The instanceExec command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-only hostname[.instancename]</code></td>
<td>The command or script is executed only on the specified instances. Specify just one instance with <code>-only</code>, but you can use <code>-only</code> multiple times on the command line. Use host names and instance names as defined in the model. You do not have to include the instance name if it is the only instance on the host.</td>
</tr>
<tr>
<td><code>-exclude hostname[.instancename]</code></td>
<td>The command or script is executed on all instances in the grid except for those specified. Specify just one instance with <code>-exclude</code>, but you can use <code>-exclude</code> multiple times on the command line. Use host names and instance names as defined in the model. You do not have to include the instance name if it is the only instance on the host.</td>
</tr>
<tr>
<td><code>-parallel n</code></td>
<td>Specifies that the command or script executes on no more than <code>n</code> instances simultaneously. The default is 10. A value of 1 results in serial execution.</td>
</tr>
<tr>
<td>`-type all</td>
<td>management</td>
</tr>
<tr>
<td><code>-up</code></td>
<td>Specifies that the command or script is executed only on instances that are part of the current membership. The default is to execute commands on all instances (whether they are running or not).</td>
</tr>
</tbody>
</table>
Examples

On each data instance, this example creates directories databases and logs under /data (with no error if the directories already exist).

% ttGridAdmin instanceExec -type data mkdir -p /data/{databases,logs}
Overall return code: 0
Commands executed on:
  mysys6host.griddata4 rc 0
  mysys5host.griddata3 rc 0
  mysys3host.griddata1 rc 0
  mysys4host.griddata2 rc 0
Return code from mysys6host.griddata4: 0
Output from mysys6host.griddata4:
Return code from mysys5host.griddata3: 0
Output from mysys5host.griddata3:
Return code from mysys3host.griddata1: 0
Output from mysys3host.griddata1:
Return code from mysys4host.griddata2: 0
Output from mysys4host.griddata2:

This example starts the TimesTen daemon on mysys5host.griddata3 (useful, for example, if the element on that instance went down).

% ttGridAdmin instanceExec -only mysys5host.griddata3 ttDaemonAdmin -start
Overall return code: 0
Commands executed on:
  mysys5host.griddata3 rc 0
Return code from mysys5host.griddata3: 0
Output from mysys5host.griddata3:
TimesTen Daemon (PID: 7586, port: 6624) startup OK.

For each data instance, this example runs the ttIsql monitor command then exits ttIsql. (Only selected portions of the ttIsql connection output and monitoring output are shown.)

% ttGridAdmin instanceExec -type data 'ttIsql -e "monitor;quit" -dsn database1'
Overall return code: 0
Commands executed on:
  mysys4host.griddata2 rc 0
  mysys5host.griddata3 rc 0
  mysys6host.griddata4 rc 0
  mysys3host.griddata1 rc 0
Return code from mysys4host.griddata2: 0
Output from mysys4host.griddata2:

TimesTen Daemon (PID: 7586, port: 6624) startup OK.

Copyright (c) 1996, 2018, Oracle and/or its affiliates. All rights reserved.
Type ? or 'help' for help, type 'exit' to quit ttIsql.

connect 'DSN=database1';
Connection successful: DSN=database1;...
monitor;

TIME_OF_1ST_CONNECT: Fri Aug 3 13:47:42 2018
...
PERM_ALLOCATED_SIZE: 262144
PERM_IN_USE_SIZE: 29997
PERM_IN_USE_HIGH_WATER: 29997
TEMP_ALLOCATED_SIZE: 131072
TEMP_IN_USE_SIZE: 19146
TEMP_IN_USE_HIGH_WATER: 22352
...

quit;
Disconnecting...
Done.
Return code from mysys5host.griddata3: 0
Output from mysys5host.griddata3:

Copyright (c) 1996, 2018, Oracle and/or its affiliates. All rights reserved.
Type ? or "help" for help, type "exit" to quit ttIsql.

connect 'DSN=database1';
Connection successful: DSN=database1;...

monitor;

TIME_OF_1ST_CONNECT: Fri Aug 3 13:47:41 2018
...
PERM_ALLOCATED_SIZE: 262144
PERM_IN_USE_SIZE: 29916
PERM_IN_USE_HIGH_WATER: 29932
TEMP_ALLOCATED_SIZE: 131072
TEMP_IN_USE_SIZE: 19613
TEMP_IN_USE_HIGH_WATER: 22819
...

quit;
Disconnecting...
Done.
Return code from mysys6host.griddata4: 0
Output from mysys6host.griddata4:

Copyright (c) 1996, 2018, Oracle and/or its affiliates. All rights reserved.
Type ? or "help" for help, type "exit" to quit ttIsql.

connect 'DSN=database1';
Connection successful: DSN=database1;...

monitor;

TIME_OF_1ST_CONNECT: Fri Aug 3 13:47:41 2018
...
PERM_ALLOCATED_SIZE: 262144
PERM_IN_USE_SIZE: 29981
PERM_IN_USE_HIGH_WATER: 29981
TEMP_ALLOCATED_SIZE: 131072
TEMP_IN_USE_SIZE: 19344
TEMP_IN_USE_HIGH_WATER: 22550
...
Notes

- The command or script is executed as the instance administrator on each instance, through passwordless SSH.
- Environment variables (such as `TIMESTEN_HOME`, `CLASSPATH`, `PATH`, and `LD_LIBRARY_PATH`) are set appropriately for each instance.
- The command returns 2000 if execution did not complete prior to the timeout.
- During execution, `stdout` and `stderr` output is displayed as part of the `stdout` and `stderr` output from the `instanceExec` command. Because output is buffered, the output from different commands is not intermingled.

List instances (instanceList)

The `instanceList` command lists information about instances in the specified version of the model.

```
 ttGridAdmin instanceList [-latest|-current|-version n] [-type all|management|data] [-install]
```

Options

The `instanceList` command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-latest</code></td>
<td>Lists instances in the latest model—the model being modified and not yet applied to the grid. This is the default.</td>
</tr>
<tr>
<td><code>-current</code></td>
<td>Lists instances in the current model—the model most recently applied to the grid.</td>
</tr>
</tbody>
</table>
Examples

This example is for a grid with two hosts on each of two systems. On each system, one host has a management instance and one has a data instance. By default, data instances as well as management instances are listed in the latest model (in the process of being modified and not yet applied).

% ttGridAdmin instanceList

<table>
<thead>
<tr>
<th>Host</th>
<th>Instance</th>
<th>Type</th>
<th>Instance Home</th>
<th>Port</th>
<th>CSPort</th>
<th>MgmtPort</th>
</tr>
</thead>
<tbody>
<tr>
<td>mysys1host</td>
<td>gridmgmt1</td>
<td>Mgmt</td>
<td>/sw/tten/grid/ttinstances/gridmgmt1/</td>
<td>10000</td>
<td>11000</td>
<td>3754</td>
</tr>
<tr>
<td>mysys2host</td>
<td>gridmgmt1</td>
<td>Mgmt</td>
<td>/sw/tten/grid/ttinstances/gridmgmt1/</td>
<td>10000</td>
<td>11000</td>
<td>3754</td>
</tr>
<tr>
<td>mysys3host</td>
<td>griddata1</td>
<td>Data</td>
<td>/sw/tten/grid/ttinstances/griddata1/</td>
<td>20000</td>
<td>21000</td>
<td></td>
</tr>
<tr>
<td>mysys4host</td>
<td>griddata2</td>
<td>Data</td>
<td>/sw/tten/grid/ttinstances/griddata2/</td>
<td>20000</td>
<td>21000</td>
<td></td>
</tr>
</tbody>
</table>

This example also shows the associated installation objects (the Comment column is omitted):

% ttGridAdmin -instanceList -install

<table>
<thead>
<tr>
<th>Host</th>
<th>Instance</th>
<th>Installation</th>
<th>Type</th>
<th>Instance Home</th>
<th>Port</th>
<th>CSPort</th>
<th>MgmtPort</th>
</tr>
</thead>
<tbody>
<tr>
<td>mysys1host</td>
<td>gridmgmt1</td>
<td>installation1</td>
<td>Mgmt</td>
<td>/sw/tten/grid/ttinstances/gridmgmt1/</td>
<td>10000</td>
<td>11000</td>
<td>3754</td>
</tr>
<tr>
<td>mysys2host</td>
<td>gridmgmt1</td>
<td>installation1</td>
<td>Mgmt</td>
<td>/sw/tten/grid/ttinstances/gridmgmt1/</td>
<td>10000</td>
<td>11000</td>
<td>3754</td>
</tr>
<tr>
<td>mysys3host</td>
<td>griddata1</td>
<td>installation1</td>
<td>Data</td>
<td>/sw/tten/grid/ttinstances/griddata1/</td>
<td>20000</td>
<td>21000</td>
<td></td>
</tr>
<tr>
<td>mysys4host</td>
<td>griddata2</td>
<td>installation1</td>
<td>Data</td>
<td>/sw/tten/grid/ttinstances/griddata2/</td>
<td>20000</td>
<td>21000</td>
<td></td>
</tr>
</tbody>
</table>

Modify an instance (instanceModify)

The instanceModify command modifies an existing instance object in the model.

`ttGridAdmin instanceModify hostname[.instancename]`

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hostname[.instancename]</td>
<td>The hostname is the name of the host where the instance is to be modified. The instancename is the name of the instance to be modified and is required only if there is more than one instance on the host.</td>
</tr>
<tr>
<td>-installation name</td>
<td>Associates the instance with a different installation on the host, specified by the name of the installation in the model.</td>
</tr>
</tbody>
</table>

Options
Examples

In this example, griddata1 is the only instance on the host.

% ttGridAdmin instanceModify mysys3host -installation altinstall -comment
  Change_from_installcreate1
Instance griddata1 on Host mysys3host modified in Model

(Note that if you have a multi-word comment, you can use underscores instead of spaces to avoid having to put the comment in quotes.)

Notes

- This command is most typically used to patch or upgrade your version of TimesTen by pointing to an installation of the desired release.

- When instanceModify updates are applied by a subsequent modelApply command, the instance is not stopped and reconfigured at that time. Instead, the next time the instance is started, TimesTen Scaleout will detect that the instance configuration does not match the model, and will reconfigure it appropriately.

Display status of instances (instanceStatus)

The instanceStatus command displays information about the status of instances in the grid, in JSON format.

ttGridAdmin instanceStatus [-type all|management|data]

Options

The instanceStatus command has the option:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-type all</td>
<td>management</td>
</tr>
</tbody>
</table>
Management instance operations

Use `ttGridAdmin` commands in this section to start, stop, switch, examine, or check status of the management instance or instances. Execute the commands from the appropriate management instance.

---

**Note:** Typically, there are two management instances, the active and standby. Before you can perform any grid management functions, a management instance must be started as the active instance, from which you can run `ttGridAdmin`. (Initially, the instance from which you create the grid becomes the active management instance.)


---

Start the active management instance (mgmtActiveStart)

The `mgmtActiveStart` command starts the current management instance (from which the command is run) as the active management instance.

```
  ttGridAdmin mgmtActiveStart
```

**Examples**

```
  % ttGridAdmin mgmtActiveStart
  This management instance is now the active
```

**Notes**

- The current management instance must previously be stopped.
- There cannot be another management instance that has been started as the active instance.

Stop the active management instance (mgmtActiveStop)

The `mgmtActiveStop` command stops the active management instance.

```
  ttGridAdmin mgmtActiveStop
```

This command is typically used as the last step in shutting down a grid. Otherwise, if there are two management instances, it is recommended to instead use `mgmtActiveSwitch`.

**Examples**

```
  % ttGridAdmin mgmtActiveStop
  Active management instance stopped
```

**Notes**

- If this command is used in a grid with two management instances (not recommended unless you are shutting down the grid), it can be run from either the active or the standby management instance. Nothing is done automatically to
then promote the standby management instance to active. (See mgmtActiveSwitch.)

- If data instances are running, then the database elements currently loaded in them will continue to operate.
- You cannot perform any management operations until you restart the active management instance.
- If data instances have stopped or failed, they cannot be restarted until you restart the active management instance.

Switch the active management instance (mgmtActiveSwitch)

The mgmtActiveSwitch command, executed from the current standby management instance, results in that instance becoming the active management instance. The original active management instance is stopped if it can be reached.

```
ttGridAdmin mgmtActiveSwitch [-force]
```

Options

The mgmtActiveSwitch command has the option:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-force</td>
<td>Specifies that the command will take effect even if the management instance from which it is run cannot be clearly identified as the standby management instance or is not ideally eligible to become the active management instance. <strong>Important:</strong> Using -force will likely result in substantial data loss. Use only as a last resort.</td>
</tr>
</tbody>
</table>

Examples

```
% ttGridAdmin mgmtActiveSwitch
This is now the active management instance
```

Notes

- This command is typically used if the active management instance has failed.
- If or when the original active management instance is back up, you can use mgmtStandbyStart to restart it as the standby.
- All data instances in the grid will automatically failover from the previous active management instance to the new active management instance.

Examine management instances (mgmtExamine)

The mgmtExamine command examines the management instances and recommends any necessary corrective action. Run the suggested commands.

```
ttGridAdmin mgmtExamine
```

Examples

This example shows output when both management instances are up. Aside from the opening note that they are both up, the output is the same as for the mgmtStatus command. See "Display status of management instances (mgmtStatus)" on page 4-93.
Management instance operations

for descriptions of the columns. (For brevity, the Message column, which had no entries, is not shown in this example.)

% ttGridAdmin mgmtExamine
Both active and standby management instances are up. No action required.

<table>
<thead>
<tr>
<th>Host</th>
<th>Instance</th>
<th>Reachable</th>
<th>RepRole(Self)</th>
<th>Role(Self)</th>
<th>Seq</th>
<th>RepAgent</th>
<th>RepActive</th>
</tr>
</thead>
<tbody>
<tr>
<td>mysys1host</td>
<td>gridmgmt1</td>
<td>Yes</td>
<td>Active</td>
<td>Active</td>
<td>554</td>
<td>Up</td>
<td>Yes</td>
</tr>
<tr>
<td>mysys2host</td>
<td>gridmgmt1</td>
<td>Yes</td>
<td>Standby</td>
<td>Standby</td>
<td>554</td>
<td>Up</td>
<td>No</td>
</tr>
</tbody>
</table>

This example shows output when the active management instance is down, including recommended actions and commands to run:

% ttGridAdmin mgmtExamine
Standby management instance is up, but active is down
Promote the standby to active

<table>
<thead>
<tr>
<th>Host</th>
<th>Instance</th>
<th>Reachable</th>
<th>RepRole(Self)</th>
<th>Role(Self)</th>
<th>Seq</th>
<th>RepAgent</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>mysys1host</td>
<td>gridmgmt1</td>
<td>No</td>
<td>Unknown</td>
<td>Unknown</td>
<td>Down</td>
<td>No</td>
<td>Management database is not available</td>
</tr>
<tr>
<td>mysys2host</td>
<td>gridmgmt1</td>
<td>Yes</td>
<td>Standby</td>
<td>Standby</td>
<td>557</td>
<td>Up</td>
<td>No</td>
</tr>
</tbody>
</table>

Recommended commands:

```
ssh -o StrictHostKeyChecking=yes -o PasswordAuthentication=no -x host1.example.com /sw/tten/gridsetup/ttinstances/gridmgmt1/bin/ttenv ttGridAdmin mgmtActiveSwitch
```

Notes

One use case is if both management instances fail, and you are not certain which one was the active. Run this command to examine them both and determine which one is "current" or "most recent", then start that one as the active management instance.

Start the standby management instance (mgmtStandbyStart)

The mgmtStandbyStart command starts the current management instance (from which the command is run) as the standby management instance.

```
ttGridAdmin mgmtStandbyStart
```

A typical scenario is when the active management instance fails, you promote the standby to active, then run this command to make the original active management instance become the new standby management instance.

Examples

% ttGridAdmin mgmtStandbyStart
Standby management instance started

Notes

- The instance must previously be stopped.
- There must be another management instance previously started as the active management instance.
- This command initiates replication between the active and standby management instances, synchronizing management data between them.
Stop the standby management instance (mgmtStandbyStop)

The `mgmtStandbyStop` command stops the standby management instance.

```
ttGridAdmin mgmtStandbyStop
```

**Examples**

```
% ttGridAdmin mgmtStandbyStop
Standby management instance stopped
```

**Notes**

- This command can be run from either the active or the standby management instance if they are both operational.
- Usage scenarios include:
  - If the standby has failed
  - If you want to stop the standby for any reason, such as to reboot it or perform maintenance
- The command will fail with an error if there is not an operational standby instance at the time the command is run.

Display status of management instances (mgmtStatus)

The `mgmtStatus` command displays status information for the management instances.

```
ttGridAdmin mgmtStatus
```

(Also see "Examine management instances (mgmtExamine)" on page 4-91. The `mgmtExamine` command recommends actions to repair any reported problems with the management instances.)

**Examples**

```
% ttGridAdmin mgmtStatus
Host        Instance  Reachable RepRole(Self) Role(Self) Seq RepAgent RepActive
----------- --------- --------- ------------- ---------- --- -------- ---------
mysys1host  gridmgmt1 Yes       Active        Active     554 Up       Yes
mysys2host  gridmgmt1 Yes       Standby       Standby    554 Up       No
```

For each instance displayed:

- **Host** and **Instance** show the name of the instance and the name of the host where it is located.
- **Reachable** indicates whether the command was successful in reaching the instance to determine its state.
- **RepRole(Self)** indicates the recorded role, if any, for the instance in replicating data between management instances.
- **Role(Self)** indicates the recorded role, if any, for the instance.
- **Seq** is the sequence number of the most recent change on the instance. If the Seq values are the same, then the two management instances are synchronized; otherwise, the one with the larger Seq value has the more recent data.
- **RepAgent** indicates whether the replication agent is running on the instance.
- **RepActive** indicates whether changes by the `mgmtStatus` command to management data on the instance were successful. The `mgmtStatus` command attempts to modify management data on each management instance, but this will not work on the standby management instance, which is read-only.

- **Message** has any further information about the instance. (For brevity, this column is not shown in the example.)
Membership operations

Use ttGridAdmin commands in this section to export or import the membership client configuration file. A typical scenario is if you want to make changes to the file.

---

**Note:** The membership configuration file is first specified when you create the grid, according to the gridCreate -membershipConfig option. See "Create a grid (gridCreate)" on page 4-55.

---

Export the membership configuration file (membershipConfigExport)

The membershipConfigExport command exports the contents of the membership client configuration file from the specified version of the grid model into a specified file.

```
 ttGridAdmin membershipConfigExport [-latest|-current|-version n] [filepath]
```

**Options**

The membershipConfigExport command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-latest</td>
<td>Export the configuration file from the latest model—the model being modified and not yet applied to the grid. This is the default.</td>
</tr>
<tr>
<td>-current</td>
<td>Export the configuration file from the current model—the model most recently applied to the grid.</td>
</tr>
<tr>
<td>-version n</td>
<td>Export the configuration file from the specified version number of the model.</td>
</tr>
<tr>
<td>filepath</td>
<td>The path and name of the file to write the contents of the membership client configuration file to. If no file is specified, the configuration is written to stdout.</td>
</tr>
</tbody>
</table>

**Examples**

```
% ttGridAdmin membershipConfigExport -latest /sw/tten/grid/zkcfg/membership2.conf
% cd /sw/tten/grid/zkcfg
% more membership2.conf
Servers zk1.example.com!2181,zk2.example.com!2181,zk3.example.com!2181
```

The example in the next section will import this file.

Import the membership configuration file (membershipConfigImport)

The membershipConfigImport command replaces the membership client configuration file in the latest model of the grid with the specified file.

```
 ttGridAdmin membershipConfigImport filepath
```

Also see information for the gridCreate -membershipConfig option in "Create a grid (gridCreate)" on page 4-55.

**Options**

The membershipConfigImport command has the option:
Membership operations

---

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>filepath</td>
<td>The path and name of the file that contains the new membership configuration.</td>
</tr>
</tbody>
</table>

**Examples**

This example imports the file created in the example from the preceding section, shown again here:

```
% cd /sw/tten/grid/zkcfg
% more membership2.conf
Servers zk1.example.com!2181,zk2.example.com!2181,zk3.example.com!2181
```

Run the command:

```
% ttGridAdmin membershipConfigImport /sw/tten/grid/zkcfg/membership2.conf
Membership configuration file /sw/tten/grid/zkcfg/membership2.conf imported
```

**Notes**

- Any membership client configuration changes according to the new file are not applied until you execute the `modelApply` command.
- Once you execute `modelApply`, the specified file is copied to each instance of the grid and its settings will take effect on each instance the next time the instance is restarted.
Model operations

Use ttGridAdmin commands in this section to apply the latest version of the model to
the grid, delete a version of the model, export a version of the model to a JSON file,
import a version of the model from a JSON file (to become the latest version), compare
two versions of the model, or list information about all versions of the model.

---

**Note:** The *latest* version of the model is the version that is pending
for edits and updates. It has not yet been applied to the model. The
*current* version of the model is the version most recently applied to the
model. Only the latest version of the model is editable. All other
versions are read-only. When the latest version is applied, it becomes
the current version and a copy is made to serve as the initial latest
version.

---

**Apply the latest version of the model (modelApply)**

The `modelApply` command applies the latest version of the model to the grid,
implementing previous operations. This includes actions such as creating physical
installations and instances according to installation and instance objects that have been
defined in the model.

```
ttGridAdmin modelApply [-nostart]
[-details]
```

**Options**

The `modelApply` command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-nostart</td>
<td>By default, the <code>modelApply</code> command automatically starts new TimesTen Scaleout instances when they are created for the grid. If you specify <code>-nostart</code>, the instances are created but not started.</td>
</tr>
<tr>
<td>-details</td>
<td>Displays additional information about the operations being performed by the command.</td>
</tr>
</tbody>
</table>

**Examples**

This example shows typical output.

```
% ttGridAdmin modelApply
Creating new model version............................................OK
Exporting current model (version 1)...................................OK
Identifying any deleted objects.......................................OK
Verifying installations...............................................OK
Creating new installations............................................OK
Verifying instances...................................................OK
Creating new instances................................................OK
Updating grid state...................................................OK
Configuring instance authentication...................................OK
Pushing new configuration files to each instance......................OK
Making model version 1 current, version 2 writable....................OK
Checking ssh connectivity of new instances............................OK
Starting new management instance......................................OK
Configuring standby management instance...............................OK
ttGridAdmin modelApply complete
```
(Output will vary depending on your situation, such as whether installations or instances in the model already existed, either from being created manually or from any previous `modelApply` commands that were only partially successful.)

Notes

- When a grid is created, version 1 of the model is created automatically. When `modelApply` is executed on the grid for the first time, version 1 of the model is made read-only and version 2 is created. Version 2 is an exact copy of version 1 and is read-write. Version 1 is then applied to the grid. Subsequent changes made to the model are made to version 2, until `modelApply` is executed again, at which time version 3 is created, and so on. There is always a writable version of the model available.

- At any given point, the writable version of the model, which has not yet been applied to the grid, is referred to as the latest version. The version that has been applied and is operational in the grid is referred to as the current version. (The current version and all previous versions are read-only.)

- The `modelApply` command communicates with each instance in the grid and creates or updates configuration files on each instance, including `timesten.conf`, as needed. The command executes these operations in parallel as much as possible, but still may take a significant amount of time to complete. Complete all the steps in getting from one desired configuration to another desired configuration before applying the model.

- It may not always be possible for `modelApply` to complete all of its operations, such as if a host is down. If there are problems, `modelApply` creates error logs in the `diag` directory of the management instance and indicates the names of those logs. The next time you execute `modelApply`, it will try again to complete any operations that failed previously, in addition to completing any new operations.

- See "Applying the changes made to the model" in Oracle TimesTen In-Memory Database Scaleout User’s Guide for additional information.

Compare models (modelCompare)

The `modelCompare` command compares two versions of the model and displays a summary of changes between them.

```
ttGridAdmin modelCompare -latest|-current|-version n
                           [-latest|-current|-version m]
```

Options

The `modelCompare` command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-latest</code></td>
<td>Specifies that the latest version of the model—the model being modified and not yet applied to the grid—is one of the versions to compare. If the command line specifies only one version, that version is compared against the latest version by default.</td>
</tr>
<tr>
<td><code>-current</code></td>
<td>Specifies that the current version of the model—the model most recently applied to the grid—is one of the versions to compare.</td>
</tr>
<tr>
<td><code>-version n</code></td>
<td>Specifies that model version n is one of the versions to compare.</td>
</tr>
</tbody>
</table>
Examples

This example shows that the physical group `location4` was added between the current model and the latest model. (Other differences shown are for meta data.)

```
% ttGridAdmin modelCompare -current -latest
6,9c6,8
<   "version"     : 8,
<   "whenCreated" : "2016-12-02T13:13:05.000Z",
<   "applied" : true,
<   "whenApplied" : "2016-12-13T14:57:41.000Z",
---
>   "version"     : 9,
>   "whenCreated" : "2016-12-13T14:57:19.000Z",
>   "applied" : false,
11c10
<   "current" : true,
---
>   "current" : false,
31a31,33
> },
>   {"type" : "physicalGroup",
>     "name" : "location4"
```

Notes

The summary of changes is displayed in UNIX `diff` format.

Export a version of a model (modelExport)

The `modelExport` command exports information about the grid for the specified version of the model in JSON format, typically to a specified file.

```
ttGridAdmin modelExport [-latest|-current|-version n] [filepath]
```

Within the grid, the hierarchy of the output includes the following:

- SQLNet
- TNSNames
- DataSpaceGroups
- Hosts
  - PhysicalGroups
  - Installations
  - Instances
- Databases
  - Connectables

Options

The `modelExport` command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-latest</td>
<td>Export the latest model—the model being modified and not yet applied to the grid. This is the default.</td>
</tr>
</tbody>
</table>
Examples

Export the current version (version 4) of the model. This is executed from a management instance:

% pwd
/sw/tten/grid/ttinstances/gridmgmt1/bin
% ttGridAdmin modelExport -current /sw/tten/grid/models/model4export.json
Model version 4 exported to /sw/tten/grid/models/model4export.json

Export the latest version (version 5) of the model, which is the default version to export. This is executed from a data instance:

% pwd
/sw/tten/grid/ttinstances/instance1/bin
% ttGridAdmin modelExport /sw/tten/grid/models/model5export.json
Model version 5 exported to /sw/tten/grid/models/model5export.json

Output files:

% pwd
/sw/tten/grid/models
% ls
model4export.json  model5export.json

Notes

■ Metrics and logs are not exported. They exist on the active management instance but are not part of the model.
■ You can execute this command from a management instance or a data instance.
■ You can use modelExport to create a backup of the model.

Import a version of the model (modelImport)

The modelImport command imports a model from a JSON file (perhaps exported earlier using the modelExport command) to update the latest version of the model, or creates a script that you can use to update the model later.

ttGridAdmin modelImport [-script scriptpath]
                          [filepath]

Options

The modelImport command has the options:
Examples

Consider a scenario where you exported the latest version (Version 5) of the model, subsequently made changes to the latest version of the model without applying them, then decided you do not want those changes after all. To undo the changes, import the file you previously exported:

```
% ttGridAdmin modelImport /sw/tten/grid/models/model5export.json
Model imported
```

Without the `-script` option, the model is imported immediately.

With the script option, a script is created that you can run later:

```
% ttGridAdmin modelImport /sw/tten/grid/models/model5export.json -script /sw/tten/grid/models/modelmodscript
Script /sw/tten/grid/models/modelmodscript created.
```

Here is an example of a resulting script:

```
% pwd
/sw/tten/grid/models
% more modelmodscript
#!/bin/sh
# Created by ttGridAdmin -modelImport
TIMESTEN_HOME=/sw/tten/grid/ttinstances/gridmgmt
if [ -e $TIMESTEN_HOME/bin/ttenv.sh ]; then
  . $TIMESTEN_HOME/bin/ttenv.sh >/dev/null 2>&1
fi
# TNSNames unchanged
#Host msys5host...
ttGridAdmin -hostCreate msys5host -internalAddress msys5.example.com -externalAddress msys5.example.com -physicalGroup location2 ttGridAdmin installationCreate msys5host.installslc -location /sw/tten/grid/ttinstallations/installadc/ ttGridAdmin instanceCreate msys5host.instance1 -installation installslc -location /sw/tten/grid/ttinstances/ -daemonPort 20000 -csPort 21000
#Host msys3host...
#Host msys1host...
#Host msys2host...
#Host msys4host...
#Dbdef database1
#Connectable unchanged!
#Connectable unchanged!
#DbDef unchanged!
#Dbdef TTGRIDADMIN
#Connectable unchanged!
#Connectable unchanged!
```

### Option Description

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-script scriptpath</code></td>
<td>Creates a script with the specified name and path. The model is not updated when you execute <code>modelImport</code>. Instead, you can execute the resulting script later to modify the latest version of the model to conform to the imported version. This allows you to review the changes beforehand. Without <code>-script</code>, the latest model is updated immediately.</td>
</tr>
<tr>
<td><code>filepath</code></td>
<td>Path and name of the JSON file from which the representation of the model is read. If <code>filepath</code> is not specified, input is read from stdin.</td>
</tr>
</tbody>
</table>
#DbDef unchanged!

**Notes**

- The `modelImport` command compares the latest version of the model with the model being imported.
- The changes to the latest version of the model are not done in an atomic transaction. Each change is done in a separate transaction, so any failure will result in complications.

**List model versions (modelList)**

The `modelList` command lists the versions of the model, indicating when each was defined, applied, and deleted, as applicable.

```
ttGridAdmin modelList
```

**Examples**

```
% ttGridAdmin modelList
<table>
<thead>
<tr>
<th>Version</th>
<th>Created</th>
<th>Applied</th>
<th>Deleted</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2016-10-06 12:59:26</td>
<td>2016-10-14 13:45:24</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>2016-10-14 13:44:45</td>
<td>2016-10-14 14:33:47</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>2016-10-14 14:33:05</td>
<td>2016-10-14 14:46:33</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>2016-10-14 14:46:20</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>
```
Oracle Database operations

Use ttGridAdmin commands in this section to import or export sqlnet.ora configuration or TNS names entries for connecting to an Oracle database.

These are Oracle Database features that allow an application in TimesTen Scaleout to interact with an Oracle database using the ttLoadFromOracle utility, OCI, or Pro*C/C++.

---

Notes:

- Do not use these commands for OCI or Pro*C/C++ connections to a TimesTen Scaleout database. Entries for tnsnames and sqlnet are made automatically by TimesTen Scaleout.
- The ttLoadFromOracle built-in procedure is for loading data from an Oracle database into TimesTen Classic or TimesTen Scaleout.
- For a summary of TNS names and sqlnet.ora, see “Connecting to a TimesTen database from OCI” in Oracle TimesTen In-Memory Database C Developer’s Guide.

---

Export a sqlnet file (SQLNetExport)

That SQLNetExport command exports sqlnet.ora configuration (that had previously been imported) from the specified version of the model, typically to a specified file.

```
ttGridAdmin SQLNetExport [-latest|-current|-version n] [filepath]
```

Options

The SQLNetExport command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-latest</td>
<td>Export sqlnet.ora configuration from the latest model—the model being modified and not yet applied to the grid. This is the default.</td>
</tr>
<tr>
<td>-current</td>
<td>Export sqlnet.ora configuration from the current model—the model most recently applied to the grid.</td>
</tr>
<tr>
<td>-version n</td>
<td>Export sqlnet.ora configuration from the specified version number of the model.</td>
</tr>
<tr>
<td>filepath</td>
<td>Path and name of the file that will contain the exported sqlnet.ora configuration. If no file is specified, the export goes to stdout.</td>
</tr>
</tbody>
</table>

Examples

This example exports sqlnet.ora from the latest version of the model (by default) then shows the contents of the file.

```
% ttGridAdmin SQLNetExport /sw/tten/grid/misc/sqlnet.ora
% cd /sw/tten/grid/misc
% more sqlnet.ora
```

To use ezconnect syntax or tnsnames, the following entries must be included in the sqlnet.ora configuration.

```
# To use ezconnect syntax or tnsnames, the following entries must be included in the sqlnet.ora configuration.
#
NAMES.DIRECTORY_PATH= (TNSNAMES, EZCONNECT)
```
Import a sqlnet file (SQLNetImport)

The SQLNetImport command imports sqlnet.ora configuration (used in communicating with an Oracle database through ttLoadFromOracle, OCI, Pro*C/C++, or ODP.NET) from the specified file into the sqlnet.ora file for the latest version of the model. This will be in place of any previously existing sqlnet.ora configuration.

```
% ttGridAdmin SQLNetImport filepath
```

---

**Note:** Any previous import is overwritten.

---

**Option**

The SQLNetImport command has the option:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>filepath</td>
<td>Path and name of the file containing sqlnet.ora configuration to import.</td>
</tr>
</tbody>
</table>

---

**Examples**

```
% ttGridAdmin SQLNetImport /tmp/sqlnet.ora
SQLNet configuration file /tmp/sqlnet.ora imported
```

---

**Notes**

- This is the only way to bring sqlnet.ora configuration into the grid. Do not manually add or manipulate configuration files.
- The resulting sqlnet.ora file will be made available across all instances of the grid when you execute modelApply.

**Export TNS names (TNSNamesExport)**

The TNSNamesExport command exports TNS names entries (that had previously been imported) from the specified version of the model, typically to a specified file.

```
% ttGridAdmin TNSNamesExport [-latest|-current|-version n] [filepath]
```

---

**Options**

The TNSNamesExport command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-latest</td>
<td>Export TNS names entries from the latest model—the model being modified and not yet applied to the grid. This is the default.</td>
</tr>
<tr>
<td>-current</td>
<td>Export TNS names entries from the current model—the model most recently applied to the grid.</td>
</tr>
<tr>
<td>-version n</td>
<td>Export TNS names entries from the specified version number of the model.</td>
</tr>
<tr>
<td>filepath</td>
<td>Path and name of the file that will contain the exported TNS names entries. If no file is specified, the export goes to stdout.</td>
</tr>
</tbody>
</table>
Examples

This example exports tnsnames.ora from the latest version of the model (by default), then shows the contents of the file.

```bash
% ttGridAdmin TNSNamesExport /sw/tten/grid/misc/tnsnames.ora
% cd /sw/tten/grid/misc
% more tnsnames.ora
...
ORCL =
  (DESCRIPTION =
   (ADDRESS = (PROTOCOL = TCP)(HOST = www.example.com)(PORT=1630))
   (CONNECT_DATA =
    (SERVICE_NAME = orcl)
  )
...```

Import TNS names (TNSNamesImport)

The TNSNamesImport command imports TNS names entries (used in communicating with an Oracle database through ttLoadFromOracle, OCI, Pro*C/C++, or ODP.NET) from the specified file into the tnsnames.ora file for the latest version of the model. This will replace any previously imported TNS names entries.

```bash
ttGridAdmin TNSNamesImport filepath```

Options

The TNSNamesImport command has the option:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>filepath</td>
<td>Path and name of the file containing TNS entries to import.</td>
</tr>
</tbody>
</table>

Examples

```bash
% ttGridAdmin TNSNamesImport /tmp/tnsnames.ora
TNSNames configuration file /tmp/tnsnames.ora imported```

Notes

- This is the only way to bring TNS names configuration into the grid. Do not manually add or manipulate configuration files.
- The resulting tnsnames.ora file will be made available across all instances of the grid when you execute modelApply.
- The tnsnames.ora file in the grid always contains entries for all connectables. You can add to that through TNSNamesImport, but you cannot remove entries other than any you have previously imported.
Physical group operations

Use the ttGridAdmin commands in this section to define or delete a physical group or to list physical groups in the model.

Create a physical group (physicalCreate)

The physicalCreate command defines a physical group in the model.

\[ \text{ttGridAdmin physicalCreate name} \]

\[ \text{[-comment comment]} \]

Options

The physicalCreate command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Specifies the name for the physical group in the model.</td>
</tr>
<tr>
<td>-comment</td>
<td>Associates a comment with the physical group object. Put the comment in quotes if there are any spaces. The comment is stored and included in output of the physicalList command.</td>
</tr>
</tbody>
</table>

Examples

\% ttGridAdmin physicalCreate location1
PhysicalGroup location1 created.
\% ttGridAdmin physicalCreate location2
PhysicalGroup location2 created.

Also see ”Assigning hosts to physical groups” in Oracle TimesTen In-Memory Database Scaleout User’s Guide.

Delete a physical group (physicalDelete)

The physicalDelete command removes a physical group from the model.

\[ \text{ttGridAdmin physicalDelete name} \]

Options

The physicalDelete command has the option:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>The name of the physical group to delete.</td>
</tr>
</tbody>
</table>

Examples

\% ttGridAdmin physicalDelete location1
PhysicalGroup location1 deleted.
\% ttGridAdmin physicalDelete location2
PhysicalGroup location2 deleted.

Notes

You cannot delete a physical group that is associated with any hosts.
List physical groups (physicalList)

The physicalList command lists all physical groups that are defined in the specified version of the model.

ttGridAdmin physicalList [-latest|-current|-version n]

Options

The physicalList command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-latest</td>
<td>Lists physical groups in the latest model—the model being modified and not yet applied to the grid. This is the default.</td>
</tr>
<tr>
<td>-current</td>
<td>Lists physical groups in the current model—the model most recently applied to the grid.</td>
</tr>
<tr>
<td>-version n</td>
<td>Lists physical groups in the specified version number of the model.</td>
</tr>
</tbody>
</table>

Examples

This example lists physical groups in the current version of the model, then adds another physical group and lists physical groups in the latest version (the default).

```% ttGridAdmin physicalList -current
PhysicalGroup Comment
--------------- -------
location1
location2
location3
% ttGridAdmin physicalCreate location4
PhysicalGroup location4 created.
% ttGridAdmin physicalList
PhysicalGroup Comment
--------------- -------
location1
location2
location3
location4```
Repository operations

Use ttGridAdmin commands in this section to create, attach, detach, or list repositories.

In TimesTen Scaleout, a repository is a file system directory tree used for database backups or exports or for collections of daemon logs. You specify the top-level directory when you create the repository. The contents of the directory and subdirectories of a repository, whether consisting of a backup, an export, or daemon logs, comprise a collection.

A repository is either mounted on each host of the grid (using NFS or equivalent), or mounted on a single host (optionally in the grid) and accessed by other hosts using scp (SSH copy).

For additional information, refer to "Migrating, Backing Up and Restoring Data" in Oracle TimesTen In-Memory Database Scaleout User’s Guide.

Attach a repository (repositoryAttach)

The repositoryAttach command attaches an existing repository to the grid, making it available for use.

ttGridAdmin repositoryAttach name
   -path path
   -method mount|scp
   [-address internalAddress]

Options

The repositoryAttach command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Name of the repository to attach.</td>
</tr>
<tr>
<td>-path path</td>
<td>Fully qualified path to the parent directory where the repository is located.</td>
</tr>
<tr>
<td></td>
<td>For -method mount, this is the full NFS path, such as /net/mysys2/repositories.</td>
</tr>
<tr>
<td></td>
<td>For -method scp, this is the full path, such as /repositories, on the system indicated by -address.</td>
</tr>
<tr>
<td>-method mount</td>
<td>scp</td>
</tr>
<tr>
<td></td>
<td>- mount: The repository is accessed through an NFS mount on each grid host.</td>
</tr>
<tr>
<td></td>
<td>- scp: The repository is accessed by each grid host using scp through passwordless SSH.</td>
</tr>
<tr>
<td></td>
<td>Note: The -method setting for repositoryAttach must match the setting that was used for repositoryCreate when the repository was created.</td>
</tr>
<tr>
<td>-address internalAddress</td>
<td>For repositories accessed through scp, this option is required and indicates the DNS name or IP address of the system where the repository is located.</td>
</tr>
<tr>
<td></td>
<td>Also see &quot;Address formats&quot; on page 4-6.</td>
</tr>
</tbody>
</table>
Examples

This example attaches a repository /repositories/repo1 that is located on the system msys2, using -method scp.

% ttGridAdmin repositoryAttach repo1 -path /repositories -method scp
-addr msys2.example.com
Repository repo1 attached

Notes

This command is typically used to attach a repository that was created from another grid, so that you can restore a backup of a database from one grid into another grid. It can also be used to reattach a repository to the grid where it was created, if it was detached.

Create a repository (repositoryCreate)

The repositoryCreate command creates a new repository that will be available for the grid.

ttGridAdmin repositoryCreate name
-addr path
-method mount|scp
-addr internalAddress

Options

The repositoryCreate command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Specifies the name for the repository. This is the name of the directory that will be created under the parent directory specified by -path.</td>
</tr>
<tr>
<td>-path path</td>
<td>Fully qualified path to the parent directory where the repository is to be created. This directory must already exist on the system(s) where the repository will be located and be readable and writable by the instance administrator. For -method mount, this is the full NFS path, such as /net/msys2/repositories. For -method scp, this is the full path, such as /repositories, on the system indicated by -addr.</td>
</tr>
</tbody>
</table>
| -method mount|scp | Indicates how grid instances access the repository. Supported options are mount or scp.  
■ mount: The repository is accessed through an NFS mount on each grid host.  
■ scp: The repository is accessed by each grid host using scp through passwordless SSH.  
**Note:** If you will later use repositoryAttach for the repository being created, the -method setting for repositoryAttach must match the setting you are using for repositoryCreate. |
| -addr internalAddress | For repositories accessed through scp, this option is required and indicates the DNS name or IP address of the system where the repository is created. Also see “Address formats” on page 4-6. |
Examples

This example creates a repository /repositories/repo1 on the system mysys2, using scp. The instance administrator must have write permission for /repositories.

```bash
% ttGridAdmin repositoryCreate repo1 -path /repositories -method scp -address mysys2.example.com
Repository repo1 created
```

This example creates the repository using mount.

```bash
% ttGridAdmin repositoryCreate repo1 -path /repositories -method mount
Repository repo1 created
```

The `repository.json` file has information about the repository.

Notes

- The repository directory is created synchronously and has permissions of 700. For repositories accessed through NFS mount, the repository directory is `path/name`. For repositories access through `scp`, the repository directory is `@address:path/name`.
- The repository is available for use as soon as it is created.
- Once a repository is created with `repositoryCreate`, you can use `repositoryAttach` to access it from other grids.

Detach a repository (repositoryDetach)

The `repositoryDetach` command detaches (disassociates) a repository from the grid, so that it will no longer be usable from the grid.

```bash
ttGridAdmin repositoryDetach name
```

Options

The `repositoryDetach` command has the option:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Name of the repository to detach (as established when it was attached or created).</td>
</tr>
</tbody>
</table>

Examples

```bash
% ttGridAdmin repositoryDetach repol
Repository repol detached
```

Notes

You can detach a repository that was created in the grid or, more typically, attached in the grid.

List repositories (repositoryList)

The `repositoryList` command lists repositories that are accessible (created or attached) in the grid, optionally including information about contents of the repositories—database backups, database exports, or collections of daemon logs and other information.

```bash
ttGridAdmin repositoryList [name]
```
## Options

The `repositoryList` command has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>name</code></td>
<td>Name of the repository to list. If no name is specified, all repositories accessible to the grid are listed.</td>
</tr>
<tr>
<td><code>-contents</code></td>
<td>Show the contents of each repository listed.</td>
</tr>
<tr>
<td><code>-details</code></td>
<td>Show details about the contents of each repository listed (use with <code>-contents</code>).</td>
</tr>
</tbody>
</table>

## Examples

This examples lists all repositories accessible to the grid (there is only one), but no contents.

```
% ttGridAdmin repositoryList
Repository Method Location              Address
---------- ------ ------------------- ------------------
repo1      scp    /repositories/repo1 mysys2.example.com
```

This example shows contents:

```
% ttGridAdmin repositoryList -contents
Repository Collection Type Date                      Details
---------- --------------- ------ ------------------------ ------------------------
repo1      B20170222145544 Backup 2017-02-22T14:55:48.000Z Database database1
```
ttGridRollout

Description

The ttGridRollout utility, run from the installation_dir/tt18.1.2.1.0/bin directory of your TimesTen installation, creates a new grid with one database definition. The database is created and loaded, its distribution is configured, then it is opened.

The utility reads a configuration file that contains user-defined parameters and attributes for the grid you want to create. TimesTen provides a configuration template that you can copy and modify.

You can specify the shape of the grid, the hosts to use, the number of management instances (one or two), and the number of data instances, among other settings. By default, if you do not specify hosts, management instances, and data instances, then a single management instance and a number of data instances suitable for the specified shape are created on your local host.

The installation from which you run ttGridRollout is copied to the other hosts you specify so that additional data instances and a standby management instance can be created as desired. If you specify two management instances, the first must be on your local host.

The ttGridRollout utility is a wrapper for the ttGridAdmin utility (also using ttInstanceCreate for the first management instance), and actions performed by ttGridRollout can optionally be performed directly using ttGridAdmin. Once you have created a grid with ttGridRollout, use ttGridAdmin to maintain it and to make any changes.

The ttGridRollout utility is typically used for creating sample grids or grids that will be used during product design and evaluation.

Important:

- Execute this utility exactly as "ttGridRollout" (for example, not as "ttgridrollout").
- The ttGridRollout utility does not support physical groups.

Required privilege

The user who runs this utility becomes the instance administrator of all instances created, and the user’s primary user group becomes the TimesTen user group.

File system write permission is required wherever installations and instances will be created.

Usage with TimesTen Scaleout

This utility is specifically for use with TimesTen Scaleout.

Syntax

```
ttGridRollout [-h | -help | -?]
ttGridRollout [-n | -dry-run] [-wait n] [-timeout n] conf_file
```
ttGridRollout has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h</td>
<td>Displays help information.</td>
</tr>
<tr>
<td>-help</td>
<td>Specifies the configuration file that contains the parameters for creating the grid and database. A read-only template, ttgrid.conf.example, is located in the installation_dir/tt18.1.2.1.0/grid/conf directory. You can copy and modify this file to set up your configuration.</td>
</tr>
<tr>
<td>-?</td>
<td>Displays the commands to be executed but does not execute them. Other options you specify will be reflected in the display of commands to be executed. <strong>Note:</strong> It is advisable to do this before executing the command.</td>
</tr>
<tr>
<td>-n</td>
<td>-dry-run Specifies how long ttGridRollout will wait for database state changes to complete before returning. By default, there is no limit to the wait. (Database operations in TimesTen Scaleout, such as creating, loading, and opening, initiate a state change that is recorded in the active management instance of the grid. The state change is complete once the database operation has completed on each instance of the grid.)</td>
</tr>
<tr>
<td>-timeout n</td>
<td>Maximum number of seconds to wait for a long-running operation to complete. The default is 600.</td>
</tr>
</tbody>
</table>

**Note:** the -wait option applies only to database operations. The -timeout option applies to any operation. These options are passed to ttGridAdmin.

Also see "Command timeouts and waits" on page 4-8.

### Configuration file parameters

The table that follows describes configuration parameters supported by the ttGridRollout configuration file (named ttgrid.conf by convention).

**Important:** These parameters are required in your configuration file:

- grid_name
- dbdef_file
- shape (optionally with data_hosts) or data_instances
- instance_location
- zoo_conf, unless all TimesTen instances and the membership server are on the local host
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cs_connect_files</td>
<td>Connectable files (.connect) for client/server connectables, as desired. You can specify multiple, comma-separated .connect files. For example:</td>
</tr>
<tr>
<td></td>
<td><code>cs_connect_files = client1.connect, client2.connect</code></td>
</tr>
<tr>
<td></td>
<td>For information about connectable files, see &quot;Create a connectable (connectableCreate)&quot; on page 4-17.</td>
</tr>
</tbody>
</table>
| data_hosts           | List of entries for hosts to be used for data instances, in JSON format.  
                      | Optionally use this with `shape` (and it cannot be used without `shape`).  
                      | Do not use both `data_hosts` and `data_instances`.  
                      | If you do not specify enough hosts for an $N \times K$ grid (see the description for `shape`), `ttGridRollout` loops back to the start of the specified host list and will place additional instances on as many hosts as necessary. If you specify too many hosts, only the first $N \times K$ hosts are used.  
                      | This parameter supports the attributes `address`, `externalAddress`, `internalAddress`, `installation_location`, and `instance_location`. See "Configuration file parameter attributes" on page 4-118, including information about default values.  
                      | Specifying address(es) is required—either `address` OR `externalAddress` and `internalAddress`.  
                      | **Example:**  
                      | `data_hosts = [  
                          { "internalAddress":"thost1-priv",  
                            "externalAddress":"thost1.example.com",  
                            "installation_location":"/u01/tthost1/TimesTen" },  
                          { "internalAddress":"thost2-priv",  
                            "externalAddress":"thost2.example.com",  
                            "installation_location":"/u01/tthost2/TimesTen" },  
                          { "internalAddress":"thost3-priv",  
                            "externalAddress":"thost3.example.com",  
                            "installation_location":"/u01/tthost3/TimesTen" },  
                          { "internalAddress":"thost4-priv",  
                            "externalAddress":"thost4.example.com",  
                            "installation_location":"/u01/tthost4/TimesTen" }  
                      ]`  
                      | **Notes:** See notes for `shape`. |
**data_instances**

List of entries for data instances, in JSON format. This parameter allows you to specify data space groups, host and instance names, and daemon and client/server port numbers.

You cannot use **data_instances** together with shape or **data_hosts**. (Specify shape, with or without **data_hosts**, or specify **data_instances**.)

This parameter supports the attributes `address`, `externalAddress`, `internalAddress`, `host`, `instance`, `dataSpaceGroup`, `daemonport`, `csport`, `installation_location`, and `instance_location`. See "Configuration file parameter attributes" on page 4-118, including information about default values.

The shape of the grid is determined by your **dataSpaceGroup** settings. If you do not specify data space groups, the grid will be 1x1, with one data space group.

Specifying address(es) is required—either `address` OR `externalAddress` and `internalAddress`.

**Example:**

```plaintext
data_instances = [ { "internalAddress":"tthost1-priv", "externalAddress":"tthost1.example.com", "dataSpaceGroup":1, "daemonport":50001, "csport":50002 }, { "internalAddress":"tthost2-priv", "externalAddress":"tthost2.example.com", "dataSpaceGroup":1, "daemonport":50001, "csport":50002 }, { "internalAddress":"tthost3-priv", "externalAddress":"tthost3.example.com", "dataSpaceGroup":2, "daemonport":50001, "csport":50002 }, { "internalAddress":"tthost4-priv", "externalAddress":"tthost4.example.com", "dataSpaceGroup":2, "daemonport":50001, "csport":50002 } ]
```

**Notes:**

- A host object is created for each instance, except where two data instances are specified to be on the same system and in the same dataspace group, in which case they will share the same host object.
- An installation object is created for each host object. Multiple installation objects can point to the same physical installation.

**dbdef_file**

Database definition file (.dbdef). This is required.

Directories are created on each host as necessary for the **DataStore** and **LogDir** locations.

For information about database definition files, see "Create a database definition (dbdefCreate)" on page 4-24.

**direct_connect_files**

Connectable files (.connect) for additional direct connectables, as desired, beyond the connectable that is automatically created when the database is created. You can specify multiple, comma-separated .connect files. For example:

```
direct_connect_files = mydbcfg1.connect, mydbcfg2.connect
```

For information about connectable files, see "Create a connectable (connectableCreate)" on page 4-17.

**grid_name**

The desired name of the grid. This is required.
### init_script

A SQL script (`ttIsql` script) for `ttGridRollout` to execute on the database after rolling out the grid (using the first data instance that was created). For example, the script may include SQL statements to create database users and schemas.

### installation_location

Path to the parent directory where you want to put the TimesTen installation on systems where the standby management instance (if applicable) and data instances are located. The `tt18.1.2.1.0` directory is directly under this location. The directory is created on each host as necessary. If you specify an existing location, the directory must be empty.

The default is to use the same location as for the installation on the local host, from which `ttGridRollout` is run.

This location is used throughout the grid, except where you override it for a particular host or instance by setting the `installation_location` attribute of the `data_hosts`, `data_instances`, or `mgmt_instances` parameter.

### instance_config

A file for custom configuration of data instances, consisting of `name=value` pairs for any settings you want to add to the instance configurations.

This is accomplished using the `ttGridAdmin instanceConfigImport` command. Also see "Import instance configuration attributes (instanceConfigImport)" on page 4-80.

### instance_location

Path to the parent directory for TimesTen instances (data and management). This is required. For each instance, the `timesten_home` directory will be named `instancename` under this location. The directory is created on each host as necessary.

This location is used throughout the grid, except where you override it for a particular host or instance by setting the `instance_location` attribute of the `data_hosts`, `data_instances`, or `mgmt_instances` parameter.

### mgmt_instances

List of entries for management instances, in JSON format. The first entry must be on the local host and will be the active management instance. The second entry (if applicable) must be on a different system and will be the standby management instance.

If you do not set `mgmt_instances`, `ttGridRollout` creates one management instance on the local host.

This parameter supports the attributes `address`, `externalAddress`, `internalAddress`, `host`, `instance`, `daemonport`, `csport`, `mgmtport`, `installation_location`, and `instance_location`. See "Configuration file parameter attributes" on page 4-118, including information about default values.

Specifying address(es) is required—either `address` OR `externalAddress` and `internalAddress`.

**Example:**

```json
mgmt_instances = [  
{ "internalAddress":"tthost1-priv",  
"externalAddress":"tthost1.example.com" },  
{ "internalAddress":"tthost2-priv",  
"externalAddress":"tthost2.example.com" }  
]
```

**Notes:**
- A host object and an installation object are created for each instance.
The desired shape of the grid, $NxK$, where:

- $N$ is the number of instances in each data space group.
- $K$ is the K-factor (replication factor) of the grid, which is by definition the number of data space groups (1 or 2).

Either specify `shape`, with or without `data_hosts`, or specify `data_instances`. If you use `shape` without `data_hosts`, all TimesTen instances are placed on the local host.

When you specify `shape` for an $NxK$ grid, $NxK$ instances will be created (such as eight instances for a 4x2 grid). The first $N$ instances will be in data space group 1, and for $k=2$ the next $N$ data instances will be in data space group 2.

Notes:

- A host object is created for each instance, except where two data instances will be on the same system and in the same datasource group, in which case they will share the same host object.
- If you specify host addresses as DNS names, default host object names are according to the addresses (such as `mysys1` for an address `mysys1.example.com`, with `_2` appended if there is a second host object on the same system, or `mysys1_mgmt` for the host of a management instance). If you specify addresses as IP addresses, default host object names are `host_n` sequentially.
- An installation object is created for each host object. Multiple installation objects can point to the same physical installation.
- As `ttGridRollout` creates instances, it names them `instance1`, `instance2`, `instance3`, and so on.

### sqlnet_config

SQL*Net configuration file (used in communicating with an Oracle database through `ttLoadFromOracle`, OCI, Pro*C/C++, or ODP.NET).

Through the `ttGridAdmin SQLNetImport` command, `ttGridRollout` applies the specified SQL*Net configuration on all data instances. Also see "Export a sqlnet file (SQLNetExport)" on page 4-103.

### tnsnames_config

TNS names configuration file (used in communicating with an Oracle database through `ttLoadFromOracle`, OCI, Pro*C/C++, or ODP.NET).

Through the `ttGridAdmin TNSNamesImport` command, `ttGridRollout` applies the specified TNS names configuration on all data instances. Also see "Import TNS names (TNSNamesImport)" on page 4-105.

### zoo_conf

Apache ZooKeeper membership service client configuration file. This parameter is required unless all management instances, data instances, and the ZooKeeper membership server will be on your local host.

For examples of ZooKeeper client configuration files, see "Membership operations" on page 4-95. For details on how to configure ZooKeeper as a membership service, see "Using Apache ZooKeeper as the membership service" in the Oracle TimesTen In-Memory Database Scaleout User’s Guide.

If you do not specify this parameter, `ttGridRollout` assumes that a ZooKeeper server already runs on the local host using the default client port setting, 2181.
**Configuration file parameter attributes**

The `ttGridRollout` configuration parameters support these attributes. Refer to the preceding table of parameters to see which attributes are supported by each parameter.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>address</strong></td>
<td>DNS name or IP address of the system for both external and internal communications, if a single address is used. Either use address or use <code>internalAddress</code> and <code>externalAddress</code>. Setting <code>-address</code> <code>xxx</code> is exactly equivalent to setting <code>-internalAddress</code> <code>xxx</code> and <code>-externalAddress</code> <code>xxx</code>. This option takes one name or address only, and a specified name must resolve to one IP address or to multiple IP addresses on the same network segment. If host names from <code>/etc/hosts</code> are being used, the <code>/etc/hosts</code> files on all instances in the grid must contain identical entries for all hosts in the grid. <strong>Note</strong>: Using a single address is not recommended for production environments. Also see &quot;Address formats&quot; on page 4-6.</td>
</tr>
<tr>
<td><strong>csport</strong></td>
<td>Port for client/server connections. If this is not specified for a data instance, <code>ttGridRollout</code> uses an available port between 46337 and 46997. If this is not specified for a management instance, <code>ttGridRollout</code> attempts to use the TimesTen default client/server port, 6625.</td>
</tr>
<tr>
<td><strong>daemonport</strong></td>
<td>Port for TimesTen daemon communications. If this is not specified for a data instance, <code>ttGridRollout</code> uses an available port between 46337 and 46997. If this is not specified for a management instance, <code>ttGridRollout</code> attempts to use the TimesTen default daemon port, 6624.</td>
</tr>
<tr>
<td><strong>dataSpaceGroup</strong></td>
<td>Desired data space group (1 or 2). The default is data space group 1. If you use the <code>data_instances</code> parameter, you can use this attribute to specify the data space group for the instance.</td>
</tr>
<tr>
<td><strong>externalAddress</strong></td>
<td>DNS name or IP address of the system for external communications (outside the grid) for client/server connections. Either use address or use <code>internalAddress</code> and <code>externalAddress</code>. Setting <code>-internalAddress</code> <code>xxx</code> and <code>-externalAddress</code> <code>xxx</code> is exactly equivalent to setting <code>-address</code> <code>xxx</code>. This option takes one name or address only, but a name may resolve to one or more IP addresses. If host names from <code>/etc/hosts</code> are being used, the <code>/etc/hosts</code> files on all instances in the grid must contain identical entries for all hosts in the grid. Also see &quot;Address formats&quot; on page 4-6.</td>
</tr>
<tr>
<td><strong>host</strong></td>
<td>Desired name of the host object in the grid model. <strong>Note</strong>: If you specify host addresses as DNS names, default host object names are according to the addresses (such as <code>mysys1</code> for an address <code>mysys1.example.com</code>, with <code>_2</code> appended if there is a second host object on the same system, or <code>mysys1_mgmt</code> for the host of a management instance). If you specify addresses as IP addresses, default host object names are <code>host_n</code> sequentially.</td>
</tr>
</tbody>
</table>
Examples

This sections provides three `ttGridRollout` examples with various types of configuration:

- shape parameter without `data_hosts` to configure a 2x2 grid with one management instance, all on the local host
- shape parameter with `data_hosts` to configure a 2x2 grid with one management instance on four systems, with the management instance and a data instance on the local host
- `data_instances` parameter to configure a 3x2 grid with two management instances on eight systems, with the first management instance on the local host

Each example includes the configuration, the dry run output showing the `ttGridAdmin` commands to be executed, and portions of the execution output. In each example, `mysys1` is the local host. Dry run output is edited for readability. (The `ttInstanceCreate` utility, to create the first management instance, is executed through the full path to the installation `bin` directory; `ttGridAdmin` is executed through the full path to the TimesTen `ttenv` environment setup script in the first management instance `bin` directory.)

Additional examples are in "Deploy a grid and database" in Oracle TimesTen In-Memory Database Scaleout User’s Guide.
shape without data_hosts: This scenario is convenient for standalone development.

Configuration:

- dbdef_file = /sw/tten/dbdef/database1.dbdef
- shape = 2x2
- zoo_conf = /sw/tten/zkconfig/membership.conf
- grid_name = grid1
- instance_location = /sw/tten/grid1/ttinstances

Dry run:

```
% ./ttGridRollout -dry-run /sw/tten/gridconfig/ttgrid1.conf
ttInstanceCreate -grid -location /sw/tten/grid1/ttinstances -name grid1_mgmt
ttGridAdmin gridCreate grid1 -k 2 -host mysys1_mgmt -address mysys1
  -membershipConfig /sw/tten/zkconfig/membership.conf
ttGridAdmin hostCreate mysys1 -address mysys1 -dataspacegroup 1
  -installationCreate mysys1 -location /sw/tten/grid1/ttinstall/installation1
  ttGridAdmin hostCreate mysys1_2 -address mysys1 -dataspacegroup 2
  ttGridAdmin installationCreate mysys1_2 -location /sw/tten/grid1/ttinstall/installation1
  ttGridAdmin instanceCreate mysys1.instance1 -location /sw/tten/grid1/ttinstances
  -daemonport 46337 -csport 46338
  ttGridAdmin instanceCreate mysys1.instance2 -location /sw/tten/grid1/ttinstances
  -daemonport 46339 -csport 46340
  ttGridAdmin instanceCreate mysys1.instance3 -location /sw/tten/grid1/ttinstances
  -daemonport 46341 -csport 46342
  ttGridAdmin instanceCreate mysys1.instance4 -location /sw/tten/grid1/ttinstances
  -daemonport 46343 -csport 46344
  ttGridAdmin dbdefCreate /sw/tten/dbdef/database1.dbdef
  ttGridAdmin modelApply
  ttGridAdmin dbCreate -wait database1
  ttGridAdmin dbDistribute database1 -add all -apply
  ttGridAdmin dbOpen -wait database1
```

Execution:

```
% ./ttGridRollout /sw/tten/gridconfig/ttgrid1.conf
INFO: Generating data_instances for 2x2 Grid
  data_instances = [
    { "address":"mysys1", "dataspacegroup":1 },
    { "address":"mysys1", "dataspacegroup":1 },
    { "address":"mysys1", "dataspacegroup":2 },
    { "address":"mysys1", "dataspacegroup":2 }
  ]
INFO: Checking Zookeeper on zk1:2181 -- OK
INFO: Checking Zookeeper on zk2:2181 -- OK
INFO: Checking Zookeeper on zk3:2181 -- OK
INFO: Checking the address for the management database -- OK
INFO: Checking connectivity to mysys1 -- OK

================================================================================
... 4-instance (2x2) grid successfully created.

Management Instance Location
----------------------------
  - mysys1:/sw/tten/grid1/ttinstances/grid1_mgmt
```
... Data Instance Locations
-----------------------
- mysys1.instance1 ==> mysys1:/sw/tten/grid1/ttinstances/instance1
- mysys1.instance2 ==> mysys1:/sw/tten/grid1/ttinstances/instance2
- mysys1_2.instance3 ==> mysys1:/sw/tten/grid1/ttinstances/instance3
- mysys1_2.instance4 ==> mysys1:/sw/tten/grid1/ttinstances/instance4
...

**shape with data_hosts**: This scenario is useful for initial testing on multiple systems.

**Configuration:**

- dbdef_file = /sw/tten/dbdef/database1.dbdef
- shape = 2x2
- zoo_conf = /sw/tten/zkconfig/membership.conf
- grid_name = grid1
- instance_location = /sw/tten/grid1/ttinstances
- data_hosts = [
  { "internalAddress":"mysys1-i", "externalAddress":"mysys1.example.com" },
  { "internalAddress":"mysys2-i", "externalAddress":"mysys2.example.com" },
  { "internalAddress":"mysys3-i", "externalAddress":"mysys3.example.com" },
  { "internalAddress":"mysys4-i", "externalAddress":"mysys4.example.com" }
]

**Dry run:**

```
% ./ttGridRollout -dry-run /sw/tten/gridconfig/ttgrid1.conf
ttInstanceCreate -grid -location /sw/tten/grid1/ttinstances -name grid1_mgmt
   -membershipConfig /sw/tten/zkconfig/membership.conf
ttGridAdmin hostCreate mysys1 -externaladdress mysys1.example.com
   -internaladdress mysys1-i -dataspacegroup 1
   ttGridAdmin installationCreate mysys1 -location /sw/tten/grid1/ttinstall/installation1
   ttGridAdmin hostCreate mysys2 -externaladdress mysys2.example.com
   -internaladdress mysys2-i -dataspacegroup 1
   ttGridAdmin installationCreate mysys2 -location /sw/tten/grid1/ttinstall/installation1
   ttGridAdmin hostCreate mysys3 -externaladdress mysys3.example.com
   -internaladdress mysys3-i -dataspacegroup 2
   ttGridAdmin installationCreate mysys3 -location /sw/tten/grid1/ttinstall/installation1
   ttGridAdmin hostCreate mysys4 -externaladdress mysys4.example.com
   -internaladdress mysys4-i -dataspacegroup 2
   ttGridAdmin installationCreate mysys4 -location /sw/tten/grid1/ttinstall/installation1
   ttGridAdmin instanceCreate mysys1.instance1 -location /sw/tten/grid1/ttinstances
   -daemonport 46337 -csport 46338
   ttGridAdmin instanceCreate mysys2.instance2 -location /sw/tten/grid1/ttinstances
   -daemonport 46339 -csport 46340
   ttGridAdmin instanceCreate mysys3.instance3 -location /sw/tten/grid1/ttinstances
   -daemonport 46341 -csport 46342
   ttGridAdmin instanceCreate mysys4.instance4 -location /sw/tten/grid1/ttinstances
   -daemonport 46343 -csport 46344
```

```
ttGridAdmin dbdefCreate /sw/tten/dbdef/database1.dbdef
   ttGridAdmin modelApply
```
ttGridAdmin dbCreate -wait database1
ttGridAdmin dbDistribute database1 -add all -apply
ttGridAdmin dbOpen -wait database1

Execution:

% ./ttGridRollout /sw/tten/gridconfig/ttgrid1.conf
INFO: Generating data_instances for 2x2 Grid
data_instances = [
  { "externaladdress":"mysys1.example.com", "internaladdress":"mysys1-i", "dataspacegroup":1 },
  { "externaladdress":"mysys2.example.com", "internaladdress":"mysys2-i", "dataspacegroup":1 },
  { "externaladdress":"mysys3.example.com", "internaladdress":"mysys3-i", "dataspacegroup":2 },
  { "externaladdress":"mysys4.example.com", "internaladdress":"mysys4-i", "dataspacegroup":2 }
]
INFO: Checking Zookeeper on zk1!2181 -- OK
INFO: Checking Zookeeper on zk2!2181 -- OK
INFO: Checking Zookeeper on zk3!2181 -- OK
INFO: Checking the address for the management database -- OK
INFO: Checking connectivity to mysys1 -- OK
INFO: Checking connectivity to mysys1-i -- OK
INFO: Checking connectivity to mysys2-i -- OK
INFO: Checking connectivity to mysys3-i -- OK
INFO: Checking connectivity to mysys4-i -- OK

================================================================================
...
================================================================================
4-instance (2x2) grid successfully created.

Management Instance Location
-------------------------------
- mysys1:/sw/tten/grid1/ttinstances/grid1_mgmt

...

Data Instance Locations
-----------------------
- mysys1.instance1 ==> mysys1-i:/sw/tten/grid1/ttinstances/instance1
- mysys2.instance2 ==> mysys2-i:/sw/tten/grid1/ttinstances/instance2
- mysys3.instance3 ==> mysys3-i:/sw/tten/grid1/ttinstances/instance3
- mysys4.instance4 ==> mysys4-i:/sw/tten/grid1/ttinstances/instance4

...

data_instances: This scenario is useful for more realistic proof-of-concept testing.

Configuration:

dbdef_file = /sw/tten/dbdef/database1.dbdef
zoo_conf = /sw/tten/zkconfig/membership.conf
grid_name = grid1
instance_location = /sw/tten/grid1/ttinstances
data_instances = [
mgmt_instances = [
    { "internalAddress":"mysys1-i", "externalAddress":"mysys1.example.com" },
    { "internalAddress":"mysys2-i", "externalAddress":"mysys2.example.com" }]

Dry run:

% ./ttGridRollout -dry-run /sw/tten/gridconfig/ttgrid1.conf
ttInstanceCreate -grid -location /sw/tten/grid1/ttinstances -name grid1_mgmt
ttGridAdmin gridCreate grid1 -k 2 -host mysys1-i_mgmt -internalAddress mysys1-i
-externalAddress mysys1.example.com -membershipConfig /sw/tten/zkconfig/membership.conf
    ttGridAdmin hostCreate mysys2-i_mgmt -internalAddress mysys2-i
    -externalAddress mysys2.example.com
    ttGridAdmin installationCreate mysys2-i_mgmt -location /sw/tten/grid1/ttinstall/installation1
    ttGridAdmin instanceCreate mysys2-i_mgmt.grid1_mgmt2 -location /sw/tten/grid1/ttinstances
    -type management
    ttGridAdmin hostCreate mysys3 -externaladdress mysys3.example.com
        -internaladdress mysys3-i -dataspacegroup 1
    ttGridAdmin installationCreate mysys3 -location /sw/tten/grid1/ttinstall/installation1
    ttGridAdmin hostCreate mysys4 -externaladdress mysys4.example.com
        -internaladdress mysys4-i -dataspacegroup 1
    ttGridAdmin installationCreate mysys4 -location /sw/tten/grid1/ttinstall/installation1
    ttGridAdmin hostCreate mysys5 -externaladdress mysys5.example.com
        -internaladdress mysys5-i -dataspacegroup 1
    ttGridAdmin installationCreate mysys5 -location /sw/tten/grid1/ttinstall/installation1
    ttGridAdmin hostCreate mysys6 -externaladdress mysys6.example.com
        -internaladdress mysys6-i -dataspacegroup 2
    ttGridAdmin installationCreate mysys6 -location /sw/tten/grid1/ttinstall/installation1
    ttGridAdmin hostCreate mysys7 -externaladdress mysys7.example.com
        -internaladdress mysys7-i -dataspacegroup 2
    ttGridAdmin installationCreate mysys7 -location /sw/tten/grid1/ttinstall/installation1
    ttGridAdmin hostCreate mysys8 -externaladdress mysys8.example.com
    -internaladdress mysys8-i -dataspacegroup 2
    ttGridAdmin installationCreate mysys8 -location /sw/tten/grid1/ttinstall/installation1
    ttGridAdmin instanceCreate mysys3.instance1 -location /sw/tten/grid1/ttinstances
        -daemonport 50001 -csport 50002
    ttGridAdmin instanceCreate mysys4.instance2 -location /sw/tten/grid1/ttinstances
        -daemonport 50001 -csport 50002
    ttGridAdmin instanceCreate mysys5.instance3 -location /sw/tten/grid1/ttinstances
        -daemonport 50001 -csport 50002
ttGridAdmin instanceCreate mysys6.instance4 -location /sw/tten/grid1/ttinstances -daemonport 50001 -csport 50002
ttGridAdmin instanceCreate mysys7.instance5 -location /sw/tten/grid1/ttinstances -daemonport 50001 -csport 50002
ttGridAdmin instanceCreate mysys8.instance6 -location /sw/tten/grid1/ttinstances -daemonport 50001 -csport 50002
ttGridAdmin dbdefCreate /sw/tten/dbdef/database1.dbdef
ttGridAdmin modelApply
ttGridAdmin dbCreate -wait database1
ttGridAdmin dbDistribute database1 -add all -apply
ttGridAdmin dbOpen -wait database1

Execution:

% ./ttGridRollout /sw/tten/gridconfig/ttgrid1.conf
INFO: Checking Zookeeper on zk1:2181 -- OK
INFO: Checking Zookeeper on zk2:2181 -- OK
INFO: Checking Zookeeper on zk3:2181 -- OK
INFO: Checking the address for the management database -- OK
INFO: Checking connectivity to mysys1-i -- OK
INFO: Checking connectivity to mysys2-i -- OK
INFO: Checking connectivity to mysys3-i -- OK
INFO: Checking connectivity to mysys4-i -- OK
INFO: Checking connectivity to mysys5-i -- OK
INFO: Checking connectivity to mysys6-i -- OK
INFO: Checking connectivity to mysys7-i -- OK
INFO: Checking connectivity to mysys8-i -- OK

=======================================================================

...  

=======================================================================

6-instance (3x2) grid successfully created.

Management Instance Locations
--------------------------------
- mysys1-i:/sw/tten/grid1/ttinstances/grid1_mgmt
- mysys2-i:/sw/tten/grid1/ttinstances/grid1_mgmt2

Data Instance Locations
-----------------------
- mysys3.instance1 => mysys3-i:/sw/tten/grid1/ttinstances/instance1
- mysys4.instance2 => mysys4-i:/sw/tten/grid1/ttinstances/instance2
- mysys5.instance3 => mysys5-i:/sw/tten/grid1/ttinstances/instance3
- mysys6.instance4 => mysys6-i:/sw/tten/grid1/ttinstances/instance4
- mysys7.instance5 => mysys7-i:/sw/tten/grid1/ttinstances/instance5
- mysys8.instance6 => mysys8-i:/sw/tten/grid1/ttinstances/instance6

...
This chapter provides reference information for most TimesTen utilities beginning with the following introductory sections:

- Overview
- Required authentication and authorization for utilities
- Utilities List

For information about utilities that are only supported in TimesTen Scaleout see Chapter 4, "TimesTen Scaleout Utilities".

**Overview**

The options for TimesTen utilities are generally not case sensitive, except for single character options. You can use `-connstr` or `-connStr` interchangeably. However `-v` and `-V` are each unique options.

All utilities return 0 for success and nonzero if an error occurs.

---

**Note:** The utility name and options listed in this chapter are case-insensitive. They appear in mixed case to make the examples and syntax descriptions easier to read.

---

**Required authentication and authorization for utilities**

The following sections describe the authentication and authorization required for utilities:

- Required user authentication for utilities
- Required privileges for executing utilities

**Required user authentication for utilities**

All utilities that require a password prompt for one.

If a `UID` connection attribute is given but no `PWD` attribute is given, either through a connection string or in the `ODBCINI` file for the specified DSN, TimesTen prompts for a password. When explicitly prompted, input is not displayed on the command line.

Generally, when no `UID` connection attribute is given, the `UID` is assumed to be the user name identified by the operating system, and TimesTen does not prompt for a password.
When a utility accepts a DSN, connection string or database path as a parameter, specify the value at the end of the command line.

---

**Note:** For security reasons, we do not recommend setting a a value for `PWD` on the command line.

---

**Required privileges for executing utilities**

Certain TimesTen command-line utilities require privileges. Each utility in this chapter describes the privilege required for execution. You may receive a `database not loaded` error if you try to execute any utility with a user other than the instance administrator and the database is not loaded into memory. In this case, TimesTen cannot determine the privileges of the user. Thus any utilities requiring privileges have to be run either as the instance administrator or executed while the database is loaded.

---

**Utilities List**

Utilities listed in Table 5–1 are described in Chapter 4, "TimesTen Scaleout Utilities".

Utilities listed in Table 5–2 are described in this chapter.

**Table 5–1 Utilities supported only in TimesTen Scaleout descriptions**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ttGridAdmin</td>
<td>Administers a TimesTen Scaleout grid.</td>
</tr>
<tr>
<td>ttGridRollout</td>
<td>Creates a new grid and database.</td>
</tr>
</tbody>
</table>

**Table 5–2 Other utilities descriptions**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Usage with TimesTen Scaleout</th>
</tr>
</thead>
<tbody>
<tr>
<td>ttAdmin</td>
<td>Specifies or changes database policies.</td>
<td>No</td>
</tr>
<tr>
<td>ttAdoptStores</td>
<td>Moves databases from a TimesTen installation to a new TimesTen installation of the same major release, but a different minor release.</td>
<td>No</td>
</tr>
<tr>
<td>ttBackup</td>
<td>Creates a backup copy of a database that can be restored at a later time using the <code>ttRestore</code> utility.</td>
<td>No</td>
</tr>
<tr>
<td>ttBulkCp</td>
<td>Copies data between TimesTen tables and ASCII files.</td>
<td>No</td>
</tr>
<tr>
<td>ttCapture</td>
<td>Captures information about the state of TimesTen.</td>
<td>No</td>
</tr>
<tr>
<td>ttCheck</td>
<td>Performs internal consistency checking within a TimesTen database.</td>
<td>No</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Usage with TimesTen Scaleout</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------------------------------------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>ttCWAdmin</td>
<td>Manages TimesTen active standby pairs that take advantage of the high availability framework of Oracle Clusterware.</td>
<td>No</td>
</tr>
<tr>
<td>ttDaemonAdmin</td>
<td>Starts and stops the TimesTen main daemon and Server.</td>
<td>No</td>
</tr>
<tr>
<td>ttDaemonLog</td>
<td>Controls and displays daemon log messages.</td>
<td>No</td>
</tr>
<tr>
<td>ttDestroy</td>
<td>Destroys a database including all checkpoint files, transaction logs and daemon catalog entries.</td>
<td>No</td>
</tr>
<tr>
<td>ttInstallationCheck</td>
<td>Examines all files in an installation of TimesTen and generates a signature for the installation.</td>
<td>Yes</td>
</tr>
<tr>
<td>ttInstallDSN</td>
<td>Generates a Windows client DSN for one or more entries listed in the provided input file and installs them into the ODBC Panel as a System DSN.</td>
<td>Yes</td>
</tr>
<tr>
<td>ttInstanceCreate</td>
<td>Create a new TimesTen instance.</td>
<td>Yes</td>
</tr>
<tr>
<td>ttInstanceDestroy</td>
<td>Destroys an existing TimesTen instance.</td>
<td>Yes</td>
</tr>
<tr>
<td>ttInstanceModify</td>
<td>Modifies certain attributes of an instance.</td>
<td>Yes</td>
</tr>
<tr>
<td>ttIsql</td>
<td>Executes SQL statements interactively.</td>
<td>Yes</td>
</tr>
<tr>
<td>ttMigrate</td>
<td>Saves and restores TimesTen objects.</td>
<td>Yes, only for migrating from TimesTen Classic to TimesTen Scaleout.</td>
</tr>
<tr>
<td>ttRepAdmin</td>
<td>Displays, sets, modifies and monitors existing replication definitions and status.</td>
<td>No</td>
</tr>
<tr>
<td>ttRestore</td>
<td>Creates a database from a backup that has been created using the ttBackup utility.</td>
<td>No</td>
</tr>
<tr>
<td>ttSchema</td>
<td>Prints out the schema, or selected objects, of a database.</td>
<td>Yes</td>
</tr>
</tbody>
</table>
### Table 5–2  (Cont.) Other utilities descriptions

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Usage with TimesTen Scaleout</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ttSize</strong></td>
<td>Estimates the amount of space that a given table, including any views in the database will consume when the table grows to include a specified number of rows.</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>ttStats</strong></td>
<td>Monitors database metrics or takes and compares snapshots of metrics.</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>ttStatus</strong></td>
<td>Displays information that describes the current state of TimesTen.</td>
<td>No</td>
</tr>
<tr>
<td><strong>ttSyslogCheck (UNIX/Linux)</strong></td>
<td>Determines if the system's <code>/etc/syslog.conf</code> file is properly configured for TimesTen.</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>ttTail</strong></td>
<td>Fetches TimesTen internal trace information from a database and displays it to stdout.</td>
<td>No</td>
</tr>
<tr>
<td><strong>ttTraceMon</strong></td>
<td>Enables and disables the TimesTen internal tracing facilities.</td>
<td>No</td>
</tr>
<tr>
<td><strong>ttUser</strong></td>
<td>Prompts for a password and returns an encrypted password.</td>
<td>None</td>
</tr>
<tr>
<td><strong>ttVersion</strong></td>
<td>Lists the TimesTen release information.</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>ttXactAdmin</strong></td>
<td>Lists ownership, status, log and lock information for each outstanding transaction.</td>
<td>No</td>
</tr>
<tr>
<td><strong>ttXactLog</strong></td>
<td>Displays a formatted dump of the contents of a TimesTen transaction log.</td>
<td>No</td>
</tr>
</tbody>
</table>
ttAdmin

Description

This utility enables you to:

- Specify settings for database loading and RAM loading policies. See "Set database loading policies" on page 5-9.
- Start and stop TimesTen cache agents for caching data from Oracle Database tables. The cache agent is a process that handles Oracle Database access on behalf of a TimesTen database. It also handles the aging and autorefresh of the cache groups in the TimesTen database. See "Set cache policies" on page 5-11.
- Specify settings to automatically or manually start and stop replication agents for specified databases. See "Set replication policies" on page 5-14.
- Specify settings for forced disconnect of all connections. See "Force disconnect" on page 5-16.

Required privilege

This utility requires no privileges to query the database.

Replication options require the ADMIN privilege.

Cache options require the CACHE_MANAGER privilege.

All other options require the ADMIN privilege.

Usage with TimesTen Scaleout

This utility is not supported in TimesTen Scaleout.

Syntax

```
 ttAdmin (-h | -help | -?)
 ttAdmin (-V | -version)
 ttAdmin -query (-connStr connection_string | DSN)

 ttAdmin [-ramPolicy always | manual | inUse [-ramGrace secs]]
 [-ramLoad | -ramUnload]
 [-autoreload | -noautoreload]
 (-connStr connection_string | DSN)

 ttAdmin [-repPolicy always | manual | norestart]
 [-repStart | -repStop]
 [-repQueryThresholdSet secs]
 [-repQueryThresholdGet]
 (-connStr connection_string | DSN)

 ttAdmin [ [-cacheUidGet] |
 [-cacheUidPwdSet -cacheUid uid [-cachePwd pwd]] | [-cachePolicy always | manual | norestart] | [-cacheStart] |
 [-cacheStop [-stopTimeout secs]]
 ]
```
ttAdmin

{ -connStr connection_string | DSN }

ttAdmin [-disconnect urgency [granularity]] { -connStr connection_string | DSN }

urgency:    -transactional | -immediate | -abort

granularity: -users | -unload

Notes

These notes apply to all modes of ttAdmin usage.

Always specify the TimesTen database location as a full path. If a relative path is specified, TimesTen would look relative to the working directory of the daemon, timesten_home/info.

For details on environment variables that you may want to set, see “Environment variables” in *Oracle TimesTen In-Memory Database Installation, Migration, and Upgrade Guide*.

This utility is supported only for TimesTen Data Manager DSNs. It is not supported for TimesTen Client DSNs.

See also

*ttStatus*
*ttCachePolicySet*
*ttCachePolicyGet*
*ttCacheUidPwdSet*
*ttCacheStart*
*ttCacheStop*
Help, version, and query options

**Options**

`ttAdmin` has these options for help, version, and settings information:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`-h</td>
<td>-help`</td>
</tr>
<tr>
<td><code>-?</code></td>
<td></td>
</tr>
<tr>
<td>`-V</td>
<td>-version`</td>
</tr>
<tr>
<td><code>-query</code></td>
<td>Displays a summary of the policy settings for the specified database.</td>
</tr>
<tr>
<td><code>-connStr connection_string</code></td>
<td>For <code>-query</code>, an ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.</td>
</tr>
<tr>
<td><code>DSN</code></td>
<td>For <code>-query</code>, an ODBC data source name of the database to be administered.</td>
</tr>
</tbody>
</table>

**Examples**

To get the version of `ttAdmin`:

```
$ ttAdmin -version
TimesTen Release 18.1.2.1.0
```

To get help for `ttAdmin`:

```
$ ttAdmin -help
Usage:
  ttAdmin [-h | -help | -?]
  ttAdmin [-V | -version]
  ttAdmin [-ramPolicy always | manual | inUse [-ramGrace <secs>]]
  ttAdmin [-disconnect <urgency> [<granularity>]]
    [-autoreload | -noautoreload]
    [-ramLoad | -ramUnload]
    [-repPolicy always | manual | norestart]
    [-repStart | -repStop]
    [-repQueryThresholdSet <seconds>]
    [-repQueryThresholdGet]
  [ {-cacheUidGet} |
    [-cacheUidPwdSet -cacheUid <uid> [-cachePw <pwd>]] |]
  [-cachePolicy always | manual | norestart] |
  [-cacheStart] |
  [-cacheStop [-stopTimeout <secs>]]]

  [-query]
  {<DSN> | [-connstr] <connStr>}
```

**Options:**

[...Option descriptions not shown...]

To display a summary of the policy settings for the database1 DSN:

```
$ ttAdmin -query database1
RAM Residence Policy : inUse
Replication Agent Policy : manual
```
Replication Manually Started        : False
Cache Agent Policy                : manual
Cache Agent Manually Started      : False
Set database loading policies

Options

ttAdmin has these options for database loading and RAM loading:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-connStr connection_string</td>
<td>An ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.</td>
</tr>
<tr>
<td>DSN</td>
<td>An ODBC data source name of the database to be administered.</td>
</tr>
<tr>
<td>-autoreload</td>
<td>-noautoreload</td>
</tr>
<tr>
<td>-ramGrace secs</td>
<td>Only effective if -ramPolicy is inUse. If nonzero, the database is kept in RAM for secs seconds before being unloaded after the last application disconnects from the database.</td>
</tr>
<tr>
<td>-ramLoad</td>
<td>Valid only when -ramPolicy is manual. Causes the database to be loaded into RAM.</td>
</tr>
<tr>
<td>-ramPolicy policy</td>
<td>Defines the policy used to determine when the database is loaded into system RAM.</td>
</tr>
<tr>
<td></td>
<td>inUse (default) - Specifies that the database is loaded in system RAM only when in use (when applications are connected). The -ramGrace option may be used to modify the behavior of this policy. This option cannot be used with temporary databases. (TimesTen only allows a temporary database to be loaded into RAM manually.) Trying to set the policy generates a warning. This policy is not supported in TimesTen Scaleout.</td>
</tr>
<tr>
<td></td>
<td>always - Specifies that the database should remain in system RAM all the time.</td>
</tr>
<tr>
<td></td>
<td>Note: RAM policy always conflicts with forced disconnect granularity unload, described in &quot;Force disconnect&quot; on page 5-16. Do not use these simultaneously.</td>
</tr>
<tr>
<td></td>
<td>manual - Specifies that the database is only to be loaded in system RAM when explicitly loaded by the user (using the ramLoad option).</td>
</tr>
<tr>
<td>-ramUnload</td>
<td>Valid only when -ramPolicy is set to manual. Causes the database to be unloaded from RAM.</td>
</tr>
</tbody>
</table>

Examples

Some performance-sensitive applications use a database referred to by DSN database1. So that applications do not have to wait for the database to be loaded from disk into RAM, this database must always remain in RAM. To accomplish this:

ttAdmin -ramPolicy always database1

The database1 database is typically always resident in RAM. However, it is not being used at a given time and should be loaded only when applications are connected to it. To change the RAM policy:

```
ttAdmin -ramPolicy inUse database1
```
To manually control whether database1 is loaded into RAM and to load it now, use the following.

```
ttAdmin -ramPolicy manual -ramLoad database1
```

Now assume database1 is not always in use. Permanently loading it into RAM would unnecessarily use memory. This database is idle for long periods, but when it is in use multiple users connect to it in rapid succession. To improve performance, it may be best to keep the database in RAM when applications are connected to it and to keep it in RAM for 5 minutes (300 seconds) after the last user disconnects. With this RAM policy, the database remains in RAM if applications are connected to the database. To set this policy:

```
ttAdmin -ramPolicy inUse -ramGrace 300 History
```

Notes

If `ttAdmin` is used with the `-ramPolicy always` option, a persistent system connection is created on the database.

The only `-ramPolicy` value supported for temporary databases is `-ramPolicy manual` with the `-ramLoad` option specified at the same time.
Set cache policies

Options

ttAdmin has these options for cache:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-connStr connection_string</td>
<td>An ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.</td>
</tr>
<tr>
<td>DSN</td>
<td>An ODBC data source name of the database to be administered.</td>
</tr>
<tr>
<td>-cachePolicy</td>
<td>Defines the policy used to determine when the cache agent for the database should run.</td>
</tr>
<tr>
<td></td>
<td>manual (default) - Specifies that the cache agent must be manually started and stopped.</td>
</tr>
<tr>
<td></td>
<td>always - Specifies that the cache agent should always be running for the database. This option immediately starts the cache agent and. When the daemon restarts, the cache agent is restarted.</td>
</tr>
<tr>
<td></td>
<td>norestart - Specifies that the cache agent for the database is not to be restarted after a failure.</td>
</tr>
<tr>
<td>-cacheStart</td>
<td>Starts a cache agent for the database.</td>
</tr>
<tr>
<td>-cacheStop</td>
<td>Stops a cache agent for the database. You should not shut down the cache agent immediately after dropping or altering a cache group. Instead, wait for at least two minutes. Otherwise, the cache agent may not get a chance to clean up the Oracle Database objects that were used by the autorefresh feature.</td>
</tr>
<tr>
<td>-cachePwd</td>
<td>The password associated with the cache administration user ID that manages autorefresh cache groups and asynchronous writethrough cache groups. The cache administration user has extended privileges. See &quot;Grant privileges to the Oracle database users&quot; in Oracle TimesTen Application-Tier Database Cache User's Guide for more details.</td>
</tr>
<tr>
<td>-cacheUid</td>
<td>The cache administration user ID. The cache administration user manages autorefresh cache groups and asynchronous writethrough cache groups. The cache administration user has extended privileges. See &quot;Grant privileges to the Oracle database users&quot; in the Oracle TimesTen Application-Tier Database Cache User's Guide for more details.</td>
</tr>
<tr>
<td>-cacheUidGet</td>
<td>Gets the current cache administration user ID for the specified database.</td>
</tr>
</tbody>
</table>

Examples

A database referred to by DSN database1 contains data cached from an Oracle database. Use the following ttAdmin command to start the cache agent for database1:

```bash
ttAdmin -cacheStart database1
```

You can also use the `-cachePolicy` option to ask the TimesTen data manager daemon to start the cache agent every time the data manager is started:

```bash
ttAdmin -cachePolicy always database1
```

To turn off the automatic start of cache agent:

```bash
ttAdmin -cachePolicy manual database1
```

To set the cache administration user ID and password, use `-cacheUidPwdSet` with `-cacheUid` and `-cachePwd`. For example:

```bash
ttAdmin -cacheUidPwdSet -cacheUid scott -cachePwd tiger database1
```

To get the current cache administration user ID for database1:

```bash
ttAdmin -cacheUidGet database1
```

Notes

Before using any cache features, you must start the cache agent. Cache options require that you specify a value for the `OracleNetServiceName` in the DSN.

When using autorefresh or asynchronous writethrough cache groups, you must specify the cache administration user ID and password. This user account performs autorefresh and asynchronous writethrough operations.

To load data from an Oracle database, the TimesTen cache agent must be running. This requires that the `ORACLE_HOME` environment variable be set to the path of the Oracle installation. See "Managing the cache agent" in Oracle TimesTen Application-Tier Database Cache User's Guide for more details.
See also

- ttStatus
- ttCachePolicySet
- ttCacheUidGet
- ttCacheUidPwdSet
- ttCacheStart
- ttCacheStop
Set replication policies

Options

`ttAdmin` has these options for replication:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-connStr <code>connection_string</code></td>
<td>An ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.</td>
</tr>
<tr>
<td>DSN</td>
<td>An ODBC data source name of the database to be administered.</td>
</tr>
<tr>
<td>-repPolicy</td>
<td>Defines the policy used to determine when the replication agent starts.</td>
</tr>
<tr>
<td></td>
<td>manual (default) - Specifies that the replication agent must be manually started and stopped.</td>
</tr>
<tr>
<td></td>
<td>always - Specifies that the agent should always be running for the database. When the daemon restarts, the replication agent is restarted.</td>
</tr>
<tr>
<td></td>
<td>norestart - Specifies that the replication agent for the database is not to be restarted after a failure.</td>
</tr>
<tr>
<td>-repQueryThresholdGet</td>
<td>Returns the number of seconds that a query can be executed by the replication agent before TimesTen writes a warning to the support log. A value of 0 indicates that no warning is sent.</td>
</tr>
<tr>
<td>-repQueryThresholdSet <code>secs</code></td>
<td>This option specifies the number of seconds that a query can be executed by the replication agent before TimesTen writes a warning to the support log. The specified value takes effect the next time the replication agent starts. The query threshold for the replication agent applies to SQL execution on detail tables of materialized views, <code>ON DELETE CASCADE</code> operations and some internal operations. The value must be greater than or equal to 0. Default is 0 and indicates that no warning is sent.</td>
</tr>
<tr>
<td>-repStart</td>
<td>Starts the replication agent.</td>
</tr>
<tr>
<td>-repStop</td>
<td>Stops the replication agent.</td>
</tr>
</tbody>
</table>

Examples

These examples show use of replication options.

```
- ttAdmin -repPolicy always repl
  RAM Residence Policy : inUse
  Replication Agent Policy : always
  Cache Agent Policy : manual
  Cache Agent Manually Started : False

- ttAdmin -repPolicy manual repl
  RAM Residence Policy : inUse
  Replication Agent Policy : manual
  Replication Manually Started : True
  Cache Agent Policy : manual
  Cache Agent Manually Started : False
```
ttAdmin -repPolicy norestart repl
RAM Residence Policy : inUse
Replication Agent Policy : norestart
Replication Manually Started : True
Cache Agent Policy : manual
Cache Agent Manually Started : False

ttAdmin -repQueryThresholdSet 100 repl
RAM Residence Policy : inUse
Replication Agent Policy : norestart
Replication Manually Started : True
Cache Agent Policy : manual
Cache Agent Manually Started : False

Notes

If ttAdmin is used with -repStart and a replication definition is not found, the replication agent is not started and ttAdmin prints out an error message. For example:

```	tAdmin -repStart repl1
*** [TimesTen][TimesTen 18.1.1.0 ODBC Driver][TimesTen]TT8191:
This store (repl1 on my_host) is not involved in a replication scheme --
file 'eeProc.c', lineno 11016, procedure 'RepAdmin()'
*** ODBC Error = S1000, TimesTen Error = 8191
```

If ttAdmin is used with -repPolicy manual (the default) or -repPolicy always, then the -ramPolicy always option should also be used. This ensures that the replication agent begins recovery after a failure as quickly as possible.

See also

- ttStatus
- ttCachePolicySet
- ttCacheUidGet
- ttCacheUidPwdSet
- ttCacheStart
- ttCacheStop
Force disconnect

Options

`ttAdmin` has these options for forced disconnect:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-connStr connection_string</code></td>
<td>An ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.</td>
</tr>
<tr>
<td>DSN</td>
<td>An ODBC data source name of the database to be administered.</td>
</tr>
<tr>
<td><code>-disconnect urgency [granularity]</code></td>
<td>Asynchronously disconnects all connected applications from the database, optionally including those that are idle or unresponsive.</td>
</tr>
</tbody>
</table>

Acceptable values for `urgency` (required) are:

- `transactional` - Allows any open transactions to be committed or rolled back before disconnecting. Does not affect idle connections.
- `immediate` - Rolls back any open transactions before immediately disconnecting. This also disconnects idle connections.
- `abort` - Aborts all direct mode application processes and `ttcserver` processes in order to disconnect.

A recommended best practice is to run `-disconnect` twice, as necessary. First run it in transactional urgency level. Then, after allowing some time, if not all connections have been closed yet, run it in immediate urgency level. Use `ttStatus` to confirm whether connections have been closed.

Use abort urgency level only as a last resort if transactional and immediate levels do not result in all connections being closed. `Abort may result in loss of data`. Abort abruptly causes every user and `ttcserver` process connected to the database to exit. This may result in lost transactions.

Acceptable values for `granularity` are:

- `users` (default) - Disconnects every user connection to the database. This is useful for administrators who want to perform database maintenance.
- `unload` - Disconnects every connection to the database, including the subdaemon. This is useful to cleanly unload the database.

**Note:** RAM policy always, described in "Set database loading policies" on page 5-9, conflicts with forced disconnect granularity `unload`. Do not use these simultaneously.

Examples

This sample script uses `-disconnect` to disconnect all connections to `database1`, first using transactional urgency level then immediate urgency level.

```bash
#!/bin/sh
#
# disconnect users and unload the database

ttAdmin -disconnect -transactional -unload database1
```
# wait 10 seconds for the disconnects to finish
COUNT = 0
while [ ttStatus | grep "pending disconnection" ] || [ $COUNT -ne 10 ]
do
    sleep 1
    COUNT=$((COUNT+1))
done

# increase urgency to immediate
if [ ttStatus | grep "pending disconnection" ]; then
ttAdmin -disconnect -immediate -unload database1
fi

Use ttStatus to check progress. During forced disconnect, output indicates the pending disconnections:

TimesTen status report as of Wed Jul 18 09:55:20 2018

Daemon pid 10457 port 6627 instance user1
TimesTen server pid 10464 started on port 6629
------------------------------------------------------------------------
Data store /databases/database1
Daemon pid 10457 port 6627 instance user1
TimesTen server pid 10464 started on port 6629
There are 14 connections to the data store, ***14 pending disconnection***

Notes

To enable the capability for forced disconnect, use the TimesTen connection attribute setting ForceDisconnectEnabled=1. See "ForceDisconnectEnabled" on page 2-38.

The -disconnect option is asynchronous. Control will quickly return to the command prompt, but the force disconnect operation may take multiple seconds or even minutes to complete. This is why the scripts above use ttStatus to monitor the status of the force disconnect operation.

The users granularity level includes all connections aside from the subdaemon. For example, in addition to user connections, this includes connections for ttcserver and ttstats.

During the forced disconnect process, any new connection request is rejected by the main daemon. However, after completion of forced disconnect, connection requests are accepted again.
ttAdoptStores

Description

On UNIX and Linux systems, use this utility to move databases from a TimesTen instance to a new TimesTen instance that is of the same major release, but of a different minor release. For example, you can move files from TimesTen 18.1.1.1.0 to TimesTen 18.1.1.2.0.

**Note:** A major release refers to the first three digits of the release number. A minor release refers to the last two digits of the release number.

This utility is useful for testing a minor release of Times with an existing database. You can install the new release of TimesTen and move one or more databases to the new release without uninstalling the old TimesTen release.

You must run the `ttAdoptStores` utility from the destination instance.

Required privilege

This utility must be run by the TimesTen Instance Administrator. The instance administrator must be the same user for both the old and new TimesTen instance.

Usage with TimesTen Scaleout

This utility is not supported in TimesTen Scaleout.

Syntax

```
ttadoptstores {-h | -help | -?}  
ttadoptstores {-V | -version}  
ttadoptstores [-quiet] -dspath path  
ttadoptstores [-quiet] -instpath path  
```

Options

`ttAdoptStores` has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-dspath path</td>
<td>Adopts a single database. The path argument must be the path to the database files (without any file extensions).</td>
</tr>
<tr>
<td>-h</td>
<td>Prints a usage message and exits.</td>
</tr>
<tr>
<td>-help</td>
<td></td>
</tr>
<tr>
<td>-?</td>
<td></td>
</tr>
<tr>
<td>-instpath path</td>
<td>Adopts all databases for an instance. The path argument must be the path to the daemon working directory (infodir). If any databases are in use, the utility fails without making any modifications. No new connections to any database are allowed in the source instance until the entire operation has completed.</td>
</tr>
<tr>
<td>-quiet</td>
<td>Do not return verbose messages.</td>
</tr>
<tr>
<td>-V</td>
<td>-version</td>
</tr>
</tbody>
</table>
### Examples

To adopt the database `/my/data/stores/ds`, use:

```
ttadoptstores -dspath /my/data/stores/ds
```

To adopt all the databases in the directory `/opt/TimesTen/ instance1`, use:

```
ttadoptstores -instpath /opt/TimesTen/instance1
```

### Notes

You cannot adopt temporary databases.

If an instance being adopted is part of a replication scheme, port numbers must match on each side of the replication scheme, unless a port number was specified as the value of the `-remoteDaemonPort` option during a `ttRepAdmin -duplicate` operation. Generally, all instances involved in the replication scheme must be updated at the same time.

This utility does not copy any `sys.odbc.ini` entries. You must move these files manually.
**ttBulkCp**

**Description**
Copies data between TimesTen tables and ASCII files. `ttBulkCp` has two modes:
- In copy-in mode (`ttBulkCp -i`), rows are copied into an existing TimesTen table from one or more ASCII files (or stdin).
- In copy-out mode (`ttBulkCp -o`), an entire TimesTen table is copied to a single ASCII output file (or stdout).

On UNIX and Linux systems, this utility is supported for TimesTen Data Manager DSNs. For Client DSNs, use the utility `ttBulkCpCS`.

This utility only copies out the objects owned by the user executing the utility, and those objects for which the owner has SELECT privileges. If the owner executing the utility has the ADMIN privilege, `ttBulkCp` copies out all objects.

**Required privilege**
This utility requires the INSERT privilege on the tables it copies information into. It requires the SELECT privilege on the tables it copies information from.

If authentication information is not supplied in the connection string or DSN, this utility prompts for a user ID and password before continuing.

**Usage with TimesTen Scaleout**
This utility is supported in TimesTen Scaleout.

**Syntax**
```
 ttBulkCp {-h | -help | -? | -helpfull}
 ttBulkCp {-V | -version}


```
ttBulkCp

[-tsprec precision] [-Q 0|1] [-localOnly]
(-connStr connection_string | DSN) [owner.]tblName
[dataFile]

Options

ttBulkCp has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-Cnone</td>
<td>-Cnone disable the use of comments in the output file. -Cc sets</td>
</tr>
<tr>
<td></td>
<td>the default comment character to c. If no default comment character is specified, the pound character (#) is used. The -C option takes the values: \t (tab) or any of the characters: ! @ # $ % ^ &amp; * ( ) = : ;</td>
</tr>
<tr>
<td>-connStr</td>
<td>An ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.</td>
</tr>
<tr>
<td>DSN</td>
<td>Specifies an ODBC data source name of the database to be copied.</td>
</tr>
<tr>
<td>-D</td>
<td>-dformat</td>
</tr>
<tr>
<td>dataFile</td>
<td>For copy-in mode, specifies the path name(s) of one or more ASCII files containing rows to be inserted into the table. If no files are given, the standard input is used. A single hyphen (-) is understood to mean the standard input. For copy-out mode, specifies the path name of the file into which rows should be copied. If no file is given, the standard output is used. A single hyphen (-) is understood to mean the standard output.</td>
</tr>
<tr>
<td>-dateMode</td>
<td>Specifies whether ttBulkCp treats an Oracle database DATE type as a simple date (without hour, minute and second fields) or as a timestamp (with hour, minute and second fields). For copy-in mode, the default behavior for input is date. For copy-out mode, the default behavior for output is timestamp. TimesTen truncates the data and issues a warning if you select -dateMode date in output mode and one or more date columns have a time component that is not 12:00:00 am. This option overrides the DATEMODE file attribute.</td>
</tr>
<tr>
<td>-directLoad</td>
<td>Selects copy-in mode that copies data from an ASCII file into a database table, but can only be used by a client using a direct connection. Avoids some of the overhead required when using a client/server connection, which provides better performance than the -i mode. Can only be used with TimesTen Classic.</td>
</tr>
<tr>
<td>-h</td>
<td>-help</td>
</tr>
<tr>
<td>-?</td>
<td>Prints a longer usage message and exits.</td>
</tr>
<tr>
<td>-helpfull</td>
<td>Selects copy-in mode that copies data from an ASCII file into a database table. Can be used by a client using either a direct connection or a client/server connection.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>-localonly</td>
<td>This option only loads rows from a specific instance. Load a specific instance in the grid and use this option. When you use this option, <code>ttBulkCp</code> selects all rows from the table, but ignores any rows that are not hashed to the specific instance. This option is only supported in TimesTen Scaleout. The default value is N.</td>
</tr>
<tr>
<td>-N ncharEncoding</td>
<td>Specifies the input and output character encoding for NCHAR types. Valid values are UTF8, UTF-8 or ASCII.</td>
</tr>
<tr>
<td>-o</td>
<td>Selects copy-out mode.</td>
</tr>
<tr>
<td>owner</td>
<td>Specifies the owner of the table to be saved or loaded. If owner is omitted, TimesTen looks for the table under the user’s name and then under the user name SYS. This parameter is case-insensitive.</td>
</tr>
<tr>
<td>-Q [0</td>
<td>1]</td>
</tr>
<tr>
<td>-s c</td>
<td>Sets the default field-separator character to c. If no default field-separator is specified, a comma (,) is used. The -s option takes the values \t (tab) or any of the characters:~ ! @ # % ^ &amp; * ( ) = : ;</td>
</tr>
<tr>
<td>tableName</td>
<td>Specifies the name of the table to be saved or loaded. This parameter is case-insensitive.</td>
</tr>
<tr>
<td>-tformat</td>
<td>Sets the time format. For a list of legal fixed values, see “Date, time and timestamp values” on page 5-29. The default value is ODBC. This option overrides the TSFORMAT file attribute. See also: -D</td>
</tr>
<tr>
<td>timeFormat</td>
<td>Sets the timestamp format. For a list of legal fixed values, see “Date, time and timestamp values” on page 5-29. The default value is DF*TF+FF, which is the concatenation of the date format, the time format and fractional seconds. This option overrides the TFORMAT file attribute. See also: -D</td>
</tr>
<tr>
<td>-tsformat</td>
<td>Sets the timestamp format. For a list of legal fixed values, see “Date, time and timestamp values” on page 5-29. The default value is DF*TF+FF, which is the concatenation of the date format, the time format and fractional seconds. This option overrides the TFORMAT file attribute. See also: -D</td>
</tr>
<tr>
<td>-V</td>
<td>-version</td>
</tr>
<tr>
<td>-v [0</td>
<td>1]</td>
</tr>
</tbody>
</table>

Use the following options in copy-out (`-o`) mode only. You must have SELECT privileges on the specified tables.
Use the following options in copy-in (-i) and directload (-directload) modes only. You must have INSERT privileges on the specified tables.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-A [0</td>
<td>1]</td>
</tr>
<tr>
<td></td>
<td>0 (default) - ttBulkCp may write attribute lines into the output file.</td>
</tr>
<tr>
<td></td>
<td>1 - Suppresses output of attribute lines.</td>
</tr>
<tr>
<td>-forceSerializable</td>
<td>The -forceSerializable option indicates that ttBulkCp should use serializable isolation regardless of the DSN or connection string settings. This is the default behavior.</td>
</tr>
<tr>
<td>-noForceSerializable</td>
<td>-noForceSerializable indicates that ttBulkCp should honor the isolation level in the DSN or connection string.</td>
</tr>
<tr>
<td></td>
<td>If you specify the -noForceSerializable option and the DSN or connection string indicates a non-serializable isolation mode, a warning is included in the output:</td>
</tr>
<tr>
<td></td>
<td>Warning: This output was produced using a non-serializable isolation level. It may therefore not reflect a transaction-consistent state of the table.</td>
</tr>
<tr>
<td>-nullFormat formatStr</td>
<td>Specifies the format in which NULL values are printed. Valid values are:</td>
</tr>
<tr>
<td></td>
<td>null (default) - The word NULL is printed for null fields.</td>
</tr>
<tr>
<td></td>
<td>empty - Nothing is printed for null fields.</td>
</tr>
<tr>
<td></td>
<td>An empty LOB is printed as NULL in no-quotes mode and as ‘’ in quote mode. When copied in, both NULL and ‘’ are interpreted as a NULL LOB.</td>
</tr>
<tr>
<td>-tsprec precision</td>
<td>When used with the -o option, truncates timestamp values to precision. ttBulkCp allows up to 6 digits in the fraction of a second field. Truncation may be necessary when copying timestamps using other RDBMS.</td>
</tr>
</tbody>
</table>

For more information on isolation modes, see "Transaction isolation levels" in Oracle TimesTen In-Memory Database Operations Guide.

Use the following options in copy-in (-i) and directload (-directload) modes only. You must have INSERT privileges on the specified tables.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-cp numTrans</td>
<td>Sets the checkpoint policy for the copy in.</td>
</tr>
<tr>
<td>-cp final</td>
<td>A value of 0 indicates that ttBulkCp should never checkpoint the database, even after the entire copy is complete. A nonzero value indicates that ttBulkCp should checkpoint the database after every numTrans transactions, and again after the entire load is complete. A value of final indicates that ttBulkCp should checkpoint the database only when the entire copy is complete. The default value is 0. Periodic checkpoints can only be enabled if periodic commits are also enabled. See the -xp option.</td>
</tr>
<tr>
<td></td>
<td>NOTE: This option is not supported in TimesTen Scaleout.</td>
</tr>
</tbody>
</table>
-d error
By default, ttBulkCp does not consider rows that are rejected because of constraint violations in a unique column or index to be errors.

-d warn
-d error - Specifies that constraint violations should be considered errors. Duplicate rows are then counted against maxErrs (see -m) and placed into the error file (see -e).

-d warn
-d Specifies that ttBulkCp should copy the offending rows into the error file but should not count them as errors.

-d ignore (default) - Specifies that ttBulkCp should silently ignore duplicate rows.
Regardless of the setting of -d, the duplicate rows are not inserted into the table.

-e errFile
Indicates the name of the file in which ttBulkCp should place information about rows that cannot be copied into the TimesTen table because of errors. These errors include parsing errors, type-conversion errors and constraint violations. The value of errFile defaults to stderr. The format of the error file is the same as the format of the input file (see "Data file format" on page 5-25), so it should be possible to correct the errors in the error file and use the corrected error file as an input file for a subsequent run of ttBulkCp.

-F firstRow
Indicates the number of the first row that should be copied. Use this option (optionally with -L) to copy a subset of rows into the TimesTen table. Rows are numbered starting at 1. If more than one input file is specified, rows are numbered consecutively throughout all the files. The default value is 1.

-L lastRow
Indicates the number of the last row that should be copied. See the description of -F. A value of 0 specifies the last row of the last input file. The default value is 0.

-m maxErrors
Specifies the maximum number of errors to report.
The default is 1.

If set to 0, ttBulkCp returns all error messages. There is no maximum limit.

-S error
By default, ttBulkCp issues an error when it encounters a value that exceeds its maximum scale. This error can be generated for a decimal value whose scale exceeds the maximum scale of its column or for a TIMESTAMP value with more than 6 decimal places of fractional seconds (sub-microsecond granularity).

-S error (default) - Specifies that ttBulkCp should not insert a row containing a value that exceeds its maximum scale into the table and that it should place an error into the error file.

-S warn - Specifies that ttBulkCp should right-truncate the value to its maximum scale before inserting the row into the table and that it should place a warning into the error file.

-S ignore - Specifies that ttBulkCp should silently right-truncate the value to its maximum scale before inserting the row into the table.
Data file format

This section describes the format the `dataFile` parameter.

Each line of a `ttBulkCp` input file is either a blank line, a comment line, an attribute line or a data line.

- Blank lines are lines with no characters at all, including whitespace characters (space and tab). Blank lines are ignored by `ttBulkCp`.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-t error</code></td>
<td>By default, <code>ttBulkCp</code> issues an error when a CHAR, VARCHAR2, NCHAR, NVARCHAR2, BINARY, VARBINARY, BLOB, CLOB, or NLOB value is longer than its maximum column width.</td>
</tr>
<tr>
<td><code>-t warn</code></td>
<td>Specifies that long string or binary attributes should not be inserted into the TimesTen table and that an error should be placed into the error file.</td>
</tr>
<tr>
<td><code>-t ignore</code></td>
<td>Specifies that long string or binary attributes should be truncated to the maximum column length before being inserted into the table but that a warning should be placed into the error file.</td>
</tr>
</tbody>
</table>

- `[no]tblLock` Specifies whether to use table-level or row-level locking, when copying rows into a TimesTen table.
  - `-tblLock` - Indicates table-level locking. This is the default.
  - `-notblLock` - Indicates row-level locking.

  For a single input stream into a table, using `-tblLock` is most efficient. Using `-notblLock` provides some performance benefit if you use multiple concurrent `ttBulkCp` sessions to insert into a single table in parallel.

- `-u error` By default, `ttBulkCp` issues an error when a real, float or double attribute underflows. Underflow occurs when a floating point number is so small that it is rounded to zero.
- `-u warn` Specifies that 0.0 should be inserted for real, float or double attributes that underflow, but that a warning should be placed into the error file.
- `-u ignore` Specifies that 0.0 should be silently inserted for real, float or double attributes that underflow.

- `-xp numRows` Sets the transaction policy for the load. A value of 0 indicates that `ttBulkCp` should perform the entire load as a single transaction and should commit that transaction whether the load succeeds or fails.

  A value of `rollback` indicates that `ttBulkCp` should perform the entire load as a single transaction and should roll that transaction back if the load fails.

  A nonzero value indicates that `ttBulkCp` should commit after every `numRows` processed rows.

  The default value is 1024.

  Use the `-xp` option with the `-cp` option to enable periodic checkpointing of the database.
Comment lines begin with the comment character. The default comment character is #; this default can be overridden with the \-C command-line option or the COMMENTCHAR file attribute (see "File attribute line format" on page 5-26). The comment character must be the first character on the line. Comment lines are ignored by ttBulkCp. Comments at the end of data lines are not supported.

File attribute lines are used for setting file attributes that control the formatting of the data file. Attribute lines begin with the ten-character sequence ##ttBulkCp. The section "File attribute line format" on page 5-26 describes the full syntax for attribute lines. Attribute lines can appear anywhere in the data file.

Data lines contain the rows of the table being copied. Data lines in the data file and rows of the table correspond one-to-one; that is, each data line completely describes exactly one row. Each data line consists of a list of column values separated by the field separator character. The default field separator is a comma (,). This default can be overridden by the \-s command-line option or the FSEP file attribute. The section "Data line format" on page 5-27 describes the full syntax for data lines.

**File attribute line format**

The format of an attribute line is:

\##ttBulkCp[:attribute=value]...

Attribute lines always begin with the ten-character sequence ##ttBulkCp, even if the comment character is not #. This sequence is followed by zero or more file attribute settings, each preceded by a colon.

File attribute settings remain in effect until the end of the input file or until they are changed by another attribute line in the same input file. The values of any file attributes that are omitted in an attribute line are left unchanged.

Most command line options take precedence over the values in the file attributes that are supported by ttBulkCp. The CHARACTERSET attribute is the only file attribute that overrides command line options.

The file attributes are:

- **CHARACTERSET**: Specifies the character set to be used to interpret the data file. If the file attribute is not set, the character set used to interpret the file is the one specified in the ConnectionCharacterSet connection attribute. For best performance, the value of the DatabaseCharacterSet connection attribute should match either the ConnectionCharacterSet connection attribute or this file attribute. If the character set supplied in ConnectionCharacterSet connection attribute or in this file attribute is different than the actual character set of the file, ttBulkCp may interpret data incorrectly.

- **VERSION**: Specifies the version of the file format used in the file, expressed as major.minor. The only supported version is 1.0.

- **DATEMODE**: Specifies whether an Oracle database DATE type is specified as simple date or as timestamp.

- **FSEP**: Specifies the field separator character used in the file. The field separator can be set to \t (tab) or any of the characters: - ! @ # $ % ^ & * ( ) = : ; | < > ? , /.

- **QUOTES**: Indicates whether character string values in the file are enclosed in double quotes. The value can be 0, to indicate that strings are not quoted, or 1, to indicate that strings are quoted. This value can be overridden with the \-Q option.
**COMMENTCHAR**: Specifies the comment character used in the file. The comment character can be set to \t (tab) or any of the characters: `~ ! @ # $ % ^ & * ( ) = : ; | < > ? , /`

The comment character can also be set to the value `none`, which disables the use of comments in the data file.

**DFORMAT**: Sets the date format. For a list of legal values, see "Date, time and timestamp values" on page 5-29. When a custom format is used, it should be enclosed in single quotes. This value can be overridden with the `-D/-dformat` command-line option. See also: TFORMAT and TSFORMAT.

**NCHARENCODING**: Indicates the encoding to be used for the NCHAR and NVARCHAR2 data types. The value may be either `ASCII` or `UTF-8`.

**TFORMAT**: Indicates the time format. For a list of legal values, see "Date, time and timestamp values" on page 5-29. When a custom format is used, it should be enclosed in single quotes. This value can be overridden with the `-tformat` command-line option. See also: DFORMAT and TSFORMAT.

**TSFORMAT**: Sets the timestamp format. For a list of legal values, see "Date, time and timestamp values" on page 5-29. When a custom format is used, it should be enclosed in single quotes. This value can be overridden with the `-tsformat` command-line option. See also: DFORMAT and TFORMAT.

### Examples

The following header line sets the field separator character to `$` and disables quoting of character strings:

```
##ttBulkCp:FSEP=$:QUOTES=0
```

The following header line disables comments and sets the date format to the Oracle format:

```
##ttBulkCp:COMMENTCHAR=none:DFORMAT=Oracle
```

The following header line set the date format to a custom format:

```
##ttBulkCp:DFORMAT='Mon DD, YYYY'
```

### Data line format

Data lines contain the row data of the table being copied. Each data line corresponds to a row of the table; rows cannot span input-file lines. A data line consists of a list of column values separated by the field separator character. Unnecessary whitespace characters should not be placed either before or after the field separator. The format of each value is determined by its type.

**NULL values**

NULL values can either be expressed as `NULL` (all capitals, no quotes) or as empty fields.

**Character and unicode strings**

CHAR, VARCHAR2, NCHAR, NVARCHAR2, CLOB, NCLOB: If quoting of character strings is enabled (the default), then strings and characters must be enclosed in double quotes. If quoting of character strings is disabled, then any double-quote characters in the string are considered to be part of the string itself. `ttBulkCp` recognizes the following backslash escapes inside a character string, regardless of whether quoting of strings is enabled:
• " The double-quote character. If character-string quoting is enabled, then all double quote characters in the string must be escaped with a backslash. If character-string quoting is disabled, then it is permissible, but not necessary, to use the backslash.

• \t The tab character.

• \n The newline character.

• \r The carriage return character.

• \ The backslash character.

• \xyz (CHAR and VARCHAR2 only) The character whose ASCII value is \xyz, where \xyz is a three-character octal number, as in \033.

• \uxyzw (NCHAR and NVARCHAR2 only) The character whose unicode value is \uxyzw, where \uxyzw is a four-digit hexadecimal number, as in \ufe4a. The \uxyzw notation is supported in both UTF-8 and ASCII encoding modes.

In addition, any of the ~ ! @ # $ % ^ & * ( ) = : ; | < > ? , / characters can be escaped with a backslash. Although it is unnecessary to escape these characters usually, doing so prevents them from being mistaken for a comment character or a field separator when character-string quoting is disabled.

If character-string quoting is enabled, the empty string (represented as " " ) is distinct from NULL. If character-string quoting is disabled, then empty strings cannot be represented, as they cannot be distinguished from NULL.

For unicode strings, unicode characters encoded using UTF-8 multibyte sequences are supported in the UTF-8 encoding mode only. If these sequences are used with the ASCII encoding mode, ttBulkCp interprets each byte in the sequence as a separate character.

For fixed-length CHAR and NCHAR fields, strings that are shorter than the field length are padded with blanks. For VARCHAR2 and NVARCHAR2 fields, the string is entered into TimesTen exactly as given in the data file. Trailing blanks are neither added nor removed.

**Binary values**

BINARY, VARBINARY, BLOB: If quoting of character strings is enabled (the default), binary values are delimited by curly braces ({...}). If quoting of character strings is disabled, then curly braces should not be used. Whether character-string quoting is enabled or disabled, binary values may start with an optional 0x or 0X.

Each byte of binary data is expressed as two hexadecimal digits. For example, the four-byte binary string:

01101000 11001010 01001001 11101111

would be expressed as the eight-character hexadecimal string:

68CA49EF

Digits represented by the letters A through F can either be upper- or lower-case. The hexadecimal string cannot contain white spaces. Because each pair of characters in the hexadecimal string is converted to a single binary byte, the hexadecimal string must contain an even number of characters. For fixed-length binary fields, if the given value is shorter than the column length, the value is padded with zeros on the right. For VARBINARY values, the binary value is inserted into TimesTen exactly as given in the data file.
If character-string quoting is enabled, a zero-length binary value (represented as `{ }`) is distinct from NULL. If character-string quoting is disabled, then zero-length binary values cannot be represented, as they cannot be distinguished from NULL.

**Integer values**

TINYINT, SMALLINT, INTEGER, BIGINT: Integer values consist of an optional sign followed by one or more digits. Integer values may not use E-notation. Examples:

-14 98765 +186

**Floating-point values**

REAL, FLOAT, DOUBLE: Floating-point values can be expressed with or without decimal points and may use E-notation. Examples:

3.1415
-0.00004
1.1e-3
5e3
.56
-682
-.62E-4
170.

**Fixed-point values**

DECIMAL, NUMERIC: Decimal values can be expressed with or without decimal points. Decimal values may not use E-notation. Examples:

5
-19.5
-11
000
-.1234
45.
-57.0
0.8888

**Inf, -Inf and NaN values**

Inf, -Inf and NaN values: Infinity and Not a Number values can be represented as strings to represent the corresponding constant value (all are case insensitive):

<table>
<thead>
<tr>
<th>String</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>NAN</code></td>
<td>NaN</td>
</tr>
<tr>
<td><code>[+]*INF</code></td>
<td>Inf</td>
</tr>
<tr>
<td><code>INF</code></td>
<td>-Inf</td>
</tr>
</tbody>
</table>

TimesTen outputs the values as: `NAN`, `INF` and `-Inf`.

**Date, time and timestamp values**

Formats for date, time and timestamp values can be specified either by selecting a fixed datetime format or by defining a custom datetime format. The custom datetime formats are defined using format specifiers similar to those used by the `TO_DATE` and `TO_CHAR` SQL functions, as described in the following table.

In many cases, it is not necessary to define the timestamp format, even when a custom date or time format is used, because the default TimesTen format (`DF*TF+FF`) is defined
in terms of the date and time formats. Therefore, setting the date format sets not only the format for date values, but also for the date portion of timestamp values. Similarly, setting the timestamp format affects both time values and the time portion of the timestamp values.

<table>
<thead>
<tr>
<th>Specifier</th>
<th>Descriptions and restrictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q</td>
<td>Quarter. Cannot be used in copy-in mode.</td>
</tr>
<tr>
<td>YYYY</td>
<td>Year (four digits).</td>
</tr>
<tr>
<td>Y,YYY</td>
<td>Year (with comma as shown).</td>
</tr>
<tr>
<td>YYY</td>
<td>Year (last three digits). Cannot be used in copy-in mode.</td>
</tr>
<tr>
<td>Y</td>
<td>Year (last digit). Cannot be used in copy-in mode.</td>
</tr>
<tr>
<td>MONTH</td>
<td>Month (full name, blank-padded to 9 characters, case-insensitive).</td>
</tr>
<tr>
<td>NON</td>
<td>Month (three character prefix, case-insensitive).</td>
</tr>
<tr>
<td>MM</td>
<td>Month (01 through 12).</td>
</tr>
<tr>
<td>DD</td>
<td>Day of the month (01 through 31).</td>
</tr>
<tr>
<td>HH24</td>
<td>Hour (00 through 23).</td>
</tr>
<tr>
<td>HH12</td>
<td>Hour (01 through 12). Must be used with AM/PM for copy-in mode.</td>
</tr>
<tr>
<td>HH</td>
<td>Hour (01 through 12). Must be used with AM/PM for copy-in mode.</td>
</tr>
<tr>
<td>NI</td>
<td>Minute (00 through 59).</td>
</tr>
<tr>
<td>SS</td>
<td>Second (00 through 59).</td>
</tr>
<tr>
<td>FF</td>
<td>Fractional seconds. Six digits, unless overridden with the -tsprec option.</td>
</tr>
<tr>
<td>FFn</td>
<td>Fractional seconds (number of digits specified by n).</td>
</tr>
<tr>
<td>+FF</td>
<td>In copy-in mode, matches, optional decimal point plus one or more fractional seconds. In copy-out mode, same as .FF.</td>
</tr>
<tr>
<td>+FFn</td>
<td>In copy-in mode, same as +FF. In copy-out mode, same as .FFn.</td>
</tr>
<tr>
<td>AM PM</td>
<td>Meridian indicator without dots. In copy-in mode, this must be used with HH or HH12, but not HH24.</td>
</tr>
<tr>
<td>A.M. P.M.</td>
<td>Meridian indicator with dots. In copy-in mode, this must be used with HH or HH12, but not HH24.</td>
</tr>
<tr>
<td>DF</td>
<td>Current date format (can only be used in timestamp format).</td>
</tr>
<tr>
<td>TF</td>
<td>Current time format (can only be used in timestamp format).</td>
</tr>
<tr>
<td>= / ; :</td>
<td>Punctuation that are matched in copy-in mode or output in copy-out mode.</td>
</tr>
<tr>
<td>&quot;text&quot;</td>
<td>Text that is matched in input mode or output in copy-out mode.</td>
</tr>
<tr>
<td>*</td>
<td>Matches 0 or more whitespace characters (space or tab) in copy-in mode or outputs 1 space in copy-out mode.</td>
</tr>
</tbody>
</table>

**Fixed date, time and timestamp formats**

For date values, the fixed formats are:
### For time values, the only fixed format is ODBC:

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODBC</td>
<td>HH24:MI:SS</td>
<td>07:47:23</td>
</tr>
</tbody>
</table>

### For timestamp values, the fixed formats are:

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODBC</td>
<td>YYYY-MM-DD*HH24:MI:SS+FF</td>
<td>2011-01-03 07:47:23</td>
</tr>
<tr>
<td>Oracle</td>
<td>DD-Mon-YYYY*HH24:MI:SS+FF</td>
<td>03-Jan-2011 07:47:23</td>
</tr>
<tr>
<td>SYBASE1</td>
<td>MM/DD/YYYY*HH24:MI:SS+FF</td>
<td>01/03/2011 07:47:23</td>
</tr>
<tr>
<td>SYBASE2</td>
<td>DD-MM-YYYY*HH24:MI:SS+FF</td>
<td>03-01-2011 07:47:23</td>
</tr>
<tr>
<td>SYBASE3</td>
<td>Mon<em>DD</em>YYYY*HH24:MI:SS+FF</td>
<td>Jan 03 2011 07:47:23</td>
</tr>
</tbody>
</table>

The default timestamp value is: 'DF*TF+FF'

---

### Examples

The following input file is for a table with five columns: two char columns, a double column, an integer column and a VARBINARY column. In the "Mountain View" line, the last three columns have NULL values.

```sql
# This is a comment.
##### So is this.
# The following line is a blank line.
```
"New York","New York",-345.09,12,(12EF87A4E5)
"Milan","Italy",0,0,(0x458F)
"Paris","France",1.4E12,NULL,(F009)
"Tokyo","Japan",-4.5E-18,26,(0x00)
"Mountain View","California",,

Here is an equivalent input file in which quotes are disabled, the comment character is '!' and the field separator is '|':

```sql
##ttBulkCp:QUOTES=0:COMMENTCHAR=$:FSEP=|
$ This is a comment.
$$$$$$ So is this.
$ The following line is a blank line.

New York|New York|-345.09|12|12EF87A4E5
Milan|Italy|0|0|0x458F
Paris|France|1.4E12|NULL|F009
Tokyo|Japan|-4.5E-18|26|0x00
Mountain View|California|||
```

The following command dumps the contents of table mytbl from database mystore into a file called mytbl.dump.

```
ttBulkCp -o mystore mytbl mytbl.dump
```

The following command loads the rows listed in file mytbl.dump into a table called mytbl on database mystore, placing any error messages into the file mytbl.err.

```
ttBulkCp -i -e mytbl.err mystore mytbl mytbl.dump
```

The above command terminates after the first error occurs. To force the copy to continue until the end of the input file (or a irrecoverable error), use `-m 0`, as in:

```
ttBulkCp -i -e mytbl.err -m 0 mystore mytbl mytbl.dump
```

To ignore errors caused by constraint violations, use `-d ignore`, as follows.

```
ttBulkCp -i -e mytbl.err -d ignore mystore mytbl mytbl.dump
```

**Notes**

`ttBulkCp` explicitly sets the `Overwrite` connection attribute to 0, to prevent accidental destruction of a database. For more information, see “Overwrite” on page 2-47.

Real, float or double values may be rounded to zero when the floating point number is small.

The connection attribute `PassThrough` with a nonzero value is not supported in this utility and returns an error.

When specifying date, time and timestamp formats, incomplete or redundant formats are not allowed in input mode. Specifiers that reference fields that are not present in the data type (for example a minute specifier in a date format) return errors in copy-out mode. In copy-in mode, the values of those specifiers are ignored.

The following caveats apply when disabling quoted strings in the `ttBulkCp` data file:

- Empty strings and zero-length binary values cannot be expressed, as they cannot be distinguished from `NULL`.
- If the field separator character appears inside a character string, it must be escaped with a backslash or else it is treated as an actual field separator.
If a data line begins with a character string and that string begins with the
comment character, that character must be escaped with a backslash or else the
line is treated as a comment. If there are no actual comments in the file, set the
comment character to none to avoid characters from being misread as comment
characters.

For UTF-8, NCHAR are converted to UTF-8 encoding and then output. UTF-8 input is
converted to NCHAR.

For ASCII, those NCHAR values that correspond to ASCII characters are output as ASCII.
For those NCHAR values outside of the ASCII range, the escaped Unicode format is used.

This utility is for use specifically with TimesTen tables. It is not supported with
passsthrough to an Oracle database.

On Windows, this utility is supported for all TimesTen Data Manager and Client
DSNs.

It is recommended that you do not run DDL SQL commands while running ttBulkCp
to avoid lock contention issues for your application.

See also

- ttBackup
- ttMigrate
- ttRestore
ttBackup

Description
Creates a backup copy of a database that can be restored at a later time using the ttRestore utility.

For an overview of the TimesTen backup and restore facility, see "Backup, Restore, and Migrate Data in TimesTen Classic" in Oracle TimesTen In-Memory Database Installation, Migration, and Upgrade Guide.

Required privilege
This utility requires the ADMIN privilege.

If authentication information is not supplied in the connection string or DSN, this utility prompts for a user ID and password before continuing.

Usage with TimesTen Scaleout
This utility is not supported in TimesTen Scaleout.

Syntax

```
ttBackup {-h | -help | -?}  
ttBackup {-V | -version}  
ttBackup -dir directory [-type backupType]  
[-fname fileprefix] [-force]  
{-connStr connection_string | DSN}
```

Options

```
ttBackup has the options:
```

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-connStr connection_string</code></td>
<td>An ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.</td>
</tr>
<tr>
<td><code>DSN</code></td>
<td>Specifies an ODBC data source name of the database to be backed up.</td>
</tr>
<tr>
<td><code>-dir directory</code></td>
<td>Specifies the directory where the backup files should be stored.</td>
</tr>
<tr>
<td><code>-fname fileprefix</code></td>
<td>Specifies the file prefix for the backup files in the backup directory. The default value for this option is the file name portion of the DataStore parameter of the database's ODBC definition.</td>
</tr>
<tr>
<td><code>-force</code></td>
<td>Forces the backup into the specified directory. If a backup exists in that directory, ttBackup overwrites it. If this option is not specified, and you are creating a backup from a database other than the one previously backed up in the specified directory, ttBackup terminates with an end message without overwriting existing files.</td>
</tr>
<tr>
<td><code>-h -help -?</code></td>
<td>Prints a usage message and exits.</td>
</tr>
</tbody>
</table>
Examples

To perform a full file backup of the FastIns database to the backup directory in /users/pat/ TimesTen/backups, use:

```
ttBackup -type fileFullEnable -dir /users/pat/TimesTen/backups FastIns
```

To copy the FastIns database to the file FastIns.back, use:

```
ttBackup -type streamFull FastIns > FastIns.back
```

On UNIX and Linux systems, to save the FastIns database to a backup tape, use:

```
ttBackup -type streamFull FastIns | dd bs=64k of=/dev/rmt0
```

To back up a database named origDSN to the directory /users/rob/tmp and restore it to the database named restoredDSN, use:

```
ttBackup -type fileFull -dir /users/rob/tmp -fname restored origDSN
ttRestore -dir /users/rob/tmp -fname restored restoredDSN
```

Notes

The ttBackup utility and the ttRestore utility backup and restore databases only when the two parts of the TimesTen release and the platform are the same. For example, you can back up and restore files between TimesTen releases 18.1.1.1.0 and 18.1.2.10. You cannot backup and restore files between releases 11.2.2.8.35 and 18.1.2.1.0. You can use the ttBulkcp or CS (UNIX and Linux only) utility to migrate databases across major releases or operating systems. You can use ttMigrate together with ttMigrateCS (client server version of ttMigrate) to migrate databases between 32- and 64-bit platforms or bit levels. You must use the -relaxedUpgrade option when...
restoring data on a new bit-level. In the case of changing bit-levels, the database cannot be involved in a replication scheme.

Follow the examples in "Moving a database between 32-bit and 64-bit platforms" in the Oracle TimesTen In-Memory Database Installation, Migration, and Upgrade Guide.

When an incremental backup has been enabled, TimesTen creates a backup hold in the transaction log file. Call the ttLogHolds built-in procedure to see information about this hold. The backup hold determines which log records should be backed up upon subsequent incremental backups. Only changes since the last incremental backup are updated. A side effect to creating the backup hold is that it prevents transaction log files from being purged upon a checkpoint operation until the hold is advanced by performing another incremental backup or removed by disabling incremental backups.

Transactions that commit after the start of the backup operation are not reflected in the backup.

Up to one checkpoint and one backup may be active at the same time, with these limitations:

- A backup never needs to wait for a checkpoint to complete.
- A backup may need to wait for another backup to complete.
- A checkpoint may need to wait for a backup to complete.

Databases containing cache groups can be backed up as normal with the ttBackup utility. However, when restoring such a backup, special consideration is required as the restored data within the cache groups may be out of date or out of sync with the data in the back end Oracle database. See the section on "Backing up and restoring a database with cache groups" in the Oracle TimesTen Application-Tier Database Cache User’s Guide for details.

You cannot back up temporary databases.

See also

- ttBulkCp
- ttMigrate
- ttRestore
ttCapture

Description
Captures information about the state of TimesTen at the time the command is used. This information may be useful in diagnosing problems. Sometimes TimesTen Customer Support must make repeated incremental requests for information to diagnose a customer’s problem in the field.

The information captured by this utility may be requested by TimesTen Customer Support and may be sent with your support email.

The utility does not interpret errors. It only collects information about the state of things and sends output to the ttcapture.date.number.log file in the directory from which you invoke the ttCapture utility. This utility collects general information that is usually relevant to support cases.

Note: You should always enclose directory and file names in double quotes, in case there are spaces in them.

Required privilege
This utility requires the instance administrator privilege.

If authentication information is not supplied in the connection string or DSN, this utility prompts for a user ID and password before continuing.

Usage with TimesTen Scaleout
This utility is supported in TimesTen Scaleout.

Syntax

```
 ttCapture {-h | -help | -?} 
 ttCapture {-V | -version} 
 ttCapture [-noinstinfo] [-nosysinfo] [-stdout | -dest dir] [-logdir dir] 
          [dspath | DSN] 
 ttCapture [-noinstinfo] [-nosysinfo] [-stdout | -dest dir] [-logdir dir] 
         [-noconnect] [dspath | DSN] 
 ttCapture -noconnect [dspath | DSN] 
```

Options

```
 ttCapture has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-dest dir</td>
<td>Writes the output file to the designated directory.</td>
</tr>
<tr>
<td>DSN</td>
<td>Specifies an ODBC data source name of the database to be checked.</td>
</tr>
</tbody>
</table>
```
Examples

To capture data on the test_db database and write the database checkpoint files to the directory D:\my_data\recover\test_db, use:

```
ttCapture -dest "D:\my_data\recover\test_db" test_db
```

Notes

This utility is supported only where the TimesTen Data Manager is installed.
ttCheck

Description

Performs internal consistency checking within a TimesTen database. You can specify a specific structure to be checked and a desired level of checking.

Required privilege

This utility requires the ADMIN privilege.

If authentication information is not supplied in the connection string or DSN, this utility prompts for a user ID and password before continuing.

Usage with TimesTen Scaleout

This utility is supported in TimesTen Scaleout.

Syntax

```
ttCheck (+h | -help | -?)
.ttCheck (+V | -version)
[-permBlkDir] [-permHeap] [-tempBlkDir] [-tmpHeap]
[-tables tblName [...] [-users userName [...] [-level levelNum] [...] [-m maxErrors] [-f outFile] [-v verbosity]
(DSN | [-connstr] connection_string | dspath)
```

Options

ttCheck has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-blkDir</td>
<td>Checks all the block directories.</td>
</tr>
<tr>
<td>-compHeap</td>
<td>Checks the compilation heap structure.</td>
</tr>
<tr>
<td>-connStr connection_string</td>
<td>An ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.</td>
</tr>
<tr>
<td>DSN</td>
<td>Specifies an ODBC data source name of the database to be checked.</td>
</tr>
</tbody>
</table>
| dspath                  | The fully qualified name of the database to be checked. This is not the DSN associated with the connection. It is the fully qualified database path name associated with the database as specified in the DataStore= parameter connection attribute in the database’s DSN.  
  For example, for a database consisting of files /home/payroll/2011.ds0, /home/payroll/2011.ds1, and several transaction log files /home/payroll/2011.logn, dspath is /home/payroll/2011. |
<p>| -f outFile              | Specifies the output file name; defaults to stdout.                         |
| -h                      | Prints a usage message and exits.                                           |
| -help                   |                                                                             |
| -?                      |                                                                             |</p>
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-header</td>
<td>Checks the content of the database header.</td>
</tr>
<tr>
<td>-heap</td>
<td>Checks all heap structures.</td>
</tr>
<tr>
<td>-indexHeap</td>
<td>Checks the index heap structure.</td>
</tr>
</tbody>
</table>
| -level levelNum | Indicates the level of checking for header, block directory, heap and table. Different structures can be checked using different levels in a same command. A level specification is applied to all structures specified to its left in the command string that do not have a level specification. A level specification is applied to all structures if no structure is specified in the command string.  
1 - Checks sanity bytes and simple fields. For example, counts enums for validity in all high-level structures.  
2 - Does all checks in level 1, plus checks the validity of structures, referenced by fields in other structures.  
3 - Does all checks in level 2, plus checks each table row for column values. For example, checks valid VARCHAR2 and FLOAT sizes.  
4 (default) - Does all checks in level 3, plus checks index/table mapping for each row and each index. |
| -log            | Checks the log buffer.                                                     |
| -m maxErrors    | Maximum number of errors to report. Default is 10; a few extra related errors may be reported. If 0, the utility only connects, then returns. |
| -permBlkDir     | Checks the permanent partition block directory.                            |
| -permHeap       | Checks the permanent heap structure.                                        |
| -tables tblName [...] | Checks table(s) specified by tblName.                                     |
| -tempBlkDir     | Checks the temporary partition block directory.                            |
| -tmpHeap        | Checks the temporary heap structure.                                       |
| -users userName [...] | Checks tables belonging to the user(s) specified by userName.          |
| -V | -version         | Prints the release number of ttCheck and exits.                           |
| -v verbosity    | 0 - No output (program’s exit status indicates if an error was found).  
1 (default) - Enable error output only.  
2 - Error output and a progress report. |

**Examples**

To perform a check of all structures in the test_db database, use:

```
ttCheck test_db
```

To perform a sanity check of all structures in the test_db database, use:

```
ttCheck -level 1 test_db
```

To perform a check of all tables in the test_db database, use:

```
ttCheck -tables test_db
```

To check the physical structures and row contents of all tables in the test_db database, use:
ttCheck -tables -level 3 test_db

To perform a sanity check of all heap structures, row contents and indexes of all tables in the test_db database, use the following.

```
ttCheck -heap -level 1 -tables -level 4 test_db
```

To check the physical structures and row contents of tables tab1 and tab2 in the test_db database, use:

```
ttCheck -tables tab1 tab2 -level 3 test_db
```

**Notes**

While primarily intended for use by TimesTen customer support to diagnose problems with internal data structures of a TimesTen database, the information returned by ttCheck may be useful to system administrators and developers.

The ttCheck utility should be run when there are no active transactions on the system.

The ttCheck utility checks views in the same manner as other tables in a database. The utility cannot verify that the contents of a view matches view query’s result.

If no structures are specified, ttCheck checks all structures. No errors are returned if a specified table's name or user is not found.

This utility may take some time to run. Verbosity level 2 enables you to print a progress report.

This utility is supported only where the TimesTen Data Manager is installed.
ttCWAdmin

Description
Manages TimesTen active standby pairs that take advantage of the high availability framework of Oracle Clusterware. This utility starts administrative processes, generates scripts and performs other functions to administer active standby pairs and the corresponding Clusterware resources.

For more information about using Oracle Clusterware to manage TimesTen active standby pairs, see Oracle TimesTen In-Memory Database Replication Guide.

These commands are available only with advanced high availability:
- ttCWAdmin -addMasterHosts
- ttCWAdmin -addSubscriberHosts
- ttCWAdmin -createVIPs
- ttCWAdmin -delMasterHosts
- ttCWAdmin -delSubscriberHosts
- ttCWAdmin -dropVIPs

These commands fail with basic high availability.

Required privilege
On Windows, any user with Administrators privileges can execute all commands in this utility.

On UNIX and Linux systems, the root user can execute all commands in this utility. These commands must be executed by the root user:
- ttCWAdmin -addMasterHosts
- ttCWAdmin -addSubscriberHosts
- ttCWAdmin -createVIPs
- ttCWAdmin -delMasterHosts
- ttCWAdmin -delSubscriberHosts
- ttCWAdmin -ocrConfig
- ttCWAdmin -dropVIPs

The administrator user can execute all other commands in this utility.

If authentication information is not supplied in the connection string or DSN, this utility prompts for a user ID and password before continuing.

Usage with TimesTen Scaleout
This utility is not supported in TimesTen Scaleout.

Syntax

ttCWAdmin {-h | -help | -?}

ttCWAdmin {-V | -version}
ttCWAdmin -init [-hosts "host_name1, host_name2[,...]"]

ttCWAdmin (-createVIPs | -dropVIPs | -create | -drop | -restore | -start | -stop | -status) [-ttclusterini path] [-dsn DSN]

ttCWAdmin - [-timeout seconds] -dsn DSN

Options

**ttCWAdmin has these options:**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-addMasterHosts</td>
<td>Adds spare hosts to the pool of master hosts dynamically, when high availability is employed. On the command line, separate multiple host names by commas. On UNIX and Linux systems, only the root user can execute this command.</td>
</tr>
<tr>
<td>-addSubscriberHosts</td>
<td>Adds spare hosts to the pool of subscriber hosts dynamically, when high availability is employed. On the command line, separate multiple host names by commas. On UNIX and Linux systems, only the root user can execute this command.</td>
</tr>
</tbody>
</table>
### Option Description

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-beginAlterSchema</td>
<td>Enables manual alteration, addition or dropping of cache groups to the active standby pair replication scheme when automatic include of new schema objects in the active standby pair scheme is not possible. Also, enables creation of PL/SQL procedures, sequences materialized views and indexes on tables with data. Enables addition of a read-only subscriber that is not managed by Oracle Clusterware. While adding objects to the schema, the active standby pair is brought down. See also: -endAlterSchema.</td>
</tr>
</tbody>
</table>
| -create                | Creates the active standby pair replication scheme for the specified DSN and creates the associated action scripts. This command:  

  - Prompts for the name of a TimesTen internal user with `ADMIN` privileges. TimesTen uses this internal user to create the active standby pair. If cache groups are being managed by Oracle Clusterware (if the attribute `CacheConnect=Y` in the `cluster.oracle.ini`), enter the TimesTen cache manager user name.  
  - Prompts for the TimesTen password for the previously entered user name.  
  - If cache groups are being used, prompts for the password for the Oracle database user that has the same name as the cache manager. This password is provided in the `OraclePWD` connection attribute when the cache manager user connects. This Oracle database user is used to set the autorefresh states for cache groups.  
  - Prompts for a random string used to encrypt the above information. |
| -createVIPs            | Creates virtual IP addresses for the active standby pair. If no DSN is specified, displays the information of all active standby pairs managed under the same TimesTen instance administrator and TimesTen instance name managed by Oracle Clusterware. |
| -delMasterHosts        | Deletes spare hosts to the pool of master hosts dynamically, when high availability is employed. On the command line, separate multiple host names by commas.  
The command fails if the indicated hosts are not spare hosts. On UNIX and Linux systems, only the `root` user can execute this command. |
| -delSubscriberHosts    | Deletes spare hosts to the pool of subscriber hosts dynamically, when high availability is employed. On the command line, separate multiple host names by commas.  
The command fails if the indicated hosts are not spare hosts. On UNIX and Linux systems, only the `root` user can execute this command. |
| -drop                  | Drops the active standby pair replication scheme and deletes its action scripts. |
| -dropVIPs              | Drops the virtual IP addresses for the active standby pair. |
| -endAlterSchema        | Issued this option after an operation using the -beginAlterSchema option. Rolls out the active standby pair after objects have been added to the schema, while recording the new replication checksum. The old standby is being destroyed and recreated through duplicate... |
### ttCWAdmin

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h</td>
<td>Prints a usage message and exits.</td>
</tr>
<tr>
<td>-help</td>
<td>Starts the TimesTen cluster agent.</td>
</tr>
<tr>
<td>-?</td>
<td>Used with <code>ttCWAdmin -shutdown</code> command. This command option tells the shutdown process to keep registered all TimesTen processes that are registered as Clusterware resources for the cluster agent and TimesTen daemon monitors for Clusterware.</td>
</tr>
<tr>
<td>-init</td>
<td>TimesTen cluster information is stored in the Oracle Cluster Registry (OCR). This option registers the admin user in the OCR. You must register the admin user once before performing any of the cluster initialization steps. On UNIX and Linux systems, login as the root user and run this command from any host in the system before creating any clusters. On Windows systems, login as the instance administrator to run this command. You do not need to perform this step when starting an existing cluster that you have shutdown.</td>
</tr>
<tr>
<td>-noderegister</td>
<td>This command reauthenticates the user names and passwords after any of them have been modified. Even if only a single password is changed, this command still prompts for all user names and passwords.</td>
</tr>
<tr>
<td>-ocrConfig</td>
<td>Relocates the database from the local host to the next available spare host specified in the MasterHosts attribute in the cluster.oracle.ini configuration file. If no spare host is available, an error is returned. If the database on the local host is active, roles are first reversed so that the remote standby store of the same cluster becomes active. The newly migrated database on the spare host always comes up as the standby database. This is useful to forcefully relocate a database if you must take the host offline, when high availability is employed. This command fails when basic High Availability (HA) is deployed for the same cluster.</td>
</tr>
<tr>
<td>-reauthenticate</td>
<td>For more details, see &quot;Changing user names or passwords when using Oracle Clusterware&quot; in the Oracle TimesTen In-Memory Database Replication Guide.</td>
</tr>
<tr>
<td>-relocate</td>
<td>For more details, see &quot;Changing user names or passwords when using Oracle Clusterware&quot; in the Oracle TimesTen In-Memory Database Replication Guide.</td>
</tr>
</tbody>
</table>
To create and start an active standby pair managed by Oracle Clusterware, using the clusterDSN DSN, enter:

```
ttCWAdmin -create -dsn clusterDSN
ttCWAdmin -start -dsn clusterDSN
```

To stop and drop an active standby pair managed by Oracle Clusterware, using the clusterDSN DSN, enter:

```
ttCWAdmin -stop -dsn clusterDSN
ttCWAdmin -drop -dsn clusterDSN
```
Notes

When you use Oracle Clusterware with TimesTen, you cannot use these commands and SQL statements:

- **CREATE ACTIVE STANDBY PAIR, ALTER ACTIVE STANDBY PAIR and DROP ACTIVE STANDBY PAIR SQL statements.**
- The `-cacheStart` and `-cacheStop` options of the `ttAdmin` utility after the active standby pair has been created.
- The `-duplicate` option of the `ttRepAdmin` utility.
- The `ttRepStart` and `ttRepStop` built-in procedures.
- The `-repStart` and `-repStop` options of the `ttAdmin` utility.

In addition, do not call `ttDaemonAdmin -stop` before calling `ttCWAdmin -shutdown`.

The TimesTen integration with Oracle Clusterware accomplishes these operations with the `ttCWAdmin` utility and the attributes in the `cluster.oracle.ini` file.
ttDaemonAdmin

Description
Starts and stops the TimesTen main daemon and Server.

Required privilege
This utility requires the instance administrator privilege.

Usage with TimesTen Scaleout
This utility is not supported in TimesTen Scaleout.

Syntax

```
ttDaemonAdmin {-h | -help | -?}
ttDaemonAdmin {-V | -version}
ttDaemonAdmin [-force] {-start | -stop | -restart}
ttDaemonAdmin [-startserver | -restartserver]
ttDaemonAdmin [-force] -stopserver
```

Options

```
ttDaemonAdmin has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h</td>
<td>Prints a usage message and exits.</td>
</tr>
<tr>
<td>-help</td>
<td></td>
</tr>
<tr>
<td>-?</td>
<td></td>
</tr>
<tr>
<td>-force</td>
<td>Starts or stops the TimesTen main daemon, even when warnings are returned or with -stopserver immediately stops the server processes.</td>
</tr>
<tr>
<td>-restart</td>
<td>Restarts the TimesTen main daemon.</td>
</tr>
<tr>
<td>-restartserver</td>
<td>Restarts the TimesTen Server.</td>
</tr>
<tr>
<td>-start</td>
<td>Starts the TimesTen main daemon.</td>
</tr>
<tr>
<td>-startserver</td>
<td>Starts the TimesTen Server daemon.</td>
</tr>
<tr>
<td>-stop</td>
<td>Stops the TimesTen main daemon.</td>
</tr>
<tr>
<td>-stopserver</td>
<td>Stops the TimesTen Server daemon.</td>
</tr>
<tr>
<td>-V</td>
<td>-version</td>
</tr>
</tbody>
</table>
```

Notes
Changes to the TimesTen Server options are temporary. To permanently set or disable the TimesTen Server options, you must change the options in the timesten.conf file.
Use the \texttt{-force} option with caution, as it may leave databases in a state where you must perform recovery procedures.

When you use this utility on Windows, you must be running with Windows Administrative privileges.

When you stop the daemon (\texttt{ttDaemonAdmin -stop}), first stop all application connections to the database. This includes stopping the replication agent and the cache agent, if they are running. This decreases startup time when the daemon is restarted. In addition, not stopping application connections or agents can result in the database becoming in validated.

If the Oracle Clusterware agent is running, you must stop it on the local host before stopping the TimesTen main daemon (\texttt{ttDaemonAdmin -stop}). If you do not stop the Clusterware agent, the main daemon stops temporarily with this command, but then restarts. To stop the Oracle Clusterware agent, use:

\texttt{ttCWAdmin -shutdown -hosts localhost}

When you use this utility to restart the server, the TimesTen daemon reads the \texttt{timesten.conf} files to see if it has been changed since it was last read. If the file has been changed, TimesTen checks for the values of the \texttt{timesten.conf} options:

\texttt{server_port server_shmipc server_shmsize noserverlog}

\textbf{See also}

For a description of all daemon options and instructions for changing the \texttt{timesten.conf} file, see Chapter 1, "TimesTen Instance Configuration File" in this reference and "Managing TimesTen daemon attributes" in the \textit{Oracle TimesTen In-Memory Database Operations Guide}. 
ttDaemonLog

Description

The TimesTen daemon (referred to as the TimesTen Data Manager Service on Windows) and its subdaemons and agents write error and status messages to the following daemon logs:

- A user error log that contains information you should be aware of, including actions you may need to take
- A support log containing everything in the user error log plus information of use by TimesTen Customer Support

The ttDaemonLog utility enables you to do the following:

- Control the types of events and categories of messages that are reported in the user error log.
- Display all messages or selected categories of messages from the log to the standard output.

Required privilege

This utility requires the instance administrator privilege.

Usage with TimesTen Scaleout

This utility is supported in TimesTen Scaleout.

Syntax

```
ttDaemonLog {-h | -help | -?}
ttDaemonLog {-V | -version}
```

Notes:

- The -file and -facility options apply only on UNIX and Linux.
- The -n option applies only on Windows and is not relevant in typical usage.

Options

ttDaemonLog has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-b</td>
<td>Prints all TimesTen-generated log entries.</td>
</tr>
</tbody>
</table>
When the end of the log is reached, `ttDaemonLog` does not terminate but continues to execute, periodically polling the log to retrieve and display additional TimesTen log records. This is useful, for example, for generating a display of log data that is updated in real time.

- `facility name` Specifies the `syslog` facility name being used.
  
  **Note:** This option applies only on UNIX and Linux.

- `file filename` Specifies the file into which TimesTen logs messages.
  
  If not specified, examine the system's `syslog` configuration to determine where TimesTen messages are being logged.
  
  **Note:** This option applies only on UNIX and Linux.

- `-h`, `-help`, `-?` Shows `ttDaemonLog` usage information and exits.

- `-maxlines` Maximum number of lines at end of the log to display. Defaults to 40 lines if `-f` is specified. If 0 is specified, there is no maximum.

- `-logcomponent component` By default, all categories of messages are logged, but you can use `-logcomponent` to specify a category to be logged, or `-nologcomponent` to specify a category to not be logged. You can specify only a single component, but can run `ttDaemonLog` with these options multiple times to determine the desired set of messages.

  If a DSN or connection string is specified, the option applies only to the specified database. You can run `ttDaemonLog` multiple times to set these options for multiple databases.

  Supported categories are:

  - `ALL` (default): For all messages.
  - `CACHE`: For messages from the cache agent, designated by `CACHE`
  - `DAEMON`: For messages from the main daemon and subdaemons
  - `DAEMONDBG`: For additional information from the main daemon and subdaemons
  - `GADMIN`: For messages from grid administrator
  - `GADMINCS`: For messages from grid administrator client server
  - `GCWRKR`: For messages from grid client worker
  - `GRID`: For messages from a TimesTen Grid
  - `GRWKR`: For messages from the grid worker
  - `REPLICATION`: For messages from the replication agent, designated by `REP`
  - `TTSTATS`: For messages from `ttStats`

- `-loglevel level` Specifies a cutoff for the level of messages that are logged in the support log. A lower value results in fewer messages. (For example, if you specify level 5, messages of level 1, 2, 3, 4, or 5 would be logged.) This option is typically relevant only for Customer Support use.

  If a DSN or connection string is specified, the option applies only to that database.

- `-logreset` Resets event logging parameters.

- `-msg messagestring` Inserts the specified text into the TimesTen user log.
Examples

By default, the ttDaemonLog utility logs messages and errors from all the TimesTen components. You can narrow the scope of what is written to the log by setting the -nologcomponent option. This option can be applied to selected databases or all databases.

To display all the output from the TimesTen daemon and server on your local computer:

```
  ttDaemonLog
```

To prevent messages and errors related to replication for all databases from being written to the log:

```
  ttDaemonLog -nologcomponent replication
```

To prevent messages and errors related to replication for the masterdsn database from being written to the log:

```
  ttDaemonLog -nologcomponent replication masterdsn
```

To prevent both replication and TimesTen Cache errors and messages from being written:

```
  ttDaemonLog -nologcomponent replication
  ttDaemonLog -nologcomponent cache
```

If, after disabling a component through the -nologcomponent option, you want to re-enable it, you can use the -logcomponent option. For example, after disabling
messages for replication and TimesTen Cache as shown in the preceding example, you can re-enable replication messages as follows:

ttDaemonLog -logcomponent replication

To re-enable logging for all TimesTen components, use the -logreset option:

ttDaemonLog -logreset

The TimesTen Server generates a message each time an application connects to or disconnects from a client DSN if these messages were specified to be generated during installation. To display just the server log messages:

ttDaemonLog -show server

To display just the replication agent messages:

ttDaemonLog -show replication

To display just the cache agent messages:

ttDaemonLog -show cache

To display all messages from the TimesTen processes:

ttDaemonLog -show all

To restore logging to its default "verbose" level, use the -setverbose option:

ttDaemonLog -setverbose

On UNIX and Linux systems, to write the log output to the file /var/adm/syslog/syslog.log:

ttDaemonLog -file /var/adm/syslog/syslog.log

On UNIX and Linux systems, to direct logging to the local7 facility:

ttDaemonLog -facility local7

Notes

While primarily intended for use by TimesTen Customer Support, this information may be useful to system administrators and developers.

This utility is supported only where the TimesTen Data Manager is installed.

To permanently set or disable verbose logging, change the options in the timesten.conf file. See "Modifying informational messages" in the Oracle TimesTen In-Memory Database Operations Guide.
ttDestroy

Description

Destroys a database including all checkpoint files, transaction logs and daemon catalog entries (though not the DSNs).

Required privilege

This utility requires the instance administrator privilege.

Usage with TimesTen Scaleout

This utility is not supported in TimesTen Scaleout.

Syntax

```
ttDestroy {-h | -help | -?}
ttDestroy {-V | -version}
ttDestroy [-wait] [-timeout secs] [-force] [-connStr connection_string | DSN | dspath]
```

Options

ttDestroy has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-connStr connection_string</td>
<td>An ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.</td>
</tr>
<tr>
<td>DSN</td>
<td>Specifies an ODBC data source name of the database to be destroyed.</td>
</tr>
<tr>
<td>dspath</td>
<td>The fully qualified name of the database to be destroyed.</td>
</tr>
<tr>
<td></td>
<td>This is not the DSN associated with the connection but the fully qualified database path name associated with the database as specified in the {DataStores} parameter of the database’s ODBC definition.</td>
</tr>
<tr>
<td></td>
<td>For example, for a database consisting of files/home/payroll/2011.ds0, /home/payroll/2011.ds1, and several transaction log files /home/payroll/2011.logn, dspath is /home/payroll/2011.</td>
</tr>
<tr>
<td>-h</td>
<td>Prints a usage message and exits.</td>
</tr>
<tr>
<td>-help</td>
<td></td>
</tr>
<tr>
<td>-?</td>
<td></td>
</tr>
<tr>
<td>-force</td>
<td>Destroy even if files are from an incompatible version or a different instance of TimesTen.</td>
</tr>
<tr>
<td>-timeout seconds</td>
<td>Indicates the time in seconds that ttDestroy should wait. If no timeout value is supplied, TimesTen waits five seconds before retrying the destroy operation.</td>
</tr>
<tr>
<td>-V</td>
<td>-version</td>
</tr>
<tr>
<td>-wait</td>
<td>Causes ttDestroy to continually retry the destroy operation until it is successful, in those situations where the destroy fails due to some temporary condition, such as when the database is in use.</td>
</tr>
</tbody>
</table>
Examples

```
ttDestroy /users/pat/TimesTen/Daily/F112697
```

Notes

Using `ttDestroy` is the only way to delete a database completely and safely. Do not remove database checkpoint or transaction log files manually.

This utility is supported only where the TimesTen Data Manager is installed.

`ttDestroy` does not perform cleanup of Oracle database objects from autorefresh or AWT cache groups. If there are autorefresh or AWT cache groups in the database, execute the `cachecleanup.sql` script to clean up the cache objects in the Oracle database for that particular database, to generate Oracle SQL to perform cleanup after the database has been destroyed.
**ttInstallationCheck**

**Description**

The `ttInstallationCheck` utility examines all files in an installation of TimesTen and will generate a signature for the installation. The signatures from two installations can be compared; if there are any differences in the installations the signatures differ.

If any of the following have occurred, the signature reported is different:

- Contents of any file have changed
- Name of any file has changed
- New files are present in the installation
- Files have been removed from the installation
- Files have incorrect permissions

**Required privilege**

This utility requires the instance administrator privilege.

**Usage with TimesTen Scaleout**

This utility is supported in TimesTen Scaleout.

**Syntax**

```
 ttInstallationCheck [-h | -help | -?]
 ttInstallationCheck [-v | -verbose | -?]
 ttInstallationCheck [-install_dir path] [-generate]
```

**Options**

`ttInstallationCheck` has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h</td>
<td>Displays help information.</td>
</tr>
<tr>
<td>-help</td>
<td></td>
</tr>
<tr>
<td>-?</td>
<td></td>
</tr>
<tr>
<td>-generate</td>
<td>Generate and print the checksum for the installation, but do not verify.</td>
</tr>
<tr>
<td>-install_dir path</td>
<td>Specifies the directory in which TimesTen is installed.</td>
</tr>
<tr>
<td>-v</td>
<td>-verbose</td>
</tr>
</tbody>
</table>
ttInstallDSN

Description
The ttInstallDSN utility, for TimesTen Scaleout, generates a Windows client DSN for each of one or more entries in the provided input file and installs them into the ODBC control panel as system DSNs. Use the ttGridAdmin gridClientExport command to generate the input file.

Required privilege
This utility requires the instance administrator privilege.

Usage with TimesTen Scaleout
This utility is for use with TimesTen Scaleout.

Syntax

```
ttInstallDSN [-h | -help | -?]
ttInstallDSN [-f file] [Client_DSN_Name | -a | -l] [-force]
```

Options

ttInstallDSN has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h</td>
<td>Displays help information.</td>
</tr>
<tr>
<td>-help</td>
<td></td>
</tr>
<tr>
<td>-?</td>
<td></td>
</tr>
<tr>
<td>-f file</td>
<td>Specifies the full path and name of a file (generated by ttGridAdmin gridClientExport) containing one or more DSN definitions. Also typically provide one of the following:</td>
</tr>
<tr>
<td>[Client_DSN_Name</td>
<td>-a</td>
</tr>
<tr>
<td></td>
<td>■ The -a option to install all DSNs from the input file.</td>
</tr>
<tr>
<td></td>
<td>■ The -l option to simply list all the DSNs in the input file. None is installed.</td>
</tr>
<tr>
<td></td>
<td>If you do not specify -f, the default file is sys.odbc.ini in the current directory.</td>
</tr>
<tr>
<td></td>
<td>If you do not specify one of the three items listed above, the default behavior is to list the DSNs in the file and ask which you want to install.</td>
</tr>
<tr>
<td>-force</td>
<td>If there are already DSNs in the Windows registry, this allows them to be overridden by the specified DSNs. Without this option, if there are already DSNs in the Windows registry, the utility issues a warning and cannot install new ones.</td>
</tr>
</tbody>
</table>

Examples

In this example, there are already DSNs in the Windows registry. The user first tries without -force and is issued a warning, so then uses -force.

```
C:\mydir> ttinstalldsn.bat -f c:\temp\sys.odbc.ini
```
---
.ini File: c:\temp\sys.odbc.ini
---
Found the following DSNs in available 'c:\temp\sys.odbc.ini'.
 0 : database1CS
[ Please select the DSN to be imported: ]
0
Warning: The following DSNs already existed and were not added:
        database1CS

C:\mydir> ttinstalldsn.bat -f c:\temp\sys.odbc.ini -force
---
.ini File: c:\temp\sys.odbc.ini
---
Found the following DSNs in available 'c:\temp\sys.odbc.ini'.
 0 : database1CS
[ Please select the DSN to be imported: ]
0
Modifying DSN 'database1CS'.
ttInstanceCreate

Description

The ttInstanceCreate utility creates a new TimesTen instance. You can specify options in one of these ways:

- On the command line.
- In a file.
- Interactively as the utility runs.

If you do not specify options on the command line, or if the only options used are `-record` and/or `-verbose`, ttInstanceCreate runs in an interactive mode, prompting the Instance Administrator for information.

If you specify the `-batch` option on the command line, ttInstanceCreate runs in interactive mode, and attempts to answer any questions by fetching the answers from a recorded batch file, generated by a previous run that specified the `-record` option. If the answer to a question is not present in the batch file, the utility prompts the Instance Administrator to answer the question interactively.

If you specify other options on the command line, they are used as the source of information. The ttInstanceCreate utility does not prompt the user for unknown values.

Usage with TimesTen Scaleout

This utility is supported in TimesTen Scaleout but is used only to create the first management instance. (Create additional instances using ttGridAdmin instanceCreate.)

Required privilege

This utility requires the instance administrator privilege.

Syntax

```
ttInstanceCreate {-h | -help | -?} [-verbose]


ttInstanceCreate [-batch [filename]]
```

Options

ttInstanceCreate has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h</code></td>
<td>Displays help information.</td>
</tr>
<tr>
<td><code>-help</code></td>
<td></td>
</tr>
<tr>
<td><code>-?</code></td>
<td></td>
</tr>
</tbody>
</table>
### ttInstanceCreate

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-batch [filename]</td>
<td>Specifies the file to be used to provide input to the <code>ttInstanceCreate</code> utility. If not specified, no input file is used.</td>
</tr>
<tr>
<td>-record filename</td>
<td>Records responses to installation questions into the file specified by <code>filename</code>. The file then can be specified as the parameter to the <code>-batch</code> option.</td>
</tr>
<tr>
<td>-name name</td>
<td>Specifies the name of the instance to be created.</td>
</tr>
<tr>
<td>-location path</td>
<td>Specifies the path of the directory in which the instance is to be created. This directory must already exist. A new directory is created in the existing directory. The name of the new directory is specified in the <code>-name</code> option. This new directory is the new Instance Home.</td>
</tr>
<tr>
<td>-daemonport daemon_port</td>
<td>The port number on which the TimesTen daemon (<code>timestend</code>) process for this instance listens. This port must not already be in use by any other application or instance on the system.</td>
</tr>
<tr>
<td>-tnsadmin location</td>
<td>If using Cache, this option configures the location to be used for the <code>TNS_ADMIN</code> setting.</td>
</tr>
<tr>
<td>-csport port</td>
<td>The value to be used for the TimesTen client/server port number for this instance. If not specified, the default is <code>daemonport + 1</code>.</td>
</tr>
<tr>
<td>-grid</td>
<td>Indicates that the instance should be configured for use with TimesTen Scaleout.</td>
</tr>
<tr>
<td>-start</td>
<td>Specifies that the instance should be started after it is created.</td>
</tr>
<tr>
<td>-clientonly</td>
<td>Specifies that the instance is client only. Most other arguments are not supported for a client only instance.</td>
</tr>
<tr>
<td>-serverhost host</td>
<td>For TimesTen Client, specify the host for the TimesTen Server. The value is the logical name, domain name, or IP address of the computer where the TimesTen Server is running. For a full instance, this defaults to the host where you are creating the instance.</td>
</tr>
<tr>
<td>-force</td>
<td>Specifies that an instance directory specified with the <code>-instance</code> option is to be overwritten if it already exists. The directory is overwritten only if:</td>
</tr>
<tr>
<td></td>
<td>1. The specified instance directory is empty, or</td>
</tr>
<tr>
<td></td>
<td>2. The specified instance directory contains a <code>conf/timesten.conf</code> file.</td>
</tr>
<tr>
<td>-verbose</td>
<td>Displays additional information during the operation of the utility.</td>
</tr>
</tbody>
</table>
ttInstanceDestroy

Description
Use the ttInstanceDestroy utility to destroy an existing instance.

The instance to be destroyed is chosen based on the current setting of the TIMES Ten_HOME environment variable.

Required privilege
This utility requires the instance administrator privilege.

Usage with TimesTen Scaleout
This utility is supported in TimesTen Scaleout, but in most circumstances use ttGridAdmin instanceDelete.

Syntax

```
ttInstanceDestroy {-h | -help | -?} [-verbose]
ttInstanceDestroy [-force]
```

Options

```
ttInstanceDestroy has the options:
```

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h</td>
<td>Displays help information.</td>
</tr>
<tr>
<td>-help</td>
<td></td>
</tr>
<tr>
<td>-?</td>
<td></td>
</tr>
<tr>
<td>-force</td>
<td>If specified, you are not asked to confirm operations.</td>
</tr>
<tr>
<td></td>
<td>If -force is not specified:</td>
</tr>
<tr>
<td></td>
<td>■ You are reminded that if you installed the startup scripts for</td>
</tr>
<tr>
<td></td>
<td>this instance as root, you must uninstall them as root with</td>
</tr>
<tr>
<td></td>
<td>the setuproot -uninstall command.</td>
</tr>
<tr>
<td></td>
<td>■ You are asked for confirmation to destroy the instance.</td>
</tr>
<tr>
<td></td>
<td>■ You are asked for confirmation to remove the info and conf</td>
</tr>
<tr>
<td></td>
<td>directories.</td>
</tr>
<tr>
<td>-verbose</td>
<td>Displays additional TimesTen installation information.</td>
</tr>
</tbody>
</table>
**ttInstanceModify**

**Description**

Use the `ttInstanceModify` utility to modify certain attributes of an instance, including:

- The installation associated with this instance.
- The daemon and server port numbers.
- The `TNS_ADMIN` for the instance.
- The configuration of TimesTen Replication with Oracle Clusterware for this instance.

The instance that is modified is the one that `$TIMESTEN_HOME` references.

If you do not specify any options for this utility, `ttInstanceModify` displays the current value of each attribute and a prompt that allows you to keep the value or change it.

If you change any of the settings, the utility:

- Shuts down the TimesTen daemon for the instance.
- Edits the `timesten.conf` file in the `timesten_home/conf` directory.
- Starts the TimesTen main daemon for the instance.

**Required privilege**

This utility requires the instance administrator privilege.

**Usage with TimesTen Scaleout**

This utility is supported in TimesTen Scaleout, but in most circumstances use `ttGridAdmin instanceModify`.

**Syntax**

```
ttInstanceModify [-h | -help | -?] [-verbose]
ttInstanceModify [-port daemonport] [-serverport cs_port] [-tnsadmin location] [-crs] [install installation_dir]
```

**Options**

`ttInstanceModify` has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h</td>
<td>Displays help information.</td>
</tr>
<tr>
<td>-help</td>
<td></td>
</tr>
<tr>
<td>-?</td>
<td></td>
</tr>
<tr>
<td>-daemonport daemonport</td>
<td>Updates the TimesTen main daemon port number.</td>
</tr>
<tr>
<td>-crs</td>
<td>Creates or modifies the instance's Oracle Clusterware configuration.</td>
</tr>
<tr>
<td>-install installation_dir</td>
<td>Changes the installation that the instance uses.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td><code>tnsadmin location</code></td>
<td>Updates the instance’s TNS_ADMIN setting (for cache).</td>
</tr>
</tbody>
</table>
ttIsql

Description
You can execute SQL statements and call TimesTen built-in procedures from ttIsql. You can execute SQL interactively from the command line. For a detailed description on running SQL from ttIsql, use the -helpfull option. In addition, you can call a TimesTen built-in procedure with call procedure-name.

The ttIsql command attempts to cancel an ongoing ODBC function when the user presses Ctrl-C.

On UNIX and Linux systems, this utility is supported for TimesTen Data Manager DSNs. Use ttIsqlCS for client/server DSNs.

The ttIsql utility starts with AUTOCOMMIT turned on, even when running a script. You can turn AUTOCOMMIT off and back on as necessary.

For more details on the ttIsql utility, see the chapter "Using the ttIsql Utility" in the Oracle TimesTen In-Memory Database Operations Guide.

Required privilege
This utility requires no privileges.

Usage with TimesTen Scaleout
This utility is supported in TimesTen Scaleout.

Syntax

```
ttIsql {-h | -help | -? | -helpcmds | - helpfull}
ttIsql {-V | -version}
ttIsql [-f inputFile] [-v verbosity] [-e commands | sql_statement]
[-interactive] [-N ncharEncoding] [-wait] {-connStr connection_string | DSN}
```

Options

ttIsql has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-connStr connection_string</td>
<td>An ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.</td>
</tr>
<tr>
<td>DSN</td>
<td>Specifies an ODBC data source name of the database to be connected.</td>
</tr>
<tr>
<td>-e commands</td>
<td>Specifies a semicolon separated list of ttIsql commands to execute on startup.</td>
</tr>
<tr>
<td>-f filename</td>
<td>Read SQL statements from filename.</td>
</tr>
<tr>
<td>-h</td>
<td>Prints a usage message and exits.</td>
</tr>
<tr>
<td>-help</td>
<td>Prints a short list of the interactive commands.</td>
</tr>
<tr>
<td>-?</td>
<td>Prints a full description of the interactive commands.</td>
</tr>
</tbody>
</table>
## Option Description

- **-interactive**
  Forces interactive mode. This is useful when running from an `emacs comint` buffer.

- **-N ncharEncoding**
  Specifies the character encoding method for NCHAR output.
  Valid values are `LOCALE` or `ASCII`. `LOCALE` (the default) sets the output format to the locale-based setting.
  If no value is specified, TimesTen uses the system’s native language characters.

- **-V | -version**
  Prints the release number of `ttIsql` and exits.

- **-v verbosity**
  Specifies the verbosity level. One of:
  0 - Shows error information only. If all commands succeed, there is no output.
  1 - The basic output generated by commands is displayed.
  2 (default) - Same as level 1, plus it shows more detailed results of commands. At this level simplified SQL error and information messages are displayed. In addition, `ttIsql` commands that are read from an external file are echoed to the display.
  3 - Same as level 2, with more detailed error and information messages.
  4 - Same as level 3, plus complete error and information messages are displayed. Also displayed are messages about prepared commands, "success" messages for each command that succeeded and content of XLA records.

- **-wait**
  Waits until successful connect.
Also see the list of `ttIsql "Set/show attributes"` on page 5-82.

Boolean commands can accept the values "ON" and "OFF" or "1" and "0".

`ttIsql` has the commands:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`accept variable[NUMBER</td>
<td>CHAR</td>
</tr>
<tr>
<td><code>allfunctions [owner_name_ pattern. ] table_name_pattern</code></td>
<td>Lists, in a single column, the names of all the PL/SQL functions that match the given pattern selected from <code>SYS.ALL_OBJECTS</code>. When a pattern is missing, the pattern defaults to &quot;%&quot;. If passthrough is enabled, lists PL/SQL functions matching the pattern in the Oracle database. See the <code>functions</code> command.</td>
</tr>
<tr>
<td><code>allindexes [owner_name_ pattern. ] table_name_pattern</code></td>
<td>Describes the indexes that it finds on the tables that match the input pattern selected from <code>SYS.ALL_OBJECTS</code>. When a pattern is missing, the patterns default to &quot;%&quot;. If passthrough is enabled, lists indexes on tables matching the pattern in the Oracle database. See the <code>indexes</code> command.</td>
</tr>
<tr>
<td><code>allpackages [owner_name_ pattern. ] table_name_pattern</code></td>
<td>Lists, in a single column, the names of all the PL/SQL packages that match the given pattern selected from <code>SYS.ALL_OBJECTS</code>. When a pattern is missing, the patterns default to &quot;%&quot;. If passthrough is enabled, lists PL/SQL packages matching the pattern in the Oracle database. See the <code>packages</code> command.</td>
</tr>
<tr>
<td><code>allprocedures [owner_name_ pattern. ] procedure_name_pattern</code></td>
<td>Lists, in a single column, the names of all the PL/SQL procedures that match the given pattern selected from <code>SYS.ALL_OBJECTS</code>. When a pattern is missing, the pattern defaults to &quot;%&quot;. If passthrough is enabled, lists PL/SQL procedures matching the pattern in the Oracle database. See the <code>procedures</code> command.</td>
</tr>
<tr>
<td><code>allsequences [owner_name_ pattern. ] table_name_pattern</code></td>
<td>Lists, in a single column, the names of all the sequences that match the given pattern selected from <code>SYS.ALL_OBJECTS</code>. When a pattern is missing, the pattern defaults to &quot;%&quot;. If passthrough is enabled, lists sequences on tables matching the pattern in the Oracle database. See the <code>sequences</code> command.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>allsynonyms</td>
<td>Lists, in a single column, the names of all synonyms that match the given pattern. When a pattern is missing, the pattern defaults to &quot;%&quot;. If passthrough is enabled, lists synonyms on tables matching the pattern in the Oracle database.</td>
</tr>
<tr>
<td></td>
<td>See the <code>synonyms</code> command.</td>
</tr>
<tr>
<td>alltables</td>
<td>Lists, in a single column, the names of all the tables that match the given pattern selected from SYS.ALL_OBJECTS. When a pattern is missing, the pattern defaults to &quot;%&quot;. If passthrough is enabled, lists tables matching the pattern in the Oracle database.</td>
</tr>
<tr>
<td></td>
<td>See the <code>tables</code> command.</td>
</tr>
<tr>
<td>allviews</td>
<td>Lists, in a single column, the names of all the views that match the specified pattern selected from SYS.ALL_OBJECTS. When a pattern is missing, the pattern defaults to &quot;%&quot;. If passthrough is enabled, lists views matching the pattern in the Oracle database.</td>
</tr>
<tr>
<td></td>
<td>See the <code>waitfor</code> command.</td>
</tr>
<tr>
<td>builtins</td>
<td>Lists, in a single column, the names of all the TimesTen built-in procedures that match the given pattern. When the pattern is missing, the pattern defaults to &quot;%&quot;.</td>
</tr>
<tr>
<td></td>
<td>See the <code>procedures</code> command.</td>
</tr>
<tr>
<td>bye</td>
<td>Exits ttIsql.</td>
</tr>
<tr>
<td>exit</td>
<td></td>
</tr>
<tr>
<td>cachegroups</td>
<td>Reports information on cache groups defined in the currently connected data source, including the state of any terminated databases that contain autorefresh cache groups. If the optional argument is not specified then information on all cache groups in the current data source is reported.</td>
</tr>
<tr>
<td>cachesqlget</td>
<td>Generates an Oracle SQL*Plus compatible script for the installation or uninstallation of Oracle database objects associated with a readonly cache group, a user managed cache group with incremental autorefresh or an AWT cache group.</td>
</tr>
<tr>
<td></td>
<td>If <code>INSTALL</code> is specified, the Oracle SQL statement to install the Oracle database objects is generated. If <code>UNINSTALL</code> is specified, the Oracle SQL statement used to remove the Oracle objects is generated.</td>
</tr>
<tr>
<td></td>
<td>If <code>UNINSTALL</code> is specified, a SQL statement to remove all Oracle database objects in the autorefresh user's account is generated. If the optional <code>filename</code> argument is included, the generated SQL statement is saved to the specified external file. If the external file exists, its contents are destroyed before writing to the file.</td>
</tr>
</tbody>
</table>
### Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cd directory</td>
<td>Changes the current directory. This is the equivalent of the <code>cd</code> command in interactive shells. After changing to the directory <code>directory</code>, the define alias <code>_CWD</code> is be set to this directory. Subsequent commands that rely on relative paths will use this directory as the starting point. Examples of affected commands are <code>spool</code>, <code>run</code>, <code>savehistory</code>, <code>host</code>, and <code>edit</code>.</td>
</tr>
<tr>
<td>clearhistory</td>
<td>Clears the history buffer. Also see <code>history</code> and <code>savehistory</code>.</td>
</tr>
<tr>
<td>clienttimeout</td>
<td>Sets the client timeout value in seconds for the current connection. If no value is specified, displays the current value. See &quot;Choose SQL and PL/SQL timeout values&quot; in Oracle TimesTen In-Memory Database Operations Guide for information about the relationship between the client timeout, SQL timeout, and PL/SQL timeout.</td>
</tr>
<tr>
<td>close [connect_id.] command_id</td>
<td>Closes the prepared command identified by connection name <code>connect_id</code> and command ID <code>command_id</code>. If <code>command_id</code> is not specified, closes the most recent command. If <code>closeall</code> is selected, closes all currently open prepared commands. Use <code>prepare</code> to create the prepared command.</td>
</tr>
<tr>
<td>closeall</td>
<td></td>
</tr>
<tr>
<td>cmdcache</td>
<td>Displays the contents of the TimesTen SQL command cache. Specify the <code>sqlcmdid</code>, <code>querytext</code> or <code>owner</code> column and query substring to search for a specific portion of a SQL query. If no column is specified, searches the <code>querytext</code> column. If passthrough is enabled, the command ID is not passed through to the Oracle database.</td>
</tr>
<tr>
<td>commit</td>
<td>Commits the current transaction (durably if <code>Durability=1</code> for the connection).</td>
</tr>
<tr>
<td>commitdurable</td>
<td>Commits the current transaction durably.</td>
</tr>
<tr>
<td>compact</td>
<td>Compacts the database.</td>
</tr>
<tr>
<td>compare varA VarB</td>
<td>Compares the values of two variables and reports if they are different. The first difference is reported.</td>
</tr>
</tbody>
</table>
### Command Description

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connect[[DSN][as]connid [adding]</td>
<td>Connects to the database with the specified ODBC connection_string.</td>
</tr>
<tr>
<td>[connection_string]</td>
<td>If no password is supplied in this format, ttIsql prompts for the password.</td>
</tr>
<tr>
<td>If no user is given, ttIsql attempts to connect using the user name of the current user as indicated by the operating system.</td>
<td></td>
</tr>
<tr>
<td>If as connid is specified, you can explicitly name the connection. The connid must be only alphanumeric characters, is case sensitive, must start with an alpha character and can only be a maximum of 30 characters in length. The name of connid is automatically supplied to the ConnectionName general connection attribute. If the connect fails, the current connection is set to a special reserved connection named &quot;none,&quot; which is never connected to anything.</td>
<td></td>
</tr>
<tr>
<td>When adding is specified, it refers to creating a new connection to the DSN specified by DSN or by the connection string.</td>
<td></td>
</tr>
<tr>
<td>createandloadfromoraquery</td>
<td>Takes a table name, the number of threads for parallel load and an Oracle SELECT statement.</td>
</tr>
<tr>
<td>[owner_name.]table_name[num_threads] query</td>
<td>Creates the table in TimesTen if the table does not exist. Then, loads the table with the query result from the Oracle database. If the command creates the table, the table column names and types are derived from the query result.</td>
</tr>
<tr>
<td>Notes:</td>
<td></td>
</tr>
<tr>
<td>■ The specified TimesTen table cannot be a system table, a synonym, a view, a materialized view or a detail table of a materialized view, a global temporary table or a cache group table.</td>
<td></td>
</tr>
<tr>
<td>■ The query cannot have any parameter bindings.</td>
<td></td>
</tr>
<tr>
<td>■ Any unsupported column types result in a warning being logged. The output issues a comment for the unsupported column data type.</td>
<td></td>
</tr>
<tr>
<td>■ If you do not supply a value for num_threads, defaults to four threads.</td>
<td></td>
</tr>
<tr>
<td>■ For details and usage information, see &quot;Loading data from an Oracle database into a TimesTen table&quot; in the Oracle TimesTen In-Memory Database Operations Guide.</td>
<td></td>
</tr>
<tr>
<td>■ You must rollback or commit after executing this operation.</td>
<td></td>
</tr>
<tr>
<td>■ Also see the NOTES section in the description of the built-in procedure ttLoadFromOracle.</td>
<td></td>
</tr>
<tr>
<td>Required Privileges:</td>
<td></td>
</tr>
<tr>
<td>Requires INSERT privilege on the table specified. Also, requires the CREATE TABLE privilege if the table does not exist. The Oracle session user must have all required privileges to execute the query on the Oracle database.</td>
<td></td>
</tr>
<tr>
<td>define name [= value]</td>
<td>Defines a string substitution alias.</td>
</tr>
<tr>
<td>If no value is provided, ttIsql displays the current definition for the specified name.</td>
<td></td>
</tr>
<tr>
<td>You must set define on to enable command substitution. See &quot;Set/show attributes&quot; on page 5-82.</td>
<td></td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>describe [owner_pattern.] name_pattern</td>
<td>List information on tables, synonyms, views, materialized views, materialized view logs, sequences, cache groups, PL/SQL functions, PL/SQL procedures, PL/SQL packages and TimesTen built-in procedures in that order when the argument is [owner_pattern.]name_pattern. Otherwise lists the specific objects that match the given pattern. Describes the parameters and results columns when the argument is sql_statement. If passthrough is set to 3, lists information about the same types of objects in the Oracle database. If * is specified, reports the prepared statements for all connections. If the table or materialized view being described is in a TimesTen Scaleout database, this command reports the distribution scheme. If the table being described is a materialized view log, the message lists the name of the materialized view for which the table is a log. If the table being described has a materialized view log on it, the message indicates the name of the materialized view log. When describing cache groups, reports information on cache groups defined in the currently connected data source, including the state of any terminated databases that contain autorefresh cache groups. If the command is describing a sequence in a TimesTen Scaleout database, displays the batch field. The command alias is desc. Use free to release the prepared command.</td>
</tr>
<tr>
<td>disconnect [all]</td>
<td>Disconnects from the database. If all is specified, disconnects and closes all connections. When disconnect finishes, the current connection is set to the reserved connection named &quot;none.&quot;</td>
</tr>
<tr>
<td>dssize [k</td>
<td>m</td>
</tr>
<tr>
<td>e: msg</td>
<td>Echoes the specified messages, terminated by the end of the line. A semicolon is not required to end the line. Messages are not echoed if verbosity is set to 0.</td>
</tr>
<tr>
<td>PROMPT msg</td>
<td></td>
</tr>
</tbody>
</table>
### Utilities 5-71

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| **edit** [file | history_search_command ] | You can use the `ttIsql edit` command to edit a file or edit `ttIsql` commands in a text editor. The `ttIsql edit` command starts a text editor such as `emacs`, `gedit`, or `vi`.

If TimesTen does not find an exact file match for the specified `file` parameter, it searches for `file.sql`. If neither file exists, `ttIsql` starts the editor with the file `file`.

You can edit a SQL statement that is stored in the history list of the current `ttIsql` session. When calling the `ttIsql edit` command specify the `!` character followed by the number of the command or a search string.

If you execute the `ttIsql edit` command with a `history_search_command` parameter, `ttIsql` executes the contents of the file after you exit the text editor. The contents of the file are executed as a single `ttIsql` command. If you do not want to execute the contents of the file, delete the contents of the file and save the file before you exit the editor.

You can only use one parameter at a time. The `history_search_command` parameter is defined as the `!` character followed by the number of the command or a search string. If you do not specify a `!` character, the `ttIsql edit` command interprets the parameter as `file`. If you do not specify a parameter or specify `!!`, the last `ttIsql` command is edited.

You can specify the default editor by defining the `ttIsql _EDITOR define alias`. The following example sets the default editor to `vi`:

```
Command> DEFINE _EDITOR=vi
```

If you do not define the `_EDITOR` define alias, `ttIsql` uses the editor specified by the `VISUAL` environment variable. If the `_EDITOR` define alias and the `VISUAL` environment variables are not set, `ttIsql` uses the editor specified by the `EDITOR` environment variable. When `_EDITOR`, `VISUAL`, and `EDITOR` are not set, `vi` is used for UNIX and Linux systems and `notepad.exe` is used for Windows.

For more details, see "Using the `ttIsql edit` command" in the Oracle TimesTen In-Memory Database Operations Guide.

| exec [connect_id.] command_id | Executes the prepared command `command_id` on connection `connect_id` or executes a PL/SQL statement.
| PLSQLSTMT | The `connect_id` optionally names a `ttIsql` connection and `command_id` is an integer from 1 to 255.

If `PLSQLSTMT` is supplied, `ttIsql` prepends the statement with `BEGIN` and appends the statement with `END`, thus allowing the PL/SQL statement to execute.

If no argument is supplied, executes the most recent command.

Use `free` to release the prepared command.

| execandfetch [connect_id.]command_id | Executes and fetches all results from prepared command `command_id` on connection `connect_id`. If `command_id` is not specified, executes and fetches all results from the most recent command.

Use `free` to release the prepared command. |
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>explain [plan for] Connid. ttisqlcmdid</td>
<td>Explains the plan for the specified SQL statement, including prepared ttisql statements, specified in the ttisqlcmdid argument, or the sqlcmdid argument. A digit that is not qualified with the sqlcmdid argument, is interpreted as a ttisql prepared statement ID. If passthrough is enabled, the command ID is not passed through to the Oracle database.</td>
</tr>
<tr>
<td>fetchall [connect_id.]command_id</td>
<td>Fetches all results from prepared command command_id on connection connect_id. If command_id is not specified, fetches all results from the most recent command. The command must already have been executed using exec. Use free to release the prepared command.</td>
</tr>
<tr>
<td>fetchnext num_rows [connect_id.]command_id</td>
<td>Fetches up to num_rows rows from prepared command command_id on connection connect_id. If command_id is not specified, fetches num_rows rows from the most recent command. The command must already have been executed using exec. Use free to release the prepared command.</td>
</tr>
<tr>
<td>fetchone [connect_id.]command_id</td>
<td>Fetches one result from prepared command command_id on connection connect_id. If command_id is not specified, fetches one result from the most recent command. The command must already have been executed using exec. Use free to release the prepared command.</td>
</tr>
<tr>
<td>free [[connect_name.]connect_id.]command_id</td>
<td>Frees prepared command command_id on connection connect_id. If no command is specified, frees the most recent command. Use prepare to create the prepared command.</td>
</tr>
<tr>
<td>functions [object_name_pattern]</td>
<td>Lists, in a single column, the names of PL/SQL functions owned by the current user that match the given pattern. When a name pattern is missing, the pattern defaults to %. If passthrough is enabled, lists PL/SQL functions matching the pattern in the Oracle database. See the allfunctions command.</td>
</tr>
<tr>
<td>grid stmt</td>
<td>Performs that specified statement on a grid database.</td>
</tr>
<tr>
<td>grid monitor [optional_monitor_column]</td>
<td>Formats the contents of the SYS.GV$MONITOR table for easy viewing. If the optional_monitor_column is specified, only that column is displayed.</td>
</tr>
</tbody>
</table>
### Command Description

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>help [command [command ...]</td>
<td>Prints brief or detailed help information for commands.</td>
</tr>
<tr>
<td>all</td>
<td>If specific commands are given as arguments then detailed help for each command is printed.</td>
</tr>
<tr>
<td>comments</td>
<td>If you do not know the exact name of a command, try typing just a few characters that may be part of the command name. ttiSql searches and displays help for any commands that include the characters.</td>
</tr>
<tr>
<td>attributes</td>
<td>If all is given as an argument then detailed help for all commands is printed.</td>
</tr>
<tr>
<td></td>
<td>If comments is given as an argument then information on using ttiSql comments within scripts is printed.</td>
</tr>
<tr>
<td></td>
<td>If attributes is given as an argument then information on the set/show attributes is printed.</td>
</tr>
<tr>
<td></td>
<td>If no argument is given then brief help information for all commands is printed.</td>
</tr>
<tr>
<td>history [-all] [-h] [-r] [num_commands]</td>
<td>ttiSql implements a csh-like command history.</td>
</tr>
<tr>
<td></td>
<td>Lists previously executed commands. The num_commands parameter specifies the number of commands to list. If the num_commands parameter is omitted then the previous 10 commands are listed by default.</td>
</tr>
<tr>
<td></td>
<td>The output of this command omits consecutive duplicate commands. Use the -all option to include the consecutive duplicate commands.</td>
</tr>
<tr>
<td></td>
<td>Use the -h option to omit the command numbers.</td>
</tr>
<tr>
<td></td>
<td>Use the -r parameter to list the commands in reverse order.</td>
</tr>
<tr>
<td></td>
<td>The history list stores up to 100 of the most recently executed commands.</td>
</tr>
<tr>
<td></td>
<td>See the savehistory and clearhistory command.</td>
</tr>
<tr>
<td>host os_command</td>
<td>Executes an operating system command. The command is executed in the same console as ttiSql.</td>
</tr>
<tr>
<td></td>
<td>This command sets the environment variable TT_CONNSTR in the environment of the process it creates.</td>
</tr>
<tr>
<td></td>
<td>The value of the variable is the connection string of the current connection.</td>
</tr>
<tr>
<td></td>
<td>To see the exit status of the command, use the define command with _EXIT_STATUS.</td>
</tr>
<tr>
<td>if-then-else</td>
<td>The if-then-else command construct enables you to implement conditional branching logic in a ttiSql session. For more details, see &quot;Syntax for the IF-THEN-ELSE command construct&quot; on page 5-79.</td>
</tr>
<tr>
<td>indexes [table_name_pattern]</td>
<td>Describes the indexes that it finds on the tables owned by the current user that match the input pattern. When a name pattern is missing, the pattern defaults to %.</td>
</tr>
<tr>
<td></td>
<td>If passthrough is enabled, lists indexes on tables matching the pattern in the Oracle database.</td>
</tr>
<tr>
<td></td>
<td>See the allindexes command.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>packages [object_name_pattern]</td>
<td>Lists, in a single column, the names of PL/SQL packages owned by the current user that match the given pattern. When a name pattern is missing, the pattern defaults to %. If passthrough is enabled, lists PL/SQL packages matching the pattern in the Oracle database. See the allpackages command.</td>
</tr>
<tr>
<td>prepare [connid.]command_id.SQL_Statement</td>
<td>Prepares the specified SQL statement. If the command_id argument is not specified, the command_id is assigned automatically. The command_id argument can take a value between 0 and 255 inclusive. If connid is specified, switches to the given connection ID. The connid must be only alphanumeric characters and are case insensitive. Use free to release the prepared command.</td>
</tr>
<tr>
<td>print [variable]</td>
<td>Prints the value of the specified bind variable or all variables if no variable is specified. If the variable is a REF CURSOR, then the results are fetched and printed.</td>
</tr>
<tr>
<td>procedures [procedure_name_pattern]</td>
<td>Lists, in a single column, the names of PL/SQL procedures owned by the current user that match the given pattern. When a name pattern is missing, the pattern defaults to %. If passthrough is enabled, lists PL/SQL procedures matching the pattern in the Oracle database. See the builtins and allprocedures commands.</td>
</tr>
<tr>
<td>quit</td>
<td>Exits ttIsql.</td>
</tr>
<tr>
<td>remark msg</td>
<td>Specifies that the message on the line should be treated as a comment. When rem or remark is the first word on the line, ttIsql reads the line and ignores it.</td>
</tr>
<tr>
<td>repschemes [scheme_owner_pattern.] scheme_name_pattern</td>
<td>Reports information on replication schemes defined in the currently connected data source. This information describes all elements associated with the replication schemes. If the optional argument is not specified then information on all replication schemes defined in the current data source is reported.</td>
</tr>
<tr>
<td>retryconnect [0</td>
<td>1]</td>
</tr>
<tr>
<td>rollback</td>
<td>Rolls back the current transaction. AutoCommit must be off. This command does not stop TimesTen Cache operations on the Oracle database, including passthrough statements, flushing, manual loading, manual refreshing, synchronous writethrough, propagating and dynamic loading.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| rpad varname desiredlength paddingstring | The `RPAD` command acts like the SQL function `RPAD()` with some limitations:  
- The desired length is in bytes, not characters.  
- The padding string is not expanded for string literal escapes, such as unicode escapes.  
- The padding string can contain partial unicode characters or full unicode characters and it may split the padding string in the middle of a multibyte character or surrogate pair.  
Only variables that are character based (CHAR, VARCHAR) can be padded with the `RPAD` command. |
| run filename [arguments]| Reads and executes SQL commands from `filename`. The run command can be nested up to five levels. |
| start filename [arguments...]| The `@@` command is identical to the `@` command only if the file is specified with an absolute path. |
| @@ filename [arguments...]| When you specify `@` with a relative path, the path is relative to the startup directory of `ttIsql`. When you specify `@@`, the path is relative to the currently running input file. Therefore `@@` is useful when used in a script that must call other scripts. It does not matter what directory the invoker of `ttIsql` is in when the script is run.  
See "Example parameters of command string substitution" on page 5-91 for a description of `arguments`. |
| savehistory [-all [-h] [-a | -f] outputfile | Writes the history buffer to the specified `outputfile`.  
Consecutive duplicate commands are omitted.  
Use the `-all` option to include the consecutive duplicate commands.  
Use the `-h` option to omit the command numbers.  
Use `-a` to append to an existing output file. Use `-f` to force the overwriting of an existing output file.  
See the `clearhistory` and `history` commands. |
| sequences [sequence_name_ pattern] | Lists, in a single column, the names of sequences owned by the current user that match the given pattern. When a name pattern is missing, the pattern defaults to `%`.  
If passthrough is enabled, lists sequences on tables matching the pattern in the Oracle database.  
See the `allsequences` command. |
| set attribute [value] | Sets the specified `set/show` attribute to the specified value.  
If no value is specified, displays the current value of the specified attribute.  
For a description of accepted attributes, see "Set/show attributes" on page 5-82. |
<p>| setjoinorder tblNames [...] | Specifies the join order for the optimizer. <code>AutoCommit</code> must be off. |
| setuseindex index_ name,correlation_name, (0 | 1) [...] | Sets the index hint for the query optimizer. |</p>
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>setvariable variable_name := value</td>
<td>Sets the value of a scalar bind variable or an element of an array bind variable. For example: <code>setvariable myvar := 'TimesTen';</code> There must be a space on either side of the assignment operator (:=). For more information, see &quot;Declaring and setting bind variables&quot; in the Oracle TimesTen In-Memory Database Operations Guide.</td>
</tr>
<tr>
<td>show {all</td>
<td>attribute}</td>
</tr>
<tr>
<td>showjoinorder {0</td>
<td>1}</td>
</tr>
<tr>
<td>sleep [n]</td>
<td>Suspends execution for n seconds. If n is not specified then execution is suspended for 1 second.</td>
</tr>
<tr>
<td>spool filename [option</td>
<td>OFF]</td>
</tr>
<tr>
<td>sqlcolumns [owner_name_pattern.]table_name_pattern</td>
<td>Prints results of an ODBC call to SQLColumns.</td>
</tr>
<tr>
<td>sqlgetinfo infotype</td>
<td>Prints results of an ODBC call to SQLGetInfo.</td>
</tr>
<tr>
<td>sqlstatistics [owner_name_pattern.]table_name_pattern</td>
<td>Prints results of an ODBC call to SQLStatistics.</td>
</tr>
<tr>
<td>sqltables [owner_name_pattern.]table_name_pattern</td>
<td>Prints results of a call to SQLTables. The pattern is a string containing an underscore (_) to match any single character or a percent sign (%) to match zero or more characters.</td>
</tr>
<tr>
<td>statsclear [owner_name.]table_name</td>
<td>Clears statistics for specified table (or all tables if no table is specified).</td>
</tr>
<tr>
<td>statsestimate [owner_name.]table_name {n rows</td>
<td>p percent}</td>
</tr>
<tr>
<td>statsupdate [owner_name_pattern.]table_name_pattern</td>
<td>Updates statistics for specified table (or all tables if no table is specified). If tblName is an empty string, statistics are estimated for all the current user’s tables in the database.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| synonyms \[[schema_pattern.]
object_pattern]\]      | Lists, in a single column, the names of synonyms owned by the current user that match the given pattern. When a name pattern is missing, the pattern defaults to \%.
If passthrough to an Oracle database is enabled, lists synonyms on tables matching the pattern in the Oracle database.
See the allsynonyms command.                                                                                                                   |
| tables [table_name_pattern] | Lists, in a single column, the names of tables owned by the current user that match the given pattern. When a name pattern is missing, the pattern defaults to \%.
If passthrough to an Oracle database is enabled, lists tables matching the pattern in the Oracle database.
See the alltables command.                                                                                                                      |
| tablesize \[[owner_name_pattern.] table_name_pattern\] | For each table that matches the pattern, lists the contents of the ALL_TAB_SIZES view.
See the ttComputeTabSizes built-in procedure.                                                                                                  |
| undefine name          | Undefines a string substitution alias.                                                                                                                                                                      |
| unsetjoinorder         | Clears join order advice to optimizer. AutoCommit must be off.                                                                                                                                              |
| unsetuseindex          | Clears the index hint for the query optimizer.                                                                                                                                                              |
| use [conn_id]          | Displays the list of current connections and their IDs. If connid is specified, switches to the given connection ID.                                                                                         |
|                        | To use the name of the first connection, you can specify con0 for the conn_id, rather than specifying the full original connection name. You cannot explicitly name a connection con0. If the first connection is disconnected, con0 refers to the connection none. |
|                        | If use fails to locate the connection id, the current connection is set to the reserved connection named “none.”                                                                                           |
|                        | See the connect command.                                                                                                                                                                                   |
| variable [variable_name [data_type] [:= value]] | Declares a bind variable that can be referenced in a statement or displays the definition of the variable if the type is missing. Type can be one of the following: (n), NUMBER, CHAR (n), VARCHAR2 (n), NVARCHAR2 (n), BLOB, CLOB, NCLOB, or REFCURSOR. If only (n) is supplied, it is assumed to be VARCHAR2 (n).
Assigns a value to a single variable or multiple values if the data type is an array. You can assign a value later with the setvariable command.
For more information, see “Declaring and setting bind variables” in the Oracle TimesTen In-Memory Database Operations Guide. |
| version                | Reports version information.                                                                                                                                                                               |
| views [table_name_pattern] | Lists, in a single column, the names of views owned by the current user that match the given pattern. When a name pattern is missing, the pattern defaults to \%. If passthrough to an Oracle database is enabled, lists views matching the pattern in the Oracle database.
See the allviews command.                                                                                                                     |
### Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>waitfor expected_result</code></td>
<td>Runs the given statement once a second until the query returns the expected result or a timeout occurs. The query must have only one column and must return exactly one row. Any errors in the query terminate the loop.</td>
</tr>
<tr>
<td><code>timeoutseconds sqlstatement</code></td>
<td></td>
</tr>
<tr>
<td><code>waitforresult expected_result</code></td>
<td>Similar to the <code>waitfor</code> command, except that the result can have 1 or more columns. Also, the result can return 0 rows. Runs the given statement once a second until the query returns the expected result or a timeout occurs. The <code>searchrow</code> and <code>searchcol</code> arguments indicate the ordinal position (1..N) of which row or column should be considered. Use <code>'*'</code> in <code>searchrow</code> or <code>searchcol</code> to indicate any row or column of the result set could have the expected value. See the <code>waitfor</code> command.</td>
</tr>
<tr>
<td><code>timeoutseconds searchrow</code></td>
<td></td>
</tr>
<tr>
<td><code>searchcol sqlstatement</code></td>
<td></td>
</tr>
<tr>
<td><code>whenever sqlerror</code></td>
<td>Provide direction on how to handle errors when in <code>ttIsql</code>. For more details, see &quot;Syntax for the WHENEVER SQLERROR command&quot; on page 5-80.</td>
</tr>
<tr>
<td><code>xlabookmarkdelete id</code></td>
<td>Deletes a persistent XLA bookmark. If a bookmark to delete is not specified then the status of all current XLA bookmarks is reported. Also see &quot;ttXlaDeleteBookmark&quot; in Oracle TimesTen In-Memory Database C Developer’s Guide. Requires ADMIN privilege or object ownership.</td>
</tr>
</tbody>
</table>

---

**Note:**

- Commands are case-sensitive.
- Ensure the correct syntax and parameters are used when executing these commands.
- Always test and validate the results to ensure accurate operation.
Syntax for the IF-THEN-ELSE command construct

This section provides the syntax for the IF-THEN-ELSE construct. For more details on using the IF-THEN-ELSE command construct, see "Conditional control with the IF-THEN-ELSE command construct" in the Oracle TimesTen In-Memory Database Operations Guide.

```
IF [NOT]
  { Literal1 | :BindVariable1 }
  ( = | IN )
  { Literal2 | :BindVariable2 | SelectStatement }
THEN 'ThenCommands'
[ ELSE 'ElseCommands' ] ;
```

The ttIsql IF-THEN-ELSE command has the parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IF</td>
<td>The IF command must end in a semicolon (;). The IF command fails if improper syntax is given, the BindVariables do not exist or the SELECT statement fails to execute or does not return just a single column.</td>
</tr>
<tr>
<td>NOT</td>
<td>Using NOT reverses the desired result of the condition.</td>
</tr>
<tr>
<td>Literal1,Literal2</td>
<td>A value that can be part of a comparison.</td>
</tr>
<tr>
<td>BindVariable1,BindVariable2</td>
<td>A bind variable is equivalent to a parameter. You can use the :BindVariable1 notation for passing bind variables into this construct. The variable can be created and set using the variable or setvariable ttIsql commands.</td>
</tr>
<tr>
<td>=</td>
<td>IN</td>
</tr>
<tr>
<td>SelectStatement</td>
<td>A provided SELECT statement must start with SELECT. The SELECT statement can return only one column. In addition, it can return only one row when the equal (=) operator is provided. The SelectStatement is not available if you are not connected to the database.</td>
</tr>
<tr>
<td>ThenCommands,ElseCommands</td>
<td>All commands in the THEN or ELSE clauses must be delimited by a semicolon and cannot contain embedded double quotes. These clauses can conditionally execute ttIsql commands, such as host or run, which cannot be executed through PL/SQL. You can use the CALL statement within the THEN or ELSE clauses. You cannot use PL/SQL blocks.</td>
</tr>
</tbody>
</table>

Restrictions for the IF-THEN-ELSE construct are as follows:

- You cannot compare variables of the LOB data type.
- The values are compared case-sensitive with strcmp. A character padded value might not match a VARCHAR2 because of the padding.
Syntax for the WHENEVER SQLERROR command

Execute the WHENEVER SQLERROR command to prescribe what to do when a SQL error occurs. For more details and examples on how to use the WHENEVER SQLERROR command, see "Error recovery with WHENEVER SQLERROR" command in the Oracle TimesTen In-Memory Database Operations Guide.

WHENEVER SQLERROR { ExitClause | ContinueClause | SUPPRESS | SLEEP Number | ExecuteClause }

When you specify EXIT, always exit ttIsql if an error occurs. ExitClause is as follows:

EXIT | SUCCESS | FAILURE | WARNING | Number | :BindVariable |
[ COMMIT | COMMIT ALL | ROLLBACK ]

When you specify CONTINUE, ttIsql continues to the next command, even if an error occurs. ContinueClause is as follows:

CONTINUE | COMMIT | COMMIT ALL | ROLLBACK | NONE |

Execute specified commands before continuing. ExecuteClause is as follows:

EXECUTE "Cmd1;Cmd2;...;"

The WHENEVER SQLERROR command options are as follows:

- **EXIT**: Always exit ttIsql if an error occurs. Specify what is performed before ttIsql exits with one of the following. SUCCESS is the default option for EXIT.
  - SUCCESS or FAILURE or WARNING: Return SUCCESS (value 0), FAILURE (value 1), or WARNING (value 2) to the operating system after ttIsql exits for any SQL error.
  - Number: Specify a number from 0 to 255 that is returned to the operating system as a return code. Once ttIsql exits, you can retrieve the error return code with the appropriate operating system commands. For example, use `echo $status` in the C shell (csh) or `echo $?` in the Bourne shell (sh) to display the return code.

  The return code can be retrieved and processed within batch command files to programmatically detect and respond to unexpected events.
  - :BindVariable: Returns the value in a bind variable that was previously created in ttIsql with the variable command. The value of the variable at the time of the error is returned to the operating system in the same manner as the Number option.

**Note:** The bind variable used within the WHENEVER SQLERROR command cannot be defined as a LOB, REFCURSOR, or any array data type.

In addition, you can specify whether to commit or rollback all changes before exiting ttIsql.

- **COMMIT**: Executes a COMMIT and saves changes only in the current connection before exiting. The other connections exit with the normal disconnect processing, which rolls back any uncommitted changes.
- **COMMIT ALL**: Executes a COMMIT and saves changes in all connections before exiting.
- **ROLLBACK**: Before exiting, executes a ROLLBACK and abandons changes in the current connection and, by default, in all other connections. The other connections exit with the normal disconnect processing, which automatically rolls back any uncommitted changes.

- **CONTINUE**: Do not exit if an error occurs. The SQL error is displayed, but the error does not cause ttIsq1 to exit. The following options enable you to specify what is done before continuing to the next ttIsq1 command:
  - **NONE**: This is the default. Take no action before continuing.
  - **COMMIT**: Executes a COMMIT and saves changes in the current connection before continuing.
  - **COMMIT ALL**: Executes a COMMIT and saves changes in all connections before continuing.
  - **ROLLBACK**: Before continuing, executes a ROLLBACK and abandons changes in the current connection and, by default, in all other connections. The other connections exit with the normal disconnect processing, which automatically rolls back any uncommitted changes.

- **SUPPRESS**: Do not show any error messages and continue.

- **SLEEP**: Sleep for a specified number of seconds before continuing.

- **EXECUTE**: Execute specified commands before continuing. Each command is separated from the other commands by a semicolon (;). If any command triggers additional errors, those errors may cause additional actions that could potentially result in a looping condition.
Set/show attributes

Also see the list of ttIsql "Commands" on page 5-66. Some commands appear here as attributes of the set command. In that case, you can use them with or without the set command.

Boolean attributes can accept the values "ON" and "OFF" or "1" and "0".

The ttIsql set command has the attributes:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td>With show command only. Displays the setting of all the ttIsql commands.</td>
</tr>
<tr>
<td>autocommit [1</td>
<td>0]</td>
</tr>
<tr>
<td>autovariables [1</td>
<td>0]</td>
</tr>
<tr>
<td>columnlabels [0</td>
<td>1]</td>
</tr>
<tr>
<td>connstr</td>
<td>Prints the connection string returned from the driver from the SQLDriverConnect call. This is the same string printed when ttIsql successfully connects to a database.</td>
</tr>
<tr>
<td>define [&amp;</td>
<td>c</td>
</tr>
<tr>
<td>dynamicloadenable [1</td>
<td>0]</td>
</tr>
<tr>
<td>echo [on</td>
<td>off]</td>
</tr>
</tbody>
</table>
Utilities 5-83

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`editline [0</td>
<td>1]`</td>
</tr>
<tr>
<td></td>
<td>If <code>editline</code> is turned off, the backspace character deletes full characters, but the rest of <code>editline</code> capabilities are unavailable.</td>
</tr>
<tr>
<td>`err</td>
<td>error</td>
</tr>
<tr>
<td></td>
<td>If no object type or object name is supplied, <code>ttIsql</code> assumes the PL/SQL object that you last attempted to create and retrieves the errors for that object.</td>
</tr>
<tr>
<td></td>
<td>If no errors associated with the given object are found, or there was no previous PL/SQL DDL, then <code>ttIsql</code> displays &quot;No errors.&quot;</td>
</tr>
<tr>
<td>`feedback [on</td>
<td>off] rows`</td>
</tr>
<tr>
<td></td>
<td>When <code>rows</code> is specified, if the statement affected more than the specified number of rows, then the feedback indicates the number of affected rows. If the number of rows affected is less than the specified threshold, the number of rows is not printed.</td>
</tr>
<tr>
<td></td>
<td>Feedback is not provided for tables, views, sequences, materialized views or indexes. It is available for PL/SQL objects.</td>
</tr>
<tr>
<td>`isolation [(READ_COMMITTED</td>
<td>1)] [SERIALIZABLE</td>
</tr>
<tr>
<td></td>
<td>You can also enable this attribute without specifying the <code>set</code> command.</td>
</tr>
<tr>
<td><code>loboffset n</code></td>
<td>Specifies the offset into the LOB that <code>ttIsql</code> should use as the starting point when it prints the resulting value of a LOB. For example if the value of the LOB is ABCEDFG, and the offset is 4, <code>ttIsql</code> prints DEFG, skipping the first 3 bytes.</td>
</tr>
<tr>
<td></td>
<td>The behavior is the same as <code>LOBOFFSET</code> in SQL*Plus.</td>
</tr>
<tr>
<td><code>long n</code></td>
<td>Reports or controls the maximum number of characters for CLOB or BLOB data or the maximum number of bytes for BLOB data that are displayed when fetched or printed.</td>
</tr>
<tr>
<td></td>
<td>The default value is 80.</td>
</tr>
<tr>
<td></td>
<td>The command setting is valid for all connections in a session.</td>
</tr>
<tr>
<td><code>longchunksize n</code></td>
<td>Specifies the size of the chunk that <code>ttIsql</code> uses to get LOB data.</td>
</tr>
<tr>
<td>`multipleconnections [1</td>
<td>ON] mc [1</td>
</tr>
<tr>
<td></td>
<td>If the argument 1 or ON is specified the prompt is changed to include the current connection and all multiple connection features are enabled.</td>
</tr>
<tr>
<td></td>
<td>If no value is supplied, the command displays the value of the <code>multipleconnections</code> setting.</td>
</tr>
<tr>
<td></td>
<td>You can also enable this attribute without specifying the <code>set</code> command.</td>
</tr>
</tbody>
</table>
**Attribute** | **Description**  
--- | ---  
`ncharencoding [encoding]` | Specifies the character encoding method for NCHAR output. Valid values are `locale` or `ASCII`.  
   | `locale` sets the output format to the locale-based setting.  
   | If no value is specified, TimesTen uses the system’s native language characters.  
   | You can also enable this attribute without specifying the `set` command.  

`nulldisplaystring "string"` | Sets or shows the string to be displayed when the NULL value appears in a result set.  
   | The option does not affect the SQL user, only the display of NULL in results sets.  

`optfirstrow [1|0]` | Enables or disables First Row Optimization.  
   | If the optional argument is omitted, First Row Optimization is enabled.  
   | You can also enable this attribute without specifying the `set` command.  

`optprofile` | Prints the current optimizer flag settings and join order.  
   | This attribute cannot be used with the `set` command.  

`passthrough [0|1|2|3]` | Sets the TimesTen Cache passthrough level for the current transaction. Because `AutoCommit` must be off to execute this command, `ttIsql` temporarily turns off `AutoCommit` when setting the passthrough level.  
   | 0 - SQL statements are executed only against TimesTen.  
   | 1 - Statements other than `INSERT`, `DELETE` or `UPDATE` and DDL are passed through if they generate a syntax error in TimesTen or if one or more tables referenced within the statement are not in TimesTen. All `INSERT`, `DELETE` and `UPDATE` statements are passed through if the target table cannot be found in TimesTen. DDL statements are not passed through.  
   | 2 - Same as 1, plus any `INSERT`, `UPDATE` and `DELETE` statement performed on `READONLY` cache group tables is passed through.  
   | 3 - All SQL statements, except `COMMIT` and `ROLLBACK`, and TimesTen built-in procedures that set or get optimizer flags are passed through. `COMMIT` and `ROLLBACK` are executed on both TimesTen and the Oracle database.  
   | If no optional argument is supplied, the current setting is displayed.  
   | After the transaction, the passthrough value is reset to the value defined in the connection string or in the DSN or the default setting if no value was supplied to either.  
   | You can also enable this attribute without specifying the `set` command.  
   | **Note:** Some Oracle objects may not be described by `ttIsql`. 

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>prefetchcount [prefetch_count_size]</code></td>
<td>Sets the prefetch count size for the current connection. If the optional argument is omitted, the current prefetch count size is reported. Setting the prefetch count size can improve result set fetch performance. The <code>prefetch_count_size</code> argument can take an integer value between 0 and 128 inclusive. When you set the prefetch count to 0, TimesTen uses a default prefetch count. The default prefetch value is isolation level specific. In read committed isolation mode, the default value is 5. In serializable isolation mode, the default value is 128. You can also enable this attribute without specifying the <code>set</code> command.</td>
</tr>
<tr>
<td><code>prompt [string]</code></td>
<td>Replaces the <code>Command&gt;</code> prompt with the specified string. To specify a prompt with spaces, you must quote the string. The leading and trailing quotes are removed. A prompt can have a string format specifier (%c) embedded. The %c is expanded with the name of the current connection.</td>
</tr>
<tr>
<td><code>querythreshold [seconds]</code></td>
<td>With the <code>show</code> command, displays the value of the Query Threshold first connection attribute. With the <code>set</code> command, modifies the value of the QueryThreshold first connection attribute that was set in the connection string or odbc.ini file. Specify a value in seconds that indicates the number of seconds that a query can execute before TimesTen writes a warning to the support log.</td>
</tr>
<tr>
<td>`rowdelimiters [0</td>
<td>off]</td>
</tr>
<tr>
<td>`serveroutput [on</td>
<td>off]`</td>
</tr>
<tr>
<td>`showcurrenttime [1</td>
<td>true</td>
</tr>
<tr>
<td>`showplan [0</td>
<td>1]`</td>
</tr>
</tbody>
</table>
### Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sqlquerytimeout [seconds]</td>
<td>Specifies the number of seconds to wait for a SQL statement to execute before returning to the application for all subsequent calls. If no time or 0 seconds is specified, displays the current timeout value. The value of seconds must be equal to or greater than 0. This attribute does not stop TimesTen Cache operations on the Oracle database, including passsthrough statements, flushing, manual loading, manual refreshing, synchronous writethrough, propagating, and dynamic loading. You can also enable this attribute without specifying the set command. See “Choose SQL and PL/SQL timeout values” in Oracle TimesTen In-Memory Database Operations Guide for information about the relationship between the client timeout, SQL timeout, and PL/SQL timeout.</td>
</tr>
<tr>
<td>timing [1</td>
<td>0]</td>
</tr>
<tr>
<td>tryhash [1</td>
<td>0]</td>
</tr>
<tr>
<td>trymaterialize [1</td>
<td>0]</td>
</tr>
<tr>
<td>trymergejoin [1</td>
<td>0]</td>
</tr>
<tr>
<td>trynestedloopjoin [1</td>
<td>0]</td>
</tr>
<tr>
<td>tryrowid [1</td>
<td>0]</td>
</tr>
<tr>
<td>tryrowlocks [1</td>
<td>0]</td>
</tr>
<tr>
<td>tryserial [1</td>
<td>0]</td>
</tr>
<tr>
<td>trytmphash [1</td>
<td>0]</td>
</tr>
<tr>
<td>Attribute</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>`trytblocks [1</td>
<td>0]`</td>
</tr>
<tr>
<td>`trytmpetable [1</td>
<td>0]`</td>
</tr>
<tr>
<td>`trytmprange [1</td>
<td>0]`</td>
</tr>
<tr>
<td>`tryrange [1</td>
<td>0]`</td>
</tr>
<tr>
<td><code>verbosity [level]</code></td>
<td>Changes the verbosity level. The verbosity level argument can be an integer value of 0, 1, 2, 3 or 4. If the optional argument is omitted then the current verbosity level is reported. You can also enable this attribute without specifying the <code>set</code> command.</td>
</tr>
<tr>
<td>`vertical [{0</td>
<td>off}</td>
</tr>
</tbody>
</table>
Comment syntax

The types of comment markers are:

`-- [comment_text]`

`/* [comment_text] */`

The C-style comments, delineated by "/*" at the beginning and "*/" at the end, can span multiple lines.

The comments delimited by the `-` character should not span multiple lines. If a comment marker is encountered while processing a line, ttIsql ignores the remainder of the line.

`'--'` at the beginning of a line is considered a SQL comment. The line is considered a comment and no part of the line is included in the processing of the SQL statement. A line that begins with `'--+'` is interpreted as a segment of a SQL statement.

The comment markers can work in the middle of a line.

Example:

`monitor; /*this is a comment after a ttIsql command*/`
**Command shortcuts**

By default, ttIsq1 supports keystroke shortcuts when entering commands. To turn this feature off, use:

```
Command> set editline=0;
```

The ttIsq1 keystroke shortcuts are:

<table>
<thead>
<tr>
<th>Keystroke</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left Arrow</td>
<td>Moves the insertion point left (back).</td>
</tr>
<tr>
<td>Right Arrow</td>
<td>Moves the insertion point right (forward).</td>
</tr>
<tr>
<td>Up Arrow</td>
<td>Scroll to the command before the one being displayed. Places the cursor at the end of the line. If the command being added to the history is identical to the most recently added command, it is skipped.</td>
</tr>
<tr>
<td>Up Arrow &lt;RETURN&gt;</td>
<td>Scrolls to the PL/SQL block before the one being displayed.</td>
</tr>
<tr>
<td>Down Arrow</td>
<td>Scrolls to a more recent command history item and puts the cursor at the end of the line. If the command being added to the history is identical to the most recently added command, it is skipped.</td>
</tr>
<tr>
<td>Down Arrow &lt;RETURN&gt;</td>
<td>Scrolls to the next PL/SQL block after the one being displayed.</td>
</tr>
<tr>
<td>Ctrl-A</td>
<td>Moves the insertion point to the beginning of the line.</td>
</tr>
<tr>
<td>Ctrl-E</td>
<td>Moves the insertion point to the end of the line.</td>
</tr>
<tr>
<td>Ctrl-K</td>
<td>&quot;Kill&quot; - Saves and erases the characters on the command line from the current position to the end of the line.</td>
</tr>
<tr>
<td>Ctrl-Y</td>
<td>&quot;Yank&quot; - Restores the characters previously saved and inserts them at the current insertion point.</td>
</tr>
<tr>
<td>Ctrl-F</td>
<td>Forward character - move forward one character. (See Right Arrow.)</td>
</tr>
<tr>
<td>Ctrl-B</td>
<td>Backward character - moved back one character. (See Left Arrow.)</td>
</tr>
<tr>
<td>Ctrl-P</td>
<td>Previous history. (See Up Arrow.)</td>
</tr>
<tr>
<td>Ctrl-N</td>
<td>Next history. (See Down Arrow.)</td>
</tr>
</tbody>
</table>
Parameters

With dynamic parameters, you are prompted for input for each parameter on a separate line. Values for parameters are specified the same way literals are specified in SQL.

`SQL_TIMESTAMP` columns can be added using dynamic parameters. (For example, values like `'1998-09-08 12:12:12'`).

Parameter values must be terminated with a semicolon character.

The possible types of values that can be entered are:

- Numeric literals. Example: 1234.5
- Time, date or timestamp literals within single quotation marks. Examples:
  - `'12:30:00''2000-10-29''2000-10-29 12:30:00''2000-10-29 12:30:00.123456'`
- Unicode string literals within single quotation marks preceded by `N`. Example: `N'abc'`
- A NULL value. Example: `NULL`
- The '*' character that indicates that the parameter input process should be stopped. Example: `*`
- The '?' character prints the parameter input help information. Example: `?`
Examples

Example parameters of command string substitution

Command> select * from dual where :a > 100 and :b < 100;
Type '?' for help on entering parameter values.
Type '*' to end prompting and abort the command.
Type '-' to leave the parameter unbound.
Type '/;' to leave the remaining parameters unbound and execute the command.

Enter Parameter 1 'A' (NUMBER) > 110
Enter Parameter 2 'B' (NUMBER) > 99
< X >
1 row found.
Command> var a number;
   exec :a := 110;

PL/SQL procedure successfully completed.

Command> print a
A : 110
Command> var b number;
   exec :b := 99;

PL/SQL procedure successfully completed.

Command> select * from dual where :a > 100 and :b < 100;
< X >
1 row found.
Command> print
A : 110
B : 99
Command> select * from dual where :a > 100 and :b < 100 and :c > 0;
Enter Parameter 3 'C' (NUMBER) > 1
< X >
1 row found.

Default options

You can set the default command-line options by exporting an environment variable called TTISQL. The value of the TTISQL environment variable is a string with the same syntax requirements as the TTISQL command line. If the same option is present in the TTISQL environment variable and the command line then the command line version always takes precedence.

Examples

Execute commands from ttIsq1.inp.

```
ttIsq1 -f ttIsq1.inp
```

Enable all output. Connect to DSN RunData and create the database if it does not exist.

```
ttIsq1 -v 4 -connStr "DSN=RunData;AutoCreate=1"
```

Print the interactive commands.

```
ttIsq1 -helpcmds
```

Print the full help text.
ttIsql -helpfull

Display the setting for all ttIsql set/show attributes:

Command> show all;
Connection independent attribute values:

autoprint = 0 (OFF)
columnlabels = 0 (OFF)
define = 0 (OFF)
echo 1 (ON)
FEEDBACK ON
multipleconnections =0 (OFF)
ncharencoding = LOCALE (US7ASCII)
prompt = 'COMMAND>'
timing = 0 (OFF)
verbosity = 2
vertical = 0 (OFF)

Connection specific attribute values:

autocommit = 1 (ON)
Client timeout = 0
Connection String DSN=repdbl_1121;UID=timesten; DataStore=/DS/repdbl_1121;
DatabaseCharacterSet=AL32UTF8; ConnectionCharacterSet=US7ASCII;
DRIVER=/sw/tthome/install/lib/libtten.so; PermSize=20;TempSize=20;
No errors.
isolation = READ_COMMITTED
Prefetch count = 5
Query threshold = 0 seconds (no threshold)
Query timeout = 0 seconds (no timeout)
servoutput OFF

Current Optimizer Settings:

Scan: 1
Hash: 1
Range: 1
TmpHash: 1
TmpTable: 1
NestedLoop: 1
MergeJoin: 1
GenPlan: 0
TblLock: 1
RowLock: 1
Rowid: 1
FirstRow: 1
IndexedOr: 1
PassThrough: 0
BranchAndBound: 1
ForceCompile: 0
CrViewSemCheck: 1
ShowJoinOrder: 0
CrViewSemCheck: 1
UserBoyerMooreStringSearch: 0
DynamicLoadEnable: 1
DynamicLoadErrorMode: 0
NoRemRowIdOpt: 0

Current Join Order:
<>
**Prepare and execute an SQL statement.**

ttIsql (c) 1996-2011, TimesTen, Inc. All rights reserved.

**Example vertical command:**

```
Command> call ttlogholds;
< 0, 265352, Checkpoint , DS.ds0 >
< 0, 265408, Checkpoint , DS.ds1 >
2 rows found.
```

```
Command> vertical call ttlogholds;

HOLDLFN:       0
HOLDLFO:       265352
TYPE:          Checkpoint
DESCRIPTION:   DS.ds0
HOLDLFN:       0
HOLDLFO:       265408
TYPE:          Checkpoint
DESCRIPTION:   DS.ds1
2 rows found.
```

To create a new user, use single quotes around the password name for an internal user:

```
Command> CREATE USER terry IDENTIFIED BY `secret';
```

To delete the XLA bookmark mybookmark, use:

```
Command> xlabookmarkdelete;
XLA Bookmark: mybookmark
Read Log File:  0
Read Offset:    268288
Purge Log File: 0
Purge Offset:   268288
PID:            2004
In Use:         No
1 bookmark found.
```

```
Command> xlabookmarkdelete mybookmark;
```
Command> xlabookmarkdelete;

0 bookmarks found.

To run a SELECT query until the result "X" is returned or until the query times out at 10 seconds, use:

ttIsql -connStr 'DSN=RunData'

Example of managing XLA bookmarks

You can use the xlabookmarkdelete command to both check the status of the current XLA bookmarks and delete them. This command requires XLA privilege or object ownership.

For example, when running the XLA application, 'xlaSimple', you can check the bookmark status by entering:

Command> xlabookmarkdelete;

XLA Bookmark: xlaSimple
Read Log File: 0
Read Offset: 630000
Purge Log File: 0
Purge Offset: 629960
PID: 2808
In Use: No
1 bookmark found.

To delete the bookmark xlaSimple, enter:

Command> xlabookmarkdelete xlaSimple;

Example parameters using "variable" and "print"

Substitution in ttIsql is modeled after substitution in SQL*Plus. To enable the substitution feature, use set define on or set define substitution_char. The substitution character when the user specifies 'on' is '\&'. It is disabled with 'set define off'.

By default, substitution is off. The default is off because the & choice for substitution character conflicts with TimesTen's use of ampersand as the BIT AND operator.

When enabled, the alphanumeric identifier following the substitution character is replace by the value assigned to that identifier. When disabled, the expansion is not performed.

New definitions can be defined even when substitution is off. You can use the define command to list the definitions ttIsql predefines.

Command> show define
define = 0 (OFF)
Command> define
DEFINE   _PID = '9042' (CHAR)
DEFINE   _O_VERSION = 'TimesTen Release 11.2.1.0.0' (CHAR)
Command> select '&_O_VERSION' from dual;
Utilities

< &_O_VERSION >
1 row found.
Command> set define on
        SELECT '&_O_VERSION' FROM DUAL;
< TimesTen Release 11.2.1.0.0 >
1 row found.

If the value is not defined, ttIsq1 prompts you for the value.

When prompting with only one substitution character specified before the identifier,
the identifier is defined only for the life of the one statement.

If two substitution characters are used and the value is prompted, it acts as if you have
explicitly defined the identifier.

Command> SELECT '&a' FROM DUAL;
Enter value for a> hi
< hi >
1 row found.
Command> define a
symbol a is UNDEFINED
The command failed.
Command> SELECT '&&a' FROM DUAL;
Enter value for a> hi there
< hi there >
1 row found.
Command> define a
DEFINE               a = "hi there" (CHAR)

Additional definitions are created with the define command:

Command> define tblname = sys.dual
        define tblname
        DEFINE         tblname = "sys.dual" (CHAR)
Command> select * from &tblname;
< X >
1 row found.

Arguments to the run command are automatically defined to '&1', '&2',... when you
add them to the run or @ (and @@) commands:

Given this script:
CREATE TABLE &1 ( a INT PRIMARY KEY, b CHAR(10) );
INSERT INTO &1 VALUES (1, '&2');
INSERT INTO &1 VALUES (2, '&3');SELECT * FROM &1;

Use the script:
Command> SET DEFINE ON
Command> @POPULATE mytable Joe Bob;

CREATE TABLE &1 ( a INT PRIMARY KEY, b CHAR(10) );
INSERT INTO &1 VALUES (1, '&2');
1 row inserted.

INSERT INTO &1 VALUES (2, '&3');
1 row inserted.

SELECT * FROM &1;
< 1, Joe  >
< 2, Bob  >
2 rows found.
This example uses the variable command. It deletes an employee from the employee table. Declare empid and name as variables with the same data types as employee_id and last_name. Delete the row, returning employee_id and last_name into the variables. Verify that the correct row was deleted.

Command> VARIABLE empid NUMBER(6) NOT NULL;
VARIABLE name VARCHAR2(25) INLINE NOT NULL;
DELETE FROM employees WHERE last_name='Ernst'
RETURNING employee_id, last_name INTO :empid,:name;
1 row deleted.
Command> PRINT empid name;
EMPID : 104
NAME : Ernst

Notes

The ttIsql utility supports only generic REF CURSOR variables, not specific REF CURSOR types.

The ttIsql utility command line accepts multiline PL/SQL statements, such as anonymous blocks, that are terminated with the "/" on it's own line. For example:

Command> set serveroutput on
BEGIN
dbms_ouput.put_line ('Hi There');
END;
/
Hi There
PL/SQL block successfully executed.

Command>

For UTF-8, NCHAR values are converted to UTF-8 encoding and then output.

For ASCII, those NCHAR values that correspond to ASCII characters are output as ASCII. For those NCHAR values outside of the ASCII range, the escaped Unicode format is used. For example:

U+3042 HIRAGANA LETTER A

is output as

Command> SELECT c1 FROM t1;
< a\u3042 >

NCHAR parameters must be entered as ASCII N-quoted literals:

Command> prepare SELECT * FROM t1 WHERE c1 = ?;
exec;

Type '";' for help on entering parameter values. Type '";' to stop the parameter entry process.

Enter Parameter 1> N'XY';

On Windows, this utility is supported for all TimesTen Data Manager and Client DSNs.
ttMigrate

Description

Performs one of these operations:

- Saves a migrate object from a TimesTen database into a binary data file.
- Restores the migrate object from the binary data file into a TimesTen database.
- Examines the contents of a binary data file created by this utility.

Migrated objects include:

- Tables
- Cache group definitions
- Views and materialized views
- Materialized view log definitions
- Sequences
- Replication schemes

Use the ttMigrate utility when upgrading major release versions of TimesTen, since database checkpoint and log files are not compatible between major releases.

For an example, see the Oracle TimesTen In-Memory Database Installation, Migration, and Upgrade Guide.

When you migrate a database into Release 11.2.1 from a previous release, TimesTen does not migrate users and user privileges. When you migrate a database between releases of Release 11.2.1 or into a release later than Release 11.2.1, TimesTen migrates users and user privileges.

Binary files produced by this utility are platform-dependent. For example a binary file produced on Windows must be restored on Windows. In client/server mode, use ttMigrateCS (UNIX and Linux systems only) utility to copy data between platforms.

Binary files produced by this utility are platform-specific. For example, a binary file produced on Windows 64-bit must be restored on Windows 64-bit. To copy data between platforms or bit levels, use ttMigrate with the ttMigrateCS client/server version (or Windows equivalent). On Windows systems, you can do the equivalent by using ttMigrate to connect to the source system from the target system through a defined TimesTen client DSN.

On UNIX and Linux systems, this utility is supported for TimesTen Data Manager DSNs. For TimesTen Client DSNs, use the utility ttMigrateCS.

Required privilege

This utility requires various privileges depending on the options specified. In general, a user must be the instance administrator or have the ADMIN privilege to use this utility.

Using the -r option requires the instance administrator privilege, as it generally creates a database. If the database has been created at the time this option is used, it requires CREATE ANY TABLE, CREATE ANY SEQUENCE, CREATE ANY VIEW, CREATE ANY MATERIALIZED VIEW, CREATE ANY CACHE GROUP, CREATE ANY INDEX privileges and ADMIN if autocreation of users is necessary. If the database is involved in replication or TimesTen Cache, then CACHE_MANAGER is also required.
Using the `-c` option to capture an entire database requires the `ADMIN` privilege. If the database is involved in replication or TimesTen Cache, then `CACHE_MANAGER` is also required. Using the `-c` option to capture a subset of the database objects (tables, views, materialized views, cache groups, sequences) requires `SELECT ANY TABLE` and `SELECT ANY SEQUENCE` privileges.

**Usage with TimesTen Scaleout**

This utility is supported for migrating from a TimesTen Classic to a TimesTen Scaleout. After the initial migration, this utility is not supported.

**Syntax**

```
ttmigrate {-h | -help | -?}
ttmigrate {-V | -version}
```

To create or append a binary data file, use:

```
ttmigrate {-a | -c} [-v verbosity] [-nf] [-nr] [-fixNaN] [-saveAsCharset charset] [-relaxedUpgrade | -exactUpgrade] [-activeDML | -noActiveDML] {DSN | -connstr connection_string} data file [objectOwner.]objectName
```

To restore a database from a binary data file created by this utility, use:

```
```

To list or display the contents of a binary data file created by this utility, use:

```
ttmigrate {-l | -L | -d | -D} dataFile [[objectowner.]name ...]
```

**Options**

```
Note: The append (-a) or create (-c) modes, the list (-1/-L) or describe (-d/-D) modes and the restore (-r) modes are exclusive of each other. You cannot specify any of these options on the same line as any other of these options.
```

**ttMigrate has the options:**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-a</code></td>
<td>Selects append mode: Appends data to a pre-existing binary data file, that was originally created using <code>ttMigrate -c</code>. See &quot;Create mode (-c) and Append mode (-a)&quot; on page 5-102 for more details.</td>
</tr>
</tbody>
</table>
Saves all tables in a foreign key hierarchy in a single transaction, maintaining consistency between these tables when there is active DML during the `ttMigrate -c` operation.

If `-noActiveDML` is specified, `ttMigrate` saves each table in its own transaction, regardless of whether it is the parent or the child of a foreign key. Use this option if there is no active DML during the `ttMigrate -c` operation.

`-noActiveDML` is the default.

Create mode: Creates an original binary data file. See "Create mode (-c) and Append mode (-a)" on page 5-102 for more details.

The cache administration user ID to use when restoring asynchronous writethrough cache groups and cache groups with the `AUTOREFRESH` attribute.

The cache administration password to use when restoring autorefresh and asynchronous writethrough cache groups and cache groups with the `AUTOREFRESH` attribute.

If the cache administration user ID is provided on the command line but the cache administration password is not, then `ttMigrate` prompts for the password.

An ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.

Selects Describe mode. Displays a short description of the objects in the data file. See "Describe mode (-d)" on page 5-105 for more details.

Selects Long-describe mode. Displays a full description of the objects in the data file. See "Long-describe mode (-D)" on page 5-105 for more details.

The path name of the data file to which migrate objects are to be saved or from which migrate objects are to be restored.

Specifies an ODBC data source name of the database to be migrated.

Specifies that `ttMigrate` should estimate statistics on restored tables and materialized views for the specified percentage of rows. Legal values for `percentRows` are 0 to 100, inclusive.

`ttMigrate` ignores this option when the `-c` or `-a` options are given.

If you specify both `-estimateStats` and `-updateStats`, statistics on restored tables are updated, not estimated.

Use of this flag may improve the performance of materialized view restoration and may also improve the performance of queries on the restored tables and views.

Converts all `NaN`, `Inf` and `-Inf` values found in migrate objects to 0.0. This is useful for migrating data into releases of TimesTen that do not support the `NaN`, `Inf` and `-Inf` values.

Prints a usage message and exits.
### Option Description

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-inline rule</td>
<td>Indicates the rule to be used for converting variable-length columns to <strong>INLINE</strong> in restore mode. The value for rule is one of:</td>
</tr>
<tr>
<td></td>
<td>preserve - <code>ttMigrate</code> preserves the original <strong>INLINE</strong> attribute of each column. This is the default, and it is required if</td>
</tr>
<tr>
<td></td>
<td>-exactUpgrade is used.</td>
</tr>
<tr>
<td></td>
<td>dsDefault - <code>ttMigrate</code> uses the database's default rule for setting the <strong>INLINE</strong> attribute of restored columns.</td>
</tr>
<tr>
<td></td>
<td>maxlen - <code>ttMigrate</code> restores as <strong>INLINE</strong> all variable-length columns with length &lt;= <code>maxlen</code> and restores as <strong>NOT INLINE</strong> all</td>
</tr>
<tr>
<td></td>
<td>variable-length columns with length greater than <code>maxlen</code>. If <code>maxlen</code> is 0 then all columns are restored as <strong>NOT INLINE</strong>.</td>
</tr>
<tr>
<td></td>
<td><strong>INLINE</strong> variable-length columns cannot successfully be replicated to <strong>NOT INLINE</strong> columns.</td>
</tr>
<tr>
<td>-l</td>
<td>Selects List mode. Lists the names of database objects in the specified data file. See &quot;List mode (-l) and Long-list mode (-L)&quot; on page 5-105 for more details.</td>
</tr>
<tr>
<td>-L</td>
<td>Selects Long-list mode. Lists the names of database objects in the specified data file and other details about the database objects. See &quot;List mode (-l) and Long-list mode (-L)&quot; on page 5-105 for more details.</td>
</tr>
<tr>
<td>-r</td>
<td>Selects Restore mode. Restores a database from a binary data file created by this utility. See &quot;Restore mode (-r)&quot; on page 5-104 for more details.</td>
</tr>
<tr>
<td>name</td>
<td>The name of the database object(s) to be saved or restored.</td>
</tr>
<tr>
<td>-nf</td>
<td>Specifies that <code>ttMigrate</code> should not save or restore foreign key information when saving or restoring ordinary (non-cached) tables.</td>
</tr>
<tr>
<td>-nr</td>
<td>Specifies that <code>ttMigrate</code> should not save or restore table rows when saving or restoring ordinary (non-cached) tables.</td>
</tr>
<tr>
<td>-noAutoCreateUsers</td>
<td>Specifies that <code>ttMigrate</code> should not create users.</td>
</tr>
<tr>
<td></td>
<td>By default, TimesTen creates “disabled” users when migrating tables from releases earlier than 11.2.1. TimesTen creates users but does not assign any privileges to these users. You must explicitly assign privileges, including CREATE SESSION, to these users after they are created.</td>
</tr>
<tr>
<td>-relaxedUpgrade</td>
<td>Save or restore the tables in a way that is compatible with a replication scheme that uses <code>TABLE DEFINITION CHECKING RELAXED</code>.</td>
</tr>
<tr>
<td></td>
<td><code>ttMigrate</code> ignores this option when the -a option is given.</td>
</tr>
<tr>
<td></td>
<td>This option should not be used in combination with a replication scheme that uses <code>TABLE DEFINITION CHECKING EXACT</code>, or else replication may no longer work.</td>
</tr>
<tr>
<td></td>
<td>The default is -exactUpgrade.</td>
</tr>
<tr>
<td>owner</td>
<td>The owner of a migrate object.</td>
</tr>
</tbody>
</table>
### ttMigrate Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-exactUpgrade</code></td>
<td>Save or restore the tables in a way that is compatible with a replication scheme that uses TABLE DEFINITION CHECKING EXACT. ttMigrate ignores this option when the -c or -a options are given. This option should not be used in combination with a replication scheme that uses TABLE DEFINITION CHECKING RELAXED, or else replication may no longer work. INLINE variable-length columns cannot successfully be replicated to NOT INLINE columns. This is the default.</td>
</tr>
<tr>
<td><code>-saveAsCharset</code></td>
<td>Saves an object in the specified connection character set. ttMigrate returns an informational message if the connection character set is different from the database character set. If this option is not set, by default, ttMigrate saves the migrated object in the database character set.</td>
</tr>
<tr>
<td><code>-updateStats</code></td>
<td>Specifies that ttMigrate should update statistics on restored tables and materialized views. ttMigrate ignores this option when the -c or -a options are given. If you specify both <code>-estimateStats</code> and <code>-updateStats</code>, statistics on restored tables are updated, not estimated. Use of this flag may improve the performance of materialized view restoration and may also improve the performance of queries on the restored tables and views.</td>
</tr>
<tr>
<td><code>-v verbosity</code></td>
<td>Specifies the verbosity level for messages printed when ttMigrate saves or restores a database. One of: 0 - Shows errors and warnings only. 1 - Prints the name of each table as it is saved or restored. 2 - Prints the name of each table or index as it is saved or restored. 3 (default) - Prints the name of each table or index as it is saved or restored and prints a dot (.) for each 10,000 rows saved or restored. ttMigrate ignores the <code>-v</code> option in List, Long-list, Describe and Long-describe modes.</td>
</tr>
<tr>
<td>`-V</td>
<td>-version`</td>
</tr>
</tbody>
</table>

The following ttMigrate options are available in restore mode (`-r`) only:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-autorefreshPaused</code></td>
<td>Restores cache groups with AUTOREFRESH attribute with autorefresh state paused. Otherwise the state is set to OFF.</td>
</tr>
<tr>
<td><code>-C ckptFreq</code></td>
<td>Specifies that ttMigrate should checkpoint the database after restoring every <code>ckptFreq</code> megabytes of data. A value of zero (the default) specifies that ttMigrate should never checkpoint the database. NOTE: This option is not supported in TimesTen Scaleout.</td>
</tr>
</tbody>
</table>
Create mode (-c) and Append mode (-a)

In create mode, ttMigrate saves migrate objects from a TimesTen database into a new binary data file. If the data file does not exist, ttMigrate creates it. Otherwise, ttMigrate overwrites the existing file, destroying its contents.

The data file format used by ttMigrate is independent of any release of TimesTen, so it is possible to use ttMigrate to migrate data from one TimesTen release to another.

In Append mode, ttMigrate appends migrate objects from a TimesTen database to an existing data file. If the data file does not exist, ttMigrate creates it.

For each ordinary (non-cached) table, ttMigrate saves:

- The table description: the name and type of each of the table’s columns, including primary key and nullability information.
- The table’s index definitions: the name of each index and the columns contained in the index. The actual contents of the index are not saved; ttMigrate only saves the information needed to rebuild the index when the table is restored.
- The table’s foreign key definitions. You can disable the saving of foreign key definitions using the -nf option.
- The rows of the table. You can disable the saving of rows using the -nr option.

For each cache group, ttMigrate saves the following:

- The convertCGTypes option determines the best type mapping from the underlying Oracle database tables to TimesTen cached tables using:
  - The types of the columns in the Oracle database tables.
  - The types of the columns stored in the migration file.
  - The TimesTen-to-Oracle type mapping rules.

- The localhost hostName option explicitly identifies the name or IP address of the local host when restoring replicated tables.

- The -noCharSetConversion option restores data, retaining the connection character set that is stored in the data file. ttMigrate does not convert the connection character set to match the database character set.
  
  If not set, ttMigrate restores the data and converts the connection character set to be the same as the database character set.
  
  See also: -saveAsCharset.

  This option may be useful for legacy TimesTen users who may have migrated pre-18.1 data into a 18.1 or later release of TimesTen as WE8ISO8859P1, when the data is actually in another character set. If, at a later time you want to have that data interpreted according to its actual character set, use this option to migrate the data into a database that uses the data’s actual character set with no character set conversion.

- The -restorePublicPrivs option restores privileges that were granted to PUBLIC after the database was created. By default, the ttMigrate utility does not restore privileges granted to PUBLIC. You must explicitly specify this option to restore privileges to PUBLIC.
The cache group definition: the cache group owner and name, the names of all tables in the cache group and any relevant cache group settings, such as the cache group duration.

**Note:** After ttMigrate has been used to restore a database, all autorefresh cache groups in the restored database have AUTOREFRESH state set to OFF, no matter how it was set on the source database. After restoring a cache group with ttMigrate -r, reset its AUTOREFRESH STATE to ON by using the ALTER CACHE GROUP statement (this can be done programmatically or with the ttIsql utility).

All the cached tables in the cache group: the table name, column information, table attributes (propagate or read-only), WHERE clause, if any, foreign key definitions and index definitions.

For each view, ttMigrate saves the following:

- All the same information as a normal table.
- The query defining the view.

For each sequence, ttMigrate saves the following:

- The complete definition of the sequence.
- The sequence's current value.

For each user (except the instance administrator), ttMigrate saves the following:

- User name.
- The user's encrypted password.
- Privileges that have been granted to the user.

For PUBLIC, ttMigrate saves all privileges that have been granted to PUBLIC after database creation.

If there are any replication schemes defined, ttMigrate saves all of the TTREP tables containing the replication schemes. Replication schemes should have names that are unique from all other database objects. It is not possible to migrate a replication scheme with the same name as any other database object.

**Note:** The ttMigrate utility does not save the rows of a cached table into the data file, even if you have not specified the -nr option. The foreign key definitions of cached tables are always saved, regardless of the use of the -nf option, as they are needed to maintain the integrity of the cache group.

By default, ttMigrate saves all database objects and users in the database to the data file, including tables, views, cache groups, sequences, users and replication schemes. Alternatively, you can give a list of database objects to be saved on the command line, except for replication schemes. The names in this list can contain the wildcard characters % (which matches one or more characters) and _ (which matches a single character). ttMigrate saves all database objects that match any of the given patterns. You do not need to be fully qualify names: If a name is given with no owner, ttMigrate saves all database objects that match the specified name or pattern, regardless of their owners.
You cannot save cached tables independently of their cache groups. If you list a cached table on the command line without also listing the corresponding cache group ttMigrate issues an error.

Use the -v option to control the information that ttMigrate prints while the save is in progress.

**Restore mode (-r)**

In Restore mode, ttMigrate restores all database objects from a data file into a TimesTen database.

For each ordinary (non-cached) table, ttMigrate restores:

- The table, using the original owner, table name, column names, types and nullability and the original primary key.
- The table's foreign keys. You can use the -nf flag to disable the restoration of foreign keys.
- All indexes on the table.
- All rows of the table. You can use the -nr flag to disable the restoration of rows.

For each cache group, ttMigrate restores:

- The cache group definition, using the original cache group owner and name.
- Each cached table in the cache group, using the original table names, column names, types and nullability, the original primary key, the table attributes (PROPAGATE or READONLY), and the WHERE clause, if any.
- The foreign key definitions of the cached tables.
- All the indexes on the cached tables.

**Note:** The ttMigrate utility does not restore the rows of cached tables, even if you have not specified the -nr option. The foreign key definitions of the cached tables are always restored, regardless of the use of the -nf option, as they are needed to maintain the integrity of the cache group.

By default, the -exactUpgrade option is set during restore.

By default, ttMigrate restores all tables and cache groups in the data file. Alternatively, you can list specific tables and cache groups to be restored on the command line. The names in this list must be fully qualified and cannot use wildcard characters.

You cannot restore cached tables independently of their cache groups. If you list a cached table on the command line without also listing the corresponding cache group, then ttMigrate issues an error.

Use the -v option to control the information that ttMigrate prints while the restoration is in progress.

The -inline option may be used to control whether variable length columns are restored as INLINE or NOT INLINE. See “Type specifications” in *Oracle TimesTen In-Memory Database SQL Reference*. In the default mode, -inlinepreserve, ttMigrate restores all variable-length columns with the same INLINE or NOT INLINE setting with which they were saved. In the other two modes, -inlinedefault and -inlinemaxlen, ttMigrate restores variable-length columns equal to or shorter than a
threshold length as **INLINE**, and restores all other variable length columns as **NOT INLINE**. For `-inlinedsDefault`, this threshold is the default automatic **INLINE** length for a TimesTen database. The `-inlinemaxlen` mode restores variable length columns with a user-specified threshold length of `maxlen` as **INLINE**, and all other variable length columns as **NOT INLINE**, even if they were saved as **INLINE**. If `maxlen` is 0, then all variable-length columns are restored as **NOT INLINE**.

**List mode (-l) and Long-list mode (-L)**

In List mode, `ttMigrate` lists the names of database objects in the specified data file, including cached tables and the replication scheme **TTREP** tables.

In Long-list mode, `ttMigrate` lists the names of database objects in the data file, including cached tables and the replication scheme **TTREP** tables, along with the number of rows in each table and the index definitions for each table, the query defining each view and the specifications for each sequence.

By default, `ttMigrate` lists the replication scheme name and all the database objects in the file. Alternatively you can provide a list of names of database objects on the command line. The names in this list must be fully qualified and cannot use wildcard characters.

**Describe mode (-d)**

In Describe mode, `ttMigrate` gives a short description for database objects in the specified file.

For each table, `ttMigrate` lists the table name, the number of rows in the table, and the table's column definitions, primary key and foreign keys. For cached tables, `ttMigrate` also lists the table attributes (**PROPAGATE** or **READONLY**) and the table's **WHERE** clause, if any.

For views, `ttMigrate` also lists the query defining the view.

For cache groups, `ttMigrate` lists the cache group name, the number of tables in the cache group, the cache group duration and describes each cached table in the cache group.

For replication schemes, `ttMigrate` lists the replication scheme name and all the **TTREP** replication scheme tables in the same manner as user tables.

By default, `ttMigrate` describes all the database objects in the file. Alternatively, you can provide a list of names of database objects on the command line. The names in this list must be fully qualified and cannot use wildcard characters.

**Long-describe mode (-D)**

In Long-describe mode, `ttMigrate` gives a full description for database objects in the specified file.

For each table, `ttMigrate` lists the table's name and the number of rows in the table, the table's column definitions, primary key, foreign keys and index definitions. For cached tables, `ttMigrate` also lists the table attributes (**PROPAGATE** or **READONLY**) and the table's **WHERE** clause, if any.

For cache groups, `ttMigrate` lists the cache group name, the number of tables in the cache group, the cache group duration and describes each cached table in the cache group.

For sequences, `ttMigrate` lists all the values used to define the sequence and its current value.
For replication schemes, ttMigrate lists all the TTREP replication scheme tables in the same manner as user tables.

By default, ttMigrate describes all of database objects in the file. Alternatively, you can provide a list of names of database objects on the command line. The names in this list must be fully qualified and cannot use wildcard characters.

Cache group data type conversions
When restoring a database that contains cache groups from a TimesTen release that is earlier than 7.0, use the -convertCGTypes option to convert the data type of columns from pre-7.0 types to more clearly map with the data types of the columns in the Oracle database with which the cache group is associated.

The following table describes the type mapping.

<table>
<thead>
<tr>
<th>Pre-7.0 TimesTen Type</th>
<th>Oracle Type</th>
<th>Converted Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>TINYINT</td>
<td>NUMBER(p,s) when s &gt; 0</td>
<td>NUMBER(p,s)</td>
</tr>
<tr>
<td>TINYINT</td>
<td>NUMBER(p,s) when s &lt;= 0</td>
<td>TT_TINYINT</td>
</tr>
<tr>
<td>SMALLINT</td>
<td>NUMBER(p,s) when s &gt; 0</td>
<td>NUMBER(p,s)</td>
</tr>
<tr>
<td>SMALLINT</td>
<td>NUMBER(p,s) when s &lt;= 0</td>
<td>TT_SMALLINT</td>
</tr>
<tr>
<td>INTEGER</td>
<td>NUMBER(p,s) when s &gt; 0</td>
<td>NUMBER(p,s)</td>
</tr>
<tr>
<td>INTEGER</td>
<td>NUMBER(p,s) when s &lt;= 0</td>
<td>TT_INTEGER</td>
</tr>
<tr>
<td>BIGINT</td>
<td>NUMBER(p,s) when s &gt; 0</td>
<td>NUMBER(p,s)</td>
</tr>
<tr>
<td>BIGINT</td>
<td>NUMBER(p,s) when s &lt;= 0</td>
<td>TT_BIGINT</td>
</tr>
<tr>
<td>NUMERIC(p,s) DECIMAL(p,s)</td>
<td>NUMBER</td>
<td>NUMBER</td>
</tr>
<tr>
<td>NUMERIC(p,s) DECIMAL(p,s)</td>
<td>NUMBER(x,y)</td>
<td>NUMBER(x,y)</td>
</tr>
<tr>
<td>NUMERIC(p,s) DECIMAL(p,s)</td>
<td>FLOAT(x)</td>
<td>NUMBER(p,s)</td>
</tr>
<tr>
<td>REAL</td>
<td>Any</td>
<td>BINARY_FLOAT</td>
</tr>
<tr>
<td>DOUBLE</td>
<td>Any</td>
<td>BINARY_DOUBLE</td>
</tr>
<tr>
<td>FLOAT(x) x &lt;= 24</td>
<td>Any</td>
<td>BINARY_FLOAT</td>
</tr>
<tr>
<td>FLOAT(x) x &gt;= 24</td>
<td>Any</td>
<td>BINARY_DOUBLE</td>
</tr>
<tr>
<td>CHAR(x)</td>
<td>Any</td>
<td>ORA_CHAR(x)</td>
</tr>
<tr>
<td>VARCHAR(x)</td>
<td>Any</td>
<td>ORAVARCHAR2(x)</td>
</tr>
<tr>
<td>BINARY(x)</td>
<td>Any</td>
<td>TT_BINARY(x)</td>
</tr>
<tr>
<td>VARBINARY(x)</td>
<td>Any</td>
<td>TT_VARBINARY(x)</td>
</tr>
<tr>
<td>DATE</td>
<td>DATE</td>
<td>ORA_DATE</td>
</tr>
<tr>
<td>TIMESTAMP</td>
<td>DATE</td>
<td>ORA_DATE</td>
</tr>
<tr>
<td>TIME</td>
<td>DATE</td>
<td>ORA_DATE</td>
</tr>
<tr>
<td>Any1</td>
<td>TIMESTAMP(m)</td>
<td>ORA_TIMESTAMP(m)</td>
</tr>
</tbody>
</table>

Note: Any means the type value does not affect the converted result type.
For information on data types, see “Data Types” in Oracle TimesTen In-Memory Database SQL Reference and “Mappings between Oracle Database and TimesTen data types” in Oracle TimesTen Application-Tier Database Cache User’s Guide.

Return codes

The ttMigrate utility restore (-r) and create (-c) commands return the following exit codes:

0 - All objects were successfully created or restored.
1 - Some objects successfully created or restored. Some objects could not be created or restored due to errors.
2 - Fatal error, for example, could not connect or could not open the data file.
3 - Ctrl-C or another signal received during the create or restore operation.

Examples

The following command dumps all database objects from database SalesDS into a file called sales.ttm. If sales.ttm exists, ttMigrate overwrites it.

```bash
ttMigrate -c SalesDS sales.ttm
```

This command appends all database objects in the SalesDS database owned by user MARY to sales.ttm:

```bash
ttMigrate -a SalesDS sales.ttm MARY.%
```

This command restores all database objects from sales.ttm into the SalesDS database:

```bash
ttMigrate -r SalesDS sales.ttm
```

This command restores MARY.PENDING and MARY.COMPLETED from sales.ttm into SalesDS (migrate objects are case-insensitive):

```bash
ttMigrate -r SalesDS sales.ttm MARY.PENDING MARY.COMPLETED
```

This command lists all migrate objects saved in sales.ttm:

```bash
ttMigrate -l sales.ttm
```

Notes

When migrating backward into a release of the Oracle TimesTen In-Memory Database that does not support features in the current release, TimesTen generally issues a warning and continues without migrating the unsupported features. In a few cases, where objects have undergone conversion, ttMigrate may fail and return an error message. This may be the case with conversions of data types, character sets and primary key representation.

The following restrictions, limitations and suggestions should be considered before preparing to use ttMigrate.

**Asynchronous materialized view:** When migrating to a previous release, asynchronous materialized views are ignored and TimesTen returns a warning.

**Cache groups:** In restore mode, the presence of foreign key dependencies between tables may require ttMigrate to reorder tables to ensure that a child table is not restored before a parent table.
Character columns in cached tables must have not only the same length but also the same byte semantics as the underlying Oracle database tables. Cache group migration fails when there is a mismatch in the length or length semantics of any of its cached tables.

The connection attribute PassThrough with a nonzero value is not supported with this utility and returns an error.

**Character sets:** By default, ttMigrate stores table data in the database character set, unless you have specified the -saveAsCharset option. At restore time, conversion to another character set can be achieved by migrating the table into a database that has a different database character set. When migrating data from a release of TimesTen that is earlier than 7.0, TimesTen assumes that the data is in the target database’s character set. If the data is not in the same database character set as the target database, the data may not be restored correctly.

When migrating columns with BYTE length semantics between two databases that both support NLS but with different database character sets, it is possible for migration to fail if the columns in the new database are not large enough to hold the values in the migrate file. This could happen, for example, if the source database uses a character set whose maximum byte-length is 4 and the destination database uses a character set whose maximum byte-length is 2.

TimesTen issues a warning whenever character set conversion takes place to alert you to the possibility of data loss due to conversion.

**Foreign key dependencies:** In restore mode, the presence of foreign key dependencies between tables may require ttMigrate to reorder tables to ensure that a child table is not restored before any of its parents. Such dependencies can also prevent a child table from being restored if any of its parent tables were not restored. For example, when restoring a table A that has a foreign key dependency on a table B, ttMigrate first checks to verify that table B exists in the database. If table B is not found, ttMigrate delays the restoration of table A until table B is restored. If table B is not restored as part of the ttMigrate session, TimesTen prints an error message indicating that table A could not be restored due to an unresolved dependency.

**Indexes:** TimesTen supports range indexes as primary-key indexes into TimesTen releases that support this feature. When migrating backward into a release that does not support range indexes as primary-key indexes, the primary keys are restored as hash indexes of the default size. When migrating forward from a release that does not support range indexes as primary-key indexes, the primary keys are restored as hash indexes of the same size as the original index.

**Materialized view logs:** TimesTen does not save the content of materialized view logs, only the definition.

**Replication:** Before attempting a full store migrate of replicated stores, ensure the host name and database name are the same for both the source and destination databases.

**System views:** TimesTen does not save the definitions or content of system views during migration.

**Other considerations:** Because ttMigrate uses a binary format, you cannot use ttMigrate to:

- Migrate databases between hardware platforms.
- Restore data saved with ttBackup or use ttBackup to restore data saved with ttMigrate.

**Platforms:** You can use ttMigrate together with ttMigrateCS (client server version of ttMigrate) to migrate databases between 32- and 64-bit platforms or bit levels. You
must use the `-relaxedUpgrade` option when restoring data on a new bit-level. In the case of changing bit-levels, the database cannot be involved in a replication scheme.

Follow the examples in "Moving a database between 32-bit and 64-bit platforms" in the *Oracle TimesTen In-Memory Database Installation, Migration, and Upgrade Guide*.

- On Windows, you can use `ttMigrate` to access databases from any release of TimesTen. On Windows, this utility is supported for all TimesTen Data Manager and Client DSNs.

- On UNIX and Linux systems, the release of `ttMigrate` must match the release of the database you are connecting to.

It is recommended that you do not run DDL SQL commands while running `ttMigrate` to avoid lock contention issues for your application.

**See also**

- `ttBackup`
- `ttBulkCp`
- `ttRestore`
ttRepAdmin

Description
Displays existing replication definitions and monitors replication status. The ttRepAdmin utility is also used when upgrading to a new release of TimesTen.

Required privilege
This utility requires the ADMIN privilege.

Usage with TimesTen Scaleout
This utility is not supported in TimesTen Scaleout.

Syntax

```
ttRepAdmin {-h | -help | -?}
ttRepAdmin {-V | -version}
ttRepAdmin -self -list [-scheme [owner.]schemeName]
  {DSN | -connStr connection_string}

.ttRepAdmin -receiver [-name receiverName]
  [-host receiverHostName] [-state receiverState] [-reset]
  [-list] [-scheme [owner.]schemeName]
  {DSN | -connStr connection_string}

.ttRepAdmin -log {DSN | -connStr connection_string}

.ttRepAdmin -showstatus {-awtmoninfo} {DSN | -connStr connection_string}

.ttRepAdmin -showconfig {DSN | -connStr connection_string}

.ttRepAdmin -bookmark {DSN | -connStr connection_string}

.ttRepAdmin -wait [-name receiverName] [-host receiverHostName]
  [-timeout seconds] {DSN | -connStr connection_string}

.ttRepAdmin -duplicate -from srcDataStoreName
  -host srcDataStoreHost
  [-localIP localIPAddress] [-remoteIP remoteIPAddress]
  [-setMasterRepStart] [-ramLoad] [-delXla]
  [-UID userId] [-PWD pwd | -PWDCrypt encryptedPwd]
  [-drop { [owner.]table ... | [owner.]sequence | ALL }]
  [-truncate { [owner.]table ... | ALL }]
  [-compression 0 | 1] [-bandwidthmax maxKbytesPerSec]
  [ (-activeDataGuard [-cacheUid cacheUid [-cachePdw cachePdw]]
    | -initCacheDr [-cacheUid cacheUid [-cachePdw cachePdw]]
    | -noDRTruncate) [-nThreads]
    | ( -keepCG [-cacheUid cacheUid [-cachePdw cachePdw]]
      | [-recoveringNode | -deferCacheUpdate] )) | -nokeepCG ) ]
  [-remoteDaemonPort portNo] [-verbosity {0|1|2}]
  [-localhost localHostName]
  {destDSN | -connStr connection_string}
```
ttRepAdmin operations

Use the ttRepAdmin utility for many replication operations. These operations fall into the following categories:

- Help and version information
- Database information
- Subscriber database operations
- Duplicate a database
- Wait for updates to complete
- Replication status
Help and version information

Use this form of `ttRepAdmin` to obtain help and the current version of TimesTen.

```
ttRepAdmin {-h | -help | -?}
ttRepadmin {-V | -version}
```

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h</td>
<td>Display help information.</td>
</tr>
<tr>
<td>-help</td>
<td></td>
</tr>
<tr>
<td>-?</td>
<td></td>
</tr>
<tr>
<td>-V</td>
<td>-version</td>
</tr>
</tbody>
</table>
Database information

Use this form of ttRepAdmin to obtain summary information about a database.

```
ttRepAdmin -self -list [-scheme [owner.]schemeName] (DSN | -connStr connection_string)
```

Options

ttRepAdmin -self -list has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSN</td>
<td>Data source name of a master or subscriber database.</td>
</tr>
<tr>
<td>-connStr connection_string</td>
<td>Connection string of a master or subscriber database, an ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.</td>
</tr>
<tr>
<td>-self</td>
<td>Specified database.</td>
</tr>
<tr>
<td>-list</td>
<td>Lists database name, host, port number, and bookmark position.</td>
</tr>
<tr>
<td>-scheme [owner.]schemeName</td>
<td>Name of replication scheme when there is more than one scheme.</td>
</tr>
</tbody>
</table>

Examples

```
ttRepAdmin -self -list my_dsn
```

The above syntax prints out information about the replication definition of the database `my_dsn`. 
Subscriber database operations

Use this form of ttRepAdmin to check the status or reset the state of a subscriber (receiver) database.

```
ttRepAdmin -receiver [-name receiverName] [-host receiverHostName] [-state receiverState] [-reset] [-list] [-scheme [owner.]schemeName] {DSN | -connStr connection_string}
```

Options

**ttRepAdmin -receiver** has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSN</td>
<td>Data source name of the master database.</td>
</tr>
<tr>
<td>-connStr connection_string</td>
<td>Connection string of the master database, an ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.</td>
</tr>
<tr>
<td>-receiver</td>
<td>Subscriber databases receiving updates from the master. Use -name and -host to specify a specific subscriber database.</td>
</tr>
<tr>
<td>-name receiverName</td>
<td>A specific subscriber (receiving) database. The receiverName is the last component in the database path name.</td>
</tr>
<tr>
<td>-host receiverHostName</td>
<td>Host name or TCP/IP address of the subscriber host.</td>
</tr>
<tr>
<td>-state start</td>
<td>Sets the state of replication for the subscriber.</td>
</tr>
<tr>
<td>-state stop</td>
<td>start (default) - Starts replication to the subscriber.</td>
</tr>
<tr>
<td>-state pause</td>
<td>stop - Stops replication to the subscriber, discarding updates.</td>
</tr>
<tr>
<td></td>
<td>pause - Pauses the replication agent, preserving updates.</td>
</tr>
<tr>
<td></td>
<td>See &quot;Set the replication state of subscribers&quot; in Oracle TimesTen In-Memory Database Replication Guide for more information.</td>
</tr>
<tr>
<td>-reset</td>
<td>Clears the bookmark in the master database log for the latest transaction to be sent to a given subscriber. This option should only be used when the transaction numbering of the master database is changed, such as when the database is re-created using ttMigrate or ttBackup. If the master database is saved and restored using ttBackup and ttRestore, transaction numbering is preserved and this option should not be used.</td>
</tr>
<tr>
<td>-list</td>
<td>Lists information about a replication definition.</td>
</tr>
<tr>
<td>-scheme [owner.]schemeName</td>
<td>Specifies the replication scheme name when there is more than one scheme.</td>
</tr>
</tbody>
</table>

Examples

```
ttRepAdmin -receiver -list my_dsn
```

The above syntax lists replication information for all the subscribers of the master database, *my_dsn*.

```
ttRepAdmin -receiver -name rep_dsn -list my_dsn
```

The above syntax lists replication information for the *rep_dsn* subscriber of the master database, *my_dsn*. 
ttRepAdmin -receiver -name rep_dsn -reset my_dsn

The above syntax resets the replication bookmark with respect to the `rep_dsn` subscriber of the master database. Should only be used when migrating a replicated database with `ttMigrate` or `ttBulkCp`.

`ttRepAdmin -receiver -name rep_dsn -state Start my_dsn`

The above syntax resets the replication state of the `rep_dsn` subscriber database to the `Start` state with respect to the master database, `my_dsn`. 
Duplicate a database

Use this form of ttRepAdmin to create a new database with the same contents as the master database.

The following must be true for you to perform the ttRepAdmin -duplicate:

- Only the instance administrator can run ttRepAdmin -duplicate.
- The instance administrator must have the same operating system username on both source and target computer to execute ttRepAdmin -duplicate.
- You must provide the user name and password with the -UID and -PWD options for an internal user with the ADMIN privilege on the source database.
- You must run ttRepAdmin on the target host.
- The DSN specified must be a direct-mode DSN, not a server DSN.

Before running the ttRepAdmin -duplicate command, use ttStatus to ensure the replication agent is started for the source database.

```
   ttRepAdmin -duplicate -from srcDataStoreName
   -host srcDataStoreHost
   [-localIP localIPAddress] [-remoteIP remoteIPAddress]
   [-setMasterRepStart] [-ramLoad] [-delXla]
   -UID userId (-PWD pwd | -PWDCrypt encryptedPwd)
   [-drop { [owner.]table ... | [owner.]sequence | ALL }]
   [-truncate { [owner.]table ... | ALL }]
   [-compression 0 | 1] [-bandwidthmax maxKbytesPerSec]
   [ ( -activeDataGuard [-cacheUid cacheUid [-cachePwd cachePwd]]
       | -initCacheDr [-cacheUid cacheUid [-cachePwd cachePwd]]
       [-noDRTruncate] [-nThreads]
       | ( -keepCG [-cacheUid cacheUid [-cachePwd cachePwd]]
           ( -recoveringNode | -deferCacheUpdate ) )]) | -nokeepCG )
   [-remoteDaemonPort portNo] [-verbosity {0|1|2}]
   [-localhost localHostName]
   {destDSN | -connStr connection_string}
```

**Options**

**ttRepAdmin -duplicate** has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-bandwidthmax maxKbytesPerSec</td>
<td>Specifies that the duplicate operation should not put more than maxKbytesPerSec KB of data per second onto the network. A value of 0 indicates that there should be no bandwidth limitation. The default is 0. The maximum is 9999999.</td>
</tr>
<tr>
<td>-compression 0</td>
<td>1</td>
</tr>
<tr>
<td>-connStr connection_string</td>
<td>Specifies the connection string of the destination database, an ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.</td>
</tr>
<tr>
<td>-delXla</td>
<td>Removes all the XLA bookmarks as part of the duplicate operation. Use this option if you do not want to copy the bookmarks to the duplicate database.</td>
</tr>
<tr>
<td>destDSN</td>
<td>Indicates the data source name of the destination database.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>-drop ({owner.}table...</td>
<td>Drops any tables or sequences that are copied as part of the -duplicate operation but which are not included in the replication scheme. ttRepAdmin ignores the option if the table is a cache group table.</td>
</tr>
<tr>
<td>-duplicate</td>
<td>Creates a duplicate of the specified database using replication to transmit the database contents across the network. See &quot;Duplicating a database&quot; in Oracle TimesTen In-Memory Database Replication Guide.</td>
</tr>
<tr>
<td>-from srcDataStoreName</td>
<td>Used with -duplicate to specify the name of the sender (or master) database. The srcDataStoreName is the last component in the database path name.</td>
</tr>
<tr>
<td>-host srcDataStoreHost</td>
<td>Defines the host name or TCP/IP address of the sender (or master) database.</td>
</tr>
<tr>
<td>-initCacheDr [-cacheUid</td>
<td>Initializes disaster recovery. Must provide the cache admin user id and password.</td>
</tr>
<tr>
<td>cacheUid -cachePwd cachePwd]</td>
<td></td>
</tr>
<tr>
<td>-keepCG</td>
<td>-noKeepCG specify whether tables in cache groups should be maintained as cache group tables or converted to regular tables in the target database. The default is -noKeepCG.</td>
</tr>
<tr>
<td>cacheUid</td>
<td>is the cache administration user ID.</td>
</tr>
<tr>
<td>cachePwd</td>
<td>is the password for the cache administrator user.</td>
</tr>
<tr>
<td>If no password is provided, ttRepAdmin prompts for a password.</td>
<td></td>
</tr>
<tr>
<td>( -keepCG [-cacheUid cacheUid -cachePwd cachePwd]</td>
<td></td>
</tr>
<tr>
<td>cacheUid</td>
<td>is the cache administration user ID.</td>
</tr>
<tr>
<td>cachePwd</td>
<td>is the password for the cache administrator user.</td>
</tr>
<tr>
<td>If no password is provided, ttRepAdmin prompts for a password.</td>
<td></td>
</tr>
<tr>
<td>-localhost hostName</td>
<td>Use with -duplicate and -setMasterRepStart to explicitly identify the name or IP address of the local host.</td>
</tr>
<tr>
<td>-localIP localIPAddress</td>
<td>Specifies the alias or IP (IPv4 or IPv6) address of the local network interface to be used. If not specified, ttRepAdmin chooses any compatible interface.</td>
</tr>
<tr>
<td>-noDRTruncate</td>
<td>Used with the -initCacheDr option, -noDRTruncate disables truncation of Oracle tables during the initial rollout process for the remote subscriber on the Disaster Recovery site. When -noDRTruncate is specified, TimesTen does not truncate the Oracle Database tables that correspond to the Asynchronous Writethrough cache group tables in an active standby pair replication scheme.</td>
</tr>
</tbody>
</table>
Duplicate a database

**Example 5–1  Duplicating a database**

On the source database, create a user and grant the `ADMIN` privilege to the user:

```sql
CREATE USER ttuser IDENTIFIED BY ttuser;
User created.

GRANT admin TO ttuser;
```
The instance administrator must have the same user name on both instances involved in the duplication. Logged in as the instance administrator, duplicate the `ds1` database on `server1` to the `ds2` database:

```
      ttRepAdmin -duplicate -from ds1 -host "server1"
          -UID ttuser -PWD ttuser
          -connStr "dsn=ds2;UID=ttuser;PWD=ttuser"
```

**Example 5–2  Duplicating a database with cache groups**

Use the `-keepCG` option to keep cache group tables when you duplicate a database. Specify the cache administration user ID and password with the `-cacheuid` and `-cachepwd` options. If you do not provide the cache administration user password, `ttRepAdmin` prompts for a password.

If the cache administration user ID is `orauser` and the password is `orapwd`, duplicate database `dsn1` on `host1`:

```
      ttRepAdmin -duplicate -from dsn1 -host host1 -uid ttuser -pwd ttuser
          -keepCG -cacheuid orauser -cachepwd orapwd "DSN=dsn2;UID=;PWD="
```

The `UID` and `PWD` for `dsn2` are specified as null values in the connection string so that the connection is made as the current operating system user, which is the instance administrator. Only the instance administrator can run `ttRepAdmin -duplicate`. If `dsn2` is configured with `PWDCrypt` instead of `PWD`, then the connection string should be "DSN=dsn2;UID=;PWDCrypt=".

**Example 5–3  Setting the replication state on the source database**

The `-setMasterRepStart` option causes the replication state in the `srcDataStoreName` database to be set to the `Start` state before it is copied across the network and then keeps the database in memory. It ensures that any updates made to the master after the duplicate operation has started are copied to the subscriber.

You can use the `-localhost` option to identify the local host by host name or IP address. These options ensure that all updates made after the duplicate operation are replicated from the remote database to the newly created or restored local database.

```
      ttRepAdmin -duplicate -from srcDataStoreName -host srcDataStoreHost
          -setMasterRepStart -ramLoad
          -UID timesten_user -PWD timesten_user
          -localhost localHostName
          [destDSN | -connStr connection_string ]
```

**Notes**

This utility can duplicate any temporary table definition in a database, but it does not replicate the contents of temporary tables.

You cannot use this utility to duplicate databases across major releases of TimesTen.
Wait for updates to complete

Use this form of ttRepAdmin to assure that all the updates in the log are replicated to all subscribers before call returns.

```
ttRepAdmin -wait [-name receiverName] [-host receiverHostName] [-timeout seconds] {DSN | -connStr connection_string}
```

Options

ttRepAdmin -wait has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSN</td>
<td>Indicates the data source name of the master database.</td>
</tr>
<tr>
<td>-connStr connection_string</td>
<td>Specifies the connection string of the master database, an ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.</td>
</tr>
<tr>
<td>-wait</td>
<td>Waits for replication to become current before continuing.</td>
</tr>
<tr>
<td>-name receiverName</td>
<td>Identifies the database. The database name is the last component in the database path name.</td>
</tr>
<tr>
<td>-host receiverHostName</td>
<td>Defines the host name or TCP/IP address of the subscriber host.</td>
</tr>
<tr>
<td>-timeout seconds</td>
<td>Specifies timeout value in seconds. ttRepAdmin returns within this amount of time, even if all updates to subscribers have not been completed.</td>
</tr>
</tbody>
</table>

Examples

```
ttRepAdmin -wait -name receiverName -host receiverHostName -timeout seconds -dsn DSN
```

The above syntax provides a way to ensure that all updates, committed at the time this program was invoked, have been transmitted to the subscriber, receiverName, and the subscriber has acknowledged that all those updates have been durably committed at the subscriber database. The timeout in seconds limits the wait.

**Note:** If ttRepAdmin -wait is invoked after all write transaction activity is quiesced at a store (there are no active transactions and no transactions have started), it may take 60 seconds or more before the subscriber sends the acknowledgment that all updates have been durably committed at the subscriber.

```
ttRepAdmin -wait -dsn DSN
```

In the above syntax, if no timeout and no subscriber name are specified, ttRepAdmin does not return until all updates committed at the time this program was invoked have been transmitted to all subscribers and all subscribers have acknowledged that all those updates have been durably committed at the subscriber database.
Replication status

Use this form of ttRepAdmin to check the size of the transaction log files, bookmark position, or replication configuration of a master database.

- `ttRepAdmin -log (DSN | -connStr connection_string)`
- `ttRepAdmin -showstatus (-awtmoninfo) (DSN | -connStr connection_string)`
- `ttRepAdmin -showconfig (DSN | -connStr connection_string)`
- `ttRepAdmin -bookmark (DSN | -connStr connection_string)`

Options

The ttRepAdmin monitor operations have the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>DSN</code></td>
<td>Indicates the data source name of the master database.</td>
</tr>
<tr>
<td><code>-awtmoninfo</code></td>
<td>If you have enabled monitoring for AWT cache groups by calling the AwtMonitorConfig procedure, you can display the monitoring results by using the this option. If AWT monitoring is enabled, ttrepadmin -awtmoninfo displays the output:</td>
</tr>
<tr>
<td></td>
<td>- TimesTen processing time: The total number of milliseconds spent in processing AWT transaction data since monitoring was enabled.</td>
</tr>
<tr>
<td></td>
<td>- Oracle bookmark time: The total number of milliseconds spent in managing AWT metadata on Oracle since monitoring was enabled.</td>
</tr>
<tr>
<td><code>-connStr connection_string</code></td>
<td>Specifies the connection string of the master database, an ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.</td>
</tr>
<tr>
<td><code>-log</code></td>
<td>Prints out number and size of transaction log files retained by replication to transmit updates to other databases.</td>
</tr>
<tr>
<td><code>-showconfig</code></td>
<td>Lists the entire replication configuration.</td>
</tr>
<tr>
<td></td>
<td>See “Show the configuration of replicated databases” in Oracle TimesTen In-Memory Database Replication Guide for more information.</td>
</tr>
<tr>
<td><code>-showstatus</code></td>
<td>Reports the current status of the specified replicated database.</td>
</tr>
<tr>
<td></td>
<td>See “Use ttRepAdmin to show replication status” in Oracle TimesTen In-Memory Database Replication Guide for more information.</td>
</tr>
<tr>
<td><code>-bookmark</code></td>
<td>Reports the latest marker record from where replication must read the log, the most recently created log sequence number, and the latest log sequence number whose record has been flushed to disk. Bookmarks are not supported if you have configured parallel replication. See “Show replicated log records” in Oracle TimesTen In-Memory Database Replication Guide for more information.</td>
</tr>
</tbody>
</table>

Result set

If AWT monitoring is enabled, this utility displays the following information in addition to other ttRepAdmin -showstatus output.
- TimesTen processing time: The total number of milliseconds spent in processing AWT transaction data since monitoring was enabled.
- Oracle bookmark time: The total number of milliseconds spent in managing AWT metadata on Oracle since monitoring was enabled.
- Oracle execute time: The total number of milliseconds spent in OCI preparation, binding and execution for AWT SQL operations since monitoring was enabled. This statistic includes network latency between TimesTen and the Oracle database.
- Oracle commit time: The total number of milliseconds spent in committing AWT updates on Oracle since monitoring was enabled. This statistic includes network latency between TimesTen and the Oracle database.
- Time since monitoring was started.
- Total number of TimesTen row operations: The total number of rows updated in AWT cache groups since monitoring was enabled.
- Total number of TimesTen transactions: The total number of transactions in AWT cache groups since monitoring was enabled.
- Total number of flushes to Oracle: The total number of times that TimesTen data has been sent to the Oracle database.

The output also includes the percentage of time spent on TimesTen processing, Oracle bookmark, Oracle execution and Oracle commits.

Examples

```
ttRepAdmin -log DSN
```

The above syntax reports the number of transaction log files that replication is retaining to transmit updates to other databases. The replication agent retains a transaction log file until all updates in that transaction log file have been successfully transferred to each subscriber database.

```
ttRepAdmin -showconfig DSN
```

The above syntax reports the entire replication configuration. It lists all the subscribers for the specified DSN, the names and details of the tables being replicated, and all the subscriptions.

```
ttRepAdmin -showstatus DSN
```

The above syntax reports the current state of the database for the specified DSN. The output includes the state of all the threads in the replication agents for the replicated databases, bookmark locations, port numbers, and communication protocols.

```
ttRepAdmin -bookmark DSN
```

The above syntax prints out the log sequence numbers of the earliest log record still needed by replication, the last log record written to disk, and the last log record generated.

```
ttRepAdmin -showstatus -awtmoninfo database1
```

[other -showstatus output]

```
...  
AWT Monitoring statistics
--------------------------
TimesTen processing time : 0.689000 millisecs (0.164307 %)
Oracle bookmark time : 3.229000 millisecs (0.770027 %)
```
Oracle execute time : 342.908000 millisecs (81.774043 %)
Oracle commit time : 72.450000 millisecs (17.277315 %)
Time since monitoring was started: 8528.641000 millisecs
Cache-connect Operational Stats:
Total Number of TimesTen row operations : 2
Total Number of TimesTen transactions : 2
Total Number of flushes to Oracle : 2

The above syntax and output shows the AWT monitoring status.

Notes

The ttRepAdmin utility is supported only for TimesTen Data Manager DSNs. It is not supported for TimesTen Client DSNs.

You must use the -scheme option when specifying more than one replication scheme, or when more than one scheme exists involving the specified database.

Using SQL configuration, you can create multiple replication schemes in the same database. If there is only one replication scheme, the ttRepAdmin utility automatically determines the scheme. If there is more than one scheme, you must use the ttRepAdmin -scheme option to specify which scheme to use.

When configuring replication for databases with the same name on different hosts, you can indicate which database you want to operate on by using -host. For example, if all the subscribers have the name DATA, you can set the replication state on host SW1 with:

```
ttRepAdmin -receiver -name DATA -host SW1 -state start DSN
```

See also

For a full description of TimesTen Replication, see Oracle TimesTen In-Memory Database Replication Guide.
For upgrade examples, see "Upgrades in TimesTen Classic" in Oracle TimesTen In-Memory Database Installation, Migration, and Upgrade Guide.
ttRestore

Description

Creates a database from a backup that has been created using the ttBackup utility. If the database exists, ttRestore does not overwrite it.

The attributes in the ttRestore connection string can contain any of the first connection or general connection attributes. It can also include the data store attribute LogDir. All other data store attributes are copied from the backup files. The LogDir attribute enables the restored database to be relocated.

The ttRestore action is somewhat more powerful than a first connect, as it can move the database. It is somewhat less powerful than creating a new database, as it cannot override the data store attributes, except for the LogDir attribute.

For an overview of the TimesTen backup and restore facility, see “Backup, Restore, and Migrate Data in TimesTen Classic” in the Oracle TimesTen In-Memory Database Installation, Migration, and Upgrade Guide.

Required privilege

This utility requires the instance administrator privilege.

Usage with TimesTen Scaleout

This utility is not supported in TimesTen Scaleout.

Syntax

```
ttRestore {-h | -help | -?} 
ttRestore {-V | -version} 
ttRestore [-fname filePrefix] [-noconn] -dir directory 
    {DSN | -connStr connection_string} 
ttRestore -i [-noconn] {DSN | -connStr connection_string} 
```

Options

ttRestore has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-connStr connection_string</td>
<td>An ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.</td>
</tr>
<tr>
<td>DSN</td>
<td>Specifies an ODBC data source name of the database to be administered.</td>
</tr>
<tr>
<td>-dir directory</td>
<td>Specifies the directory where the backup files are stored.</td>
</tr>
<tr>
<td>-fname filePrefix</td>
<td>Specifies the file prefix for the backup files in the backup directory. The backup files must have been stored in the backup directory with this prefix. The default value for this parameter is the file name portion of the DataStore parameter of the database's ODBC definition.</td>
</tr>
<tr>
<td>-h</td>
<td>Prints a usage message and exits.</td>
</tr>
<tr>
<td>-help</td>
<td></td>
</tr>
<tr>
<td>-?</td>
<td></td>
</tr>
</tbody>
</table>
Examples

```
ttRestore -dir /users/pat/TimesTen/backups
-fname FastInsBkup FastIns
```

To back up a database named `origDSN` to the directory `/users/rob/tmp` and restore it to database named `restoredDSN`, use:

```
ttBackup -dir /users/rob/tmp -fname restored origDSN
ttRestore -dir /users/rob/tmp -fname restored restoredDSN
```

The value of `fname` is the name that you want for the prefix portion of the backup file name.

On UNIX and Linux systems, to restore a tape backup to the `FastIns` database, use:

```
dd bs=64k if=/dev/rmt0 | ttRestore -i FastIns
```

Notes

The `ttBackup` utility and the `ttRestore` utility back up and restore databases only when the first two parts of the TimesTen release and the platform are the same. For example, you can back up and restore files between release 18.1.1.1.0 and release 18.1.1.2.0 or 18.1.2.1.0. You cannot back up and restore files between release 11.2.2.8.0 and release 18.1.1.1.0, or between release 18.1.2.1.0 and release 18.2.1.1.0. You can use the `ttBulkcp` or `ttMigrateCS` (UNIX and Linux systems only) utility to migrate databases across major releases or operating systems. You can use `ttMigrate` together with `ttMigrateCS` (client server version of `ttMigrate`) to migrate databases between 32- and 64-bit platforms or bit levels. You must use the `-relaxedUpgrade` option when restoring data on a new bit-level. In the case of changing bit-levels, the database cannot be involved in a replication scheme.

Follow the examples in "Moving a database between 32-bit and 64-bit platforms" in the Oracle TimesTen In-Memory Database Installation, Migration, and Upgrade Guide.

You can backup databases containing cache groups with the `ttBackup` utility. However, when restoring such a backup, special consideration is required as the restored data within the cache groups may be out of date or out of sync with the data in the back end Oracle database. See the section on "Backing up and restoring a database with cache groups" in the Oracle TimesTen Application-Tier Database Cache User’s Guide for details.

See also

- `ttBackup`
- `ttBulkCp`
- `ttMigrate`
ttSchema

Description

Prints out the schema, or selected objects, of a database. The utility can list the following schema objects that are found in SQL CREATE statements:

- Tables
- Indexes
- Cache group definitions
- Sequences
- Views
- Materialized view logs
- Column definitions, including partition information
- PL/SQL program units

The level of detail in the listing and the objects listed are controlled by options. The output represents a point in time snapshot of the state of a database rather than a history of how the database came to arrive at its current state, perhaps through ALTER statements. An entire database, including data, cannot be completely reconstructed from the output of ttSchema. The ttIsql utility can play back the output of ttSchema utility to rebuild the full schema of a database.

On UNIX and Linux systems, this utility is supported for TimesTen Data Manager DSNs. For TimesTen Client DSNs, use the utility ttSchemaCS.

Required privilege

This utility requires no privileges beyond those needed to perform describe operations on database objects.

This utility prints information only about the objects owned by the user executing the utility, and those objects for which the owner has SELECT privileges. If the owner executing the utility has ADMIN privilege, ttSchema prints information about all objects.

Usage with TimesTen Scaleout

This utility is supported in TimesTen Scaleout.

Syntax

ttSchema {-h | -help | -?}

ttSchema {-V | -version}

ttSchema [-l] [-c] [-fixedTypes] [-st | -systemTables]
[-list {all | tables | views | sequences |
cachegroups | repschemes | synonyms | plsql} [,...] ]
[-plsqlAttrs | -noplsqlAttrs]
[-plsqlCreate | -[no]plsqlCreateOrReplace]
{-connStr connection_string | DSN }
[[@owner.]object_name][...]

Options

ttSchema has the options:
**Option** | **Description**  
--- | ---  
`-connStr connection_string` | An ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.  
`-c` | Compatibility mode. Limits the use of TimesTen-specific and release-specific keywords and extensions. This may be useful if the `ttSchema` output is being used as input to an older TimesTen release, or to some other database system, such as the Oracle database. The `-c` option prevents the `INLINE` and `NOT INLINE` keywords from being output.  
**DSN** | Specifies an ODBC data source name of the database from which to get a schema.  
`-fixedTypes` | Uses fully qualified data type names.  
`-h` | Prints a usage message and exits.  
`-help` |  
`-?` |  
`-l` | One per-line listing of objects in the database.  
`-list {all | tables | views | sequences | cachegroups | repschemes | synonyms | plsql}[,....]` | A comma-delimited (no space after comma) list of objects to generate. Lists only those types of objects specified. Default is `-list all`.  
`-list views` also displays information about materialized view logs.  
`[owner.]object_name` | Limits the scope of the output to specified database object(s).  
`-plsqlAttrs | -noplsqlAttrs` | Controls whether `ttSchema` emits `ALTER SESSION` statements with `CREATE` statements for PL/SQL program units.  
If `-plsqlAttrs` is specified, `ttSchema` emits `ALTER SESSION` statements to set these attributes before emitting a `CREATE` statement. This output from `ttSchema` can be fed back into `ttIsql` (or `sqlplus`) to create the same procedures, with the same compiler options as were specified in the original database (default).  
If `-noplsqlAttrs` is specified, only the `CREATE` statement is generated.  
`-plsqlCreate | -[no]plsqlCreateOrReplace` | If `-plsqlCreate` is specified, `ttSchema` emits `CREATE PROCEDURE, CREATE PACKAGE or CREATE FUNCTION` statements for PL/SQL program units.  
If `-plsqlCreateOrReplace` (default) is specified, `ttSchema` emits `CREATE or REPLACE` statements.  
`-st | -systemTables` | Include system tables. System tables are omitted by default.  
`-V | -version` | Prints the release number of `ttSchema` and exits.  

**Examples**

Objects in the `orddsn` database are created with these SQL statements:

```sql
CREATE TABLE ttuser.customer (
    cust_num INTEGER NOT NULL PRIMARY KEY,
    region CHAR(2) NOT NULL,
    name VARCHAR2(80),
    address VARCHAR2(255) NOT NULL);
```
CREATE SEQUENCE ttuser.custid MINVALUE 1 MAXVALUE 1000000;

CREATE TABLE ttuser.orders {
  ord_num INTEGER NOT NULL PRIMARY KEY,
  cust_num INTEGER NOT NULL,
  when_placed TIMESTAMP NOT NULL,
  when_shipped TIMESTAMP,
  FOREIGN KEY(cust_num) REFERENCES ttuser.customer (cust_num));

CREATE MATERIALIZED VIEW ttuser.order_summary AS
  SELECT cust.name, ord.ord_num, count(*) ord_count
  FROM ttuser.orders ord, ttuser.customer cust
  WHERE ord.cust_num = cust.cust_num
  GROUP BY cust.name, ord.ord_num;

Example 5–4 ttSchema for the database
Return the schema for the orderdsn database.

% ttSchema orderdsn
-- Database is in Oracle type mode
create table TTUSER.CUSTOMER {
  CUST_NUM NUMBER(38) NOT NULL,
  REGION   CHAR(2 BYTE) NOT NULL,
  "NAME"   VARCHAR2(80 BYTE) INLINE NOT NULL,
  ADDRESS  VARCHAR2(255 BYTE) NOT INLINE NOT NULL,
  primary key (CUST_NUM));

create table TTUSER.ORDERS {
  ORD_NUM      NUMBER(38) NOT NULL,
  CUST_NUM     NUMBER(38) NOT NULL,
  WHEN_PLACED  TIMESTAMP(6) NOT NULL,
  WHEN_SHIPPED TIMESTAMP(6),
  primary key (ORD_NUM),
  foreign key (CUST_NUM) references TTUSER.CUSTOMER (CUST_NUM));

create sequence TTUSER.CUSTID
  increment by 1
  minvalue 1
  maxvalue 1000000
  start with 1
  cache 20;

create materialized view TTUSER.ORDER_SUMMARY as
  SELECT CUST.NAME 'NAME', ORD.ORD_NUM 'ORD_NUM', COUNT(*) 'ORD_COUNT'
  FROM TTUSER.ORDERS ORD, TTUSER.CUSTOMER CUST WHERE ORD.CUST_NUM =
  CUST.CUST_NUM GROUP BY CUST.NAME, ORD.ORD_NUM ;

Example 5–5 Listing specific objects
Return only the materialized views and sequences for the orderdsn database.

% ttSchema -list views,sequences orderdsn
-- Database is in Oracle type mode
create sequence TTUSER.CUSTID
  increment by 1
  minvalue 1
  maxvalue 1000000
  start with 1
  cache 20;
create materialized view TTUSER.ORDER_SUMMERY as
  SELECT CUST.NAME "NAME", ORD.ORD_NUM "ORD_NUM", COUNT(*) "ORD_COUNT"
  FROM TTUSER.ORDERS ORD, TTUSER.CUSTOMER CUST
  WHERE ORD.CUST_NUM = CUST.CUST_NUM
  GROUP BY CUST.NAME, ORD.ORD_NUM ;

Example 5–6  Specifying an object

Return the schema information for the orders table in the orderdsn database.

% ttSchema orderdsn ttuser.orders
-- Database is in Oracle type mode
Warning: tables may not be printed in an order that can satisfy foreign key
reference constraints
create table TTUSER.ORDERS {
  ORD_NUM      NUMBER(38) NOT NULL,
  CUST_NUM     NUMBER(38) NOT NULL,
  WHEN_PLACED  TIMESTAMP(6) NOT NULL,
  WHEN_SHIPPED ORA_TIMESTAMP(6) NOT NULL,
  primary key (ORD_NUM),
  foreign key (CUST_NUM) references TTUSER.CUSTOMER (CUST_NUM));

Example 5–7  Specifying fixed data types

Return the schema information for the orderdsn database, using fixed data type
names.

% ttSchema -fixedTypes orderdsn
-- Database is in Oracle type mode
create table TTUSER.CUSTOMER {
  CUST_NUM NUMBER(38) NOT NULL,
  REGION   ORA_CHAR(2 BYTE) NOT NULL,
  "NAME"   ORA_VARCHAR2(80 BYTE) INLINE NOT NULL,
  ADDRESS  ORA_VARCHAR2(255 BYTE) NOT INLINE NOT NULL,
  primary key (CUST_NUM));
create table TTUSER.ORDERS {
  ORD_NUM      NUMBER(38) NOT NULL,
  CUST_NUM     NUMBER(38) NOT NULL,
  WHEN_PLACED  ORA_TIMESTAMP(6) NOT NULL,
  WHEN_SHIPPED ORA_TIMESTAMP(6),
  primary key (ORD_NUM),
  foreign key (CUST_NUM) references TTUSER.CUSTOMER (CUST_NUM));
create sequence TTUSER.CUSTID
  increment by 1
  minvalue 1
  maxvalue 1000000
  start with 1
  cache 20;
create materialized view TTUSER.ORDER_SUMMERY as
  SELECT CUST.NAME "NAME", ORD.ORD_NUM "ORD_NUM",
  COUNT(*) "ORD_COUNT" FROM TTUSER.ORDERS ORD, TTUSER.CUSTOMER CUST
  WHERE ORD.CUST_NUM = CUST.CUST_NUM
  GROUP BY CUST.NAME, ORD.ORD_NUM ;

Notes

The SQL generated does not produce a history of transformations through ALTER
statements, nor does it preserve table partitions, although the output gives information
on table partitions in the form of SQL comments. The ttSchema utility prints out the
partition numbers for the columns that are not in the initial partition. The initial partition is 0, so partition 1 as printed by ttSchema is secondary partition 1, not the initial partition. For more details on partitions, see "Understanding partitions when using ALTE TABLE," in the "ALTER TABLE" section of the Oracle TimesTen In-Memory Database SQL Reference.

The connection attribute PassThrough with a nonzero value is not supported with this utility and returns an error.

Output is not guaranteed to be compatible with DDL recognized by previous releases of TimesTen.

It is recommended that you do not run DDL SQL commands while running ttSchema to avoid lock contention issues for your application.
ttSize

**Description**

Estimates the amount of space that a given table, including any views in the database will consume when the table grows to include rows. You can use this utility on existing tables or to estimate table sizes when creating tables. If you do not specify an owner, ttSize prints size information for all tables of the given table name. The size information includes space occupied by any indexes defined on the table.

The memory required for varying-length columns is estimated by using the average length of the columns in the current table as the average length of the columns in the final table. If there are no rows in the current table, then ttSize assumes that the average column length is one half the maximum column length.

The memory required for LOB columns is estimated by using the average length of the columns in the current table as the average length of the columns in the final table. When no rows are being inserted into the table, computations do not include LOB columns.

The table is scanned when this utility is called. Avoid the scan of the table by specifying an optional non-NULL frac value, which should be between 0 and 1. The ttSize utility uses this value to estimate the average size of varying-length columns. The maximum size of each varying-length column is multiplied by the frac value to compute the estimated average size of VARBINARY or VARCHAR columns. If the frac option is not specified, the existing rows in the table are scanned and the average length of the varying-length columns in the existing rows is used. If frac is not specified and the table has no rows in it, then frac is assumed to have the value 0.5.

**Required privilege**

This utility requires no privileges beyond those needed to perform select operations on the specified database objects.

**Usage with TimesTen Scaleout**

This utility is supported in TimesTen Scaleout.

**Syntax**

```
ttSize {-h | -help | -?}
ttSize {-V | -version}
ttSize -tbl [owner.]tableName [-rows rows] [- frac fraction]
   {-connStr connection_string | DSN}
```

**Options**

**ttSize** has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-connStr connection_string</td>
<td>An ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.</td>
</tr>
<tr>
<td><strong>DSN</strong></td>
<td>Specifies the name of a data source to which ttSize should connect to retrieve table information.</td>
</tr>
</tbody>
</table>
Examples

To estimate the space required for a table, create the table in TimesTen, populate it with a sample of representative rows, create desired indexes and execute ttSize with those definitions. For example, to estimate the size of the NAMEID table in the data source FixedDs when it grows to 200,000 rows, execute:

```
ttSize -tbl Nameid -rows 200000 FixedDs
```

Rows = 200000

Total in-line row bytes = 7139428

Total = 7139428

Notes

LOB columns are treated similar to var-type columns, unless there are no rows being inserted into the table. The average size computation does not include LOB columns in such cases.

The columns PERM_ALLOCATED_SIZE and PERM_IN_USE_SIZE show the currently allocated size of the database (in KB units) and the in-use size of the database. The system updates this information each time a connection is made or released and each time a transaction is committed or rolled back.

This utility is supported only for TimesTen Data Manager DSNs. It is not supported for TimesTen Client DSNs.
The \texttt{ttStats} utility is supported in TimesTen Classic and TimesTen Scaleout, but supports different syntax and options. The following sections describe the \texttt{ttStats} utility for:

- \textbf{TimesTen Classic}
- \textbf{TimesTen Scaleout}

Table \ref{table:ttStats} shows the main differences of how the \texttt{ttStats} utility works in TimesTen Classic and TimesTen Scaleout.

\begin{table}[h]
\centering
\begin{tabular}{|l|l|l|}
\hline
\textbf{Action} & \textbf{TimesTen Classic} & \textbf{TimesTen Scaleout} \\
\hline
Create a snapshot & Execute the \texttt{ttStats} \linebreak -snapshot utility to create a snapshot. The \texttt{-description} option for \texttt{-snapshot} is optional & The \texttt{ttStats} daemon creates snapshots based on the \linebreak \texttt{pollSec} parameter of the \texttt{ttStatsConfig} built-in \linebreak procedure. \linebreak If you execute \texttt{ttStats} \linebreak \texttt{-snapshot -description} \linebreak \texttt{description} in TimesTen \linebreak Scaleout, you can associate a \linebreak description to the latest system \linebreak generated snapshot. \\
\hline
Drop a snapshot & Execute the \texttt{ttStats} \linebreak \texttt{-drop -begin_snap} \linebreak \texttt{snapid1} \linebreak \texttt{-end_snap} \linebreak \texttt{snapid2} utility to drop snapshots. & The \texttt{ttStats} daemon \linebreak automatically drops snapshots \linebreak based on the \texttt{retentionDays} \linebreak parameter of the \texttt{ttStatsConfig} \linebreak built-in \linebreak procedure. \linebreak The \texttt{-drop} option of the \texttt{ttStats} utility is not \linebreak supported in TimesTen \linebreak Scaleout. \\
\hline
View information about a snapshot & Execute the \texttt{ttStats} \linebreak \texttt{-report} \linebreak utility without any arguments. \linebreak The \texttt{ttStats} utility prints the \linebreak snapshot ID, date, time, \linebreak capture level, and the \linebreak description of all snapshots. & Execute the \texttt{ttStats} \linebreak \texttt{-snapshotInfo} utility. The \texttt{ttStats} \linebreak utility prints the \linebreak snapshot ID, date, time, and \linebreak the description of all snapshots. \\
\hline
Generate a report between two snapshots & Execute the \texttt{ttStats} \linebreak \texttt{-report} \linebreak \texttt{-snap1} \linebreak \texttt{snapid1} \linebreak \texttt{-snap2} \linebreak \texttt{snapid2} utility. The utility \linebreak generates a report between two \linebreak snapshot IDs. & Execute the \texttt{ttStats} \linebreak \texttt{-report} \linebreak \texttt{-snap1} \linebreak \texttt{snapid1} \linebreak \texttt{-snap2} \linebreak \texttt{snapid2} utility or the \texttt{ttStats} \linebreak \texttt{-report} \linebreak \texttt{-timestamp1} \linebreak \texttt{ts1} \linebreak \texttt{-timestamp2} \linebreak \texttt{ts2} utility. The \texttt{ttStats} \linebreak utility generates a report \linebreak between two snapshot IDs or timestamps. \linebreak You can only generate a text \linebreak version of a report in TimesTen \linebreak Scaleout. \\
\hline
\end{tabular}
\caption{Differences between \texttt{ttStats} in TimesTen Classic and TimesTen Scaleout}
\label{table:ttStats}
\end{table}
This section describes how the ttStats utility works in TimesTen Classic.

**Description**

The ttStats utility monitors database metrics (statistics, states, and other information) or takes and compares snapshots of metrics. It can perform the following functions.

- Monitor and display database performance metrics in real-time, calculating rates of change during each preceding interval.

- Collect and store snapshots of metrics to the database then produce reports with values and rates of change from a specified pair of snapshots. (These functions are performed through calls to the TT_STATS PL/SQL package.)

The ttStats utility gathers metrics from TimesTen Classic system tables, views, and built-in procedures. In reports, this includes information such as a summary of memory usage, connections, and load profile, followed by metrics (as applicable) for SQL statements, transactions, PL/SQL memory, replication, logs and log holds, checkpoints, cache groups, latches, locks, XLA, and TimesTen connection attributes. Monitoring displays a smaller set of key data, as shown later in this section.

For client DSNs, use the ttStatsCS version of the utility (UNIX and Linux systems or Windows).

There are three modes of operation:

- **Monitor mode** (default mode): Tracks database performance in real-time by monitoring a pre-determined set of metrics, displays those metrics (primarily those whose values have changed since the last display), and calculates rates of change in the values where appropriate. Information is output to the standard output for display to the user and is not stored to disk.

  If the duration or number of iterations is not specified, the monitoring runs until interrupted with Ctrl-C.

  **Note:** The set of metrics displayed in monitor mode is subject to change, depending on changes to the system tables and built-in procedures from which metrics are gathered.

- **Snapshot mode**: Takes a snapshot of metrics, according to the capture level, and stores them to database SYS_SNAPSHOT_XXXX system tables. Once the snapshot is taken, its ID number is displayed to the standard output. The capture level applies only to metrics from SYS_SYSTEMSTATS. For metrics from other sources, the same data are collected regardless of the capture level.

  By default, a "typical" set of metrics is collected, which suits most purposes, but you can specify a reduced "basic" set of metrics, all available metrics, or only those metrics from sources other than SYSTEMSTATS.

- **Report mode**: Generates a report from two specified snapshots of metrics. Reports are in HTML format by default, but you can request plain text format. You can specify an output file or display output to the standard output. For those familiar with Oracle Database performance analysis tools, the ttStats reports are similar in nature to Oracle Automatic Workload Repository (AWR) reports.
In monitor mode, the overhead of reading from the database is avoided. In snapshot mode and report mode, the `ttStats` utility is a convenient front end to the `TT_STATS` PL/SQL package provided by TimesTen. Refer to "TT_STATS" in Oracle TimesTen In-Memory Database PL/SQL Packages Reference for details on that package.

Notes: The `ttStats` utility has the following dependencies and limitations:

- Monitor mode requires features added to the `SYS.SYSTEMSTATS` table in TimesTen release 11.2.2.4.0.
- Snapshot and report modes require the `TT_STATS` PL/SQL package, added in TimesTen release 11.2.2.5.0.
- The utility cannot be used if you are connecting to TimesTen through a driver manager.

For information about built-in procedures mentioned, and the data they gather, see Chapter 3, "Built-In Procedures".

Required privilege

- **Monitor mode**: No special privilege is required to run monitor mode, but `ADMIN` privilege is required for the monitoring information to include data from the `ttSQLCmdCacheInfo` built-in procedure and `transaction_log_api` (XLA) table.

- **Snapshot and report mode**: By default, only the instance administrator has privilege to run in snapshot or report mode, due to security restrictions of the `TT_STATS` PL/SQL package. Any other user, including an `ADMIN` user, must be granted `EXECUTE` privilege for the `TT_STATS` package by the instance administrator or by an `ADMIN` user, such as in the following example:

```
GRANT EXECUTE ON SYS.TT_STATS TO scott;
```

Syntax

These are the supported name/value pairs:

```
ttStats [-h | -help]
ttStats [-V | -version]
ttStats [-monitor] [-interval seconds] [-duration seconds] [-iterations count] (DSN | -connStr connection_string)
ttStats -snapshot [-level capture_level] [-description snap_desc] (DSN | -connStr connection_string)
ttStats -report [-snap1 snapid1 -snap2 snapid2] [-html | -text] [-outputFile filename] (DSN | -connStr connection_string)
ttStats -drop [-begin_snap snapid1 [-end_snap snapid2]] (DSN | -connStr connection_string)
```

Note: Specify only one of `-monitor`, `-snapshot`, or `-report`.

Options

These are the supported options for the `ttStats` utility in TimesTen Scaleout:
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h</td>
<td>Prints the list of options and exits.</td>
</tr>
<tr>
<td>-help</td>
<td><strong>Note:</strong> This is also the result if nothing is entered on the ttStats command line, or if options are entered without a DSN or connection string.</td>
</tr>
<tr>
<td>-V</td>
<td>Prints the TimesTen release number and exits.</td>
</tr>
<tr>
<td>-version</td>
<td></td>
</tr>
<tr>
<td>-monitor</td>
<td>Run in real-time monitor mode. Monitors a pre-determined set of metrics and repeatedly displays the metrics and rates of change. Unlike in snapshot mode, nothing is stored to the database. <strong>Note:</strong> This is the default mode if neither <code>-monitor</code>, <code>-snapshot</code>, nor <code>-report</code> is specified.</td>
</tr>
<tr>
<td>-interval seconds</td>
<td>For monitor mode, this is the time interval between sets of metrics that are displayed, in seconds. The default is 10 seconds. Shorter intervals may negatively impact system performance.</td>
</tr>
<tr>
<td>-duration seconds</td>
<td>For monitor mode, this is the duration of how long ttStats runs, in seconds. After this duration, the utility exits. Also see information for the <code>-iterations</code> option.</td>
</tr>
<tr>
<td>-iterations count</td>
<td>For monitor mode, this is the number of iterations ttStats performs in gathering and displaying metrics. After these iterations, the utility exits. <strong>Note:</strong> If you specify both <code>-duration</code> and <code>-iterations</code>, monitoring stops when the first of the two limits is reached. If you specify neither, monitoring continues until interrupted by Ctrl-C</td>
</tr>
</tbody>
</table>
| -snapshot   | Collect a snapshot of metrics according to the capture level and store the metrics in the database. Once the snapshot is captured, its ID number is displayed. **Notes:**
|              | - TimesTen gathers all `SYSTEMSTATS` when you take a snapshot, but only those within the specified capture level have meaningful accumulated values. Metrics outside of the specified level have a value of 0 (zero). |
|              | - This option is implemented by a call to the `CAPTURE_SNAPSHOT` procedure of the `TT_STATS` PL/SQL package. |
| -level capture_level | For snapshot mode, this is the level of metrics to capture. The possible settings are as follows:  
|              | - 0: For metrics outside of `SYS.SYSTEMSTATS` only.  
|              | - 1: For only "basic" metrics.  
|              | - 2 (default): For "typical" metrics. This includes the basic metrics. This level is appropriate for most purposes.  
|              | - 3: For all available metrics.  
|              | Use the same level for any two snapshots to be used in a report. **Notes:**  
|              | - These levels correspond to the capture levels `NONE`, `BASIC`, `TYPICAL`, and `ALL` for the `TT_STATS` PL/SQL package.  
|              | - The capture level applies only to metrics from the `SYS.SYSTEMSTATS` table. For metrics from other sources, the same data are collected regardless of the capture level. |
| -description snap_desc | For snapshot mode, optionally use this to provide any description or notes for the snapshot, for example to distinguish it from other snapshots. |
This section provides examples of `ttStats` monitoring and report output.

**Note:** Examples are for illustrative purposes only. Details are subject to change.

**Monitor example**
This section shows sample output from monitor mode.

```bash
% ttStats sampledb
```
Connected to TimesTen Version 18.1.2.1.0 TimesTen Cache version 18.1.2.1.0.

<table>
<thead>
<tr>
<th>Description</th>
<th>Current</th>
<th>Rate/Sec</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>date.2016-Feb-20 16:49:25</td>
<td>-869676175380467200</td>
<td>1 sample #, not rate</td>
<td></td>
</tr>
<tr>
<td>connections.count</td>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>db.size.temp_high_water_mark.kb</td>
<td>7153</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>lock.locks_granted.immediate</td>
<td>832</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>log.log_bytes_per_transaction</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>log.lob.bookmark.log_force_lsn</td>
<td>0/12027904</td>
<td></td>
<td></td>
</tr>
<tr>
<td>log.lob.bookmark.log_write_lsn</td>
<td>0/12050944</td>
<td></td>
<td></td>
</tr>
<tr>
<td>log.lob.checkpoint_hold_lsn</td>
<td>0/12025856</td>
<td>sampledb.ds0</td>
<td></td>
</tr>
<tr>
<td>log.lob.checkpoint_hold_lsn</td>
<td>0/12023808</td>
<td>sampledb.dsl</td>
<td></td>
</tr>
<tr>
<td>stmt.executes.count</td>
<td>44</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>stmt.executes.selects</td>
<td>32</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** The number following the date and time is a numeric representation of the time of the snapshot and can be ignored.

The following command line example specifies that monitoring should stop after two iterations and uses a connection string to set a connection attribute value.

```
% ttStats -iterations 2
-connStr "DSN=sampledb;PLSQL_MEMORY_ADDRESS=20000000"
```

**Snapshot example**
The following examples take two snapshots at the default typical level:

```
% ttStats -snapshot sampledb
```

Connected to TimesTen Version 18.01.01.0001 TimesTen Cache version 18.1.2.1.0.
Snapshot 1 at TYPICAL level was successfully captured.

```
% ttStats -snapshot sampledb
```

Connected to TimesTen Version 18.1.2.1.0 TimesTen Cache version 18.1.2.1.0
Snapshot 2 at TYPICAL level was successfully captured.

**Report examples**
The following example creates a report from the snapshots generated in the previous section.

```
% ttStats -report -outputFile testreport.html -snap1 1 -snap2 2 sampledb
```

Connected to TimesTen Version 18.01.01.0001 TimesTen Cache version 18.1.2.1.0.
Report testreport.html was created.

The rest of this section shows excerpts from tables of metrics that a ttStats report generates. This output was produced using the default HTML format.
Figure 5–1 shows most of a report summary. The summary is good for a quick look at database metrics, with further details provided in the subsequent tables. It includes the following sections:

- **Memory Usage and Connections:** This information includes information about memory usage (the `db.size` metrics) and connections established (the `connections.established` metrics), including the number of client/server connections and direct connections. Any nonzero value for `connections.established.threshold_exceeded`, indicates too many connections.

- **Load Profile:** This gives an idea of the workload, showing the number of checkpoints, sorts (such as for ORDER BY statements), log buffer waits (delays when the log buffer fills and flushes to disk), inserts, updates, deletes, parses (such as for prepares), commits, and rollbacks. Consider whether there may be too many parses or too many durable commits (which are more expensive than non-durable commits).

- **Instance Efficiency Percentage:** Command Cache Hit %, Non-Parse/Execs %, Lock Hit %, and Log Buffer No Wait % are shown. All should be near 100%.
  - Lock Hit % estimates the percentage of lock requests that are granted without waiting.
  - Non-Parse/Execs % represents the percentage of SQL statement executions that do not require a prepare or reprepare.
  - Command Cache Hit % estimates the percentage of executions of SQL commands that can be found in the command cache.
  - Log Buffer No Wait % estimates the percentage of log insertions that do not have to wait due to log buffer waits.

---

**Note:** Examples are not shown for SWT cache group metrics, local cache group metrics, dynamic global cache group metrics, grid metrics, or latch metrics.

To include latch metrics, you must enable them for the database, using the `ttXactAdmin` utility as follows:

```
% ttXactAdmin -latchstats on DSN
```
Summary

Figure 5–1  ttStats report: summary

Statement statistics  Figure 5–2 shows statement metrics from a report. Both external metrics (stmt.executes, stmtprepares, and stmt.reprepares metrics) and internal metrics (zzinternal metrics) are shown. External metrics are generally of more interest. The stmt.executes.count value is the sum of all the other stmt.executes values.
Figure 5–2  ttStats report: statement statistics

Statement Statistics

<table>
<thead>
<tr>
<th>Statistics</th>
<th>Value</th>
<th>Rate (Per Second)</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>stmt executes alters</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>stmt executes count</td>
<td>4726</td>
<td>562.25</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>stmt executes creates</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>stmt executes deletes</td>
<td>1</td>
<td>13</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>stmt executes drops</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>stmt executes inserts</td>
<td>2475</td>
<td>309.38</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>stmt executes merges</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>stmt executes selects</td>
<td>2240</td>
<td>280</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>stmt executes updates</td>
<td>16</td>
<td>2</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>stmt preprocess command_cache_hits</td>
<td>5</td>
<td>63</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>stmt prepares count</td>
<td>20</td>
<td>2.5</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>stmt prepares automatic</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>stmt prepares count</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
</tbody>
</table>

Transaction statistics  Figure 5–3 shows transaction metrics from a report. The txn.commits.count value is the sum of the txn.commits.durable and txn.commits.nondurable values. Other metrics shown are subsets of these metrics.

Figure 5–3  ttStats report: transaction statistics

Transaction Statistics

<table>
<thead>
<tr>
<th>Statistics</th>
<th>Value</th>
<th>Rate (Per Second)</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>txn.commits.count</td>
<td>125</td>
<td>15</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>txn.commits.durable</td>
<td>1</td>
<td>.13</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>txn.commits.internal</td>
<td>125</td>
<td>15</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>txn.commits.internal.xs</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>txn.commits.nondurable</td>
<td>127</td>
<td>18</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>txn.commits.replicated.durable</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>txn.commits.replicated.nondurable</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>txn.needs txns</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>txn.needs txns with dependencies</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>zcz internal para txns</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>zcz internal para txns with dependencies</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>zcz internal para txns</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
</tbody>
</table>

SQL statistics: sort by executions  Figure 5–4 shows an excerpt of SQL execution metrics from the SQL Statistics section of a report. When you look at the "sort by executions" metrics and "sort by preparations" metrics (shown in the next section), note which statements are used a lot and the number of preparations and the number of executions for each statement. Ideally, a statement is not prepared many times.
Figure 5–4  ttStats report: SQL execution statistics

SQL Statistics

SQL Sort by Executions  Only top 30 SQL Commands are displayed

<table>
<thead>
<tr>
<th>Executions</th>
<th>% Total</th>
<th>Cmd ID</th>
<th>Cmd Text</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>2190</td>
<td>43.81</td>
<td>215038464</td>
<td>SELECT DESC_ID FROM SNAPSHOT_DESCRIPTION WHERE TRO</td>
<td>SQLCmdCacheInfo</td>
</tr>
<tr>
<td>2170</td>
<td>43.32</td>
<td>215039269</td>
<td>INSERT INTO SNAPSHOT_VALUE_PARAMATY VALUES (X4, 0)</td>
<td>SQLCmdCacheInfo</td>
</tr>
<tr>
<td>142</td>
<td>2.84</td>
<td>215041605</td>
<td>INSERT INTO SNAPSHOT_DESCRIPTION VALUES (02, 0)</td>
<td>SQLCmdCacheInfo</td>
</tr>
<tr>
<td>142</td>
<td>2.84</td>
<td>215041605</td>
<td>SELECT MAX (DESC_ID) FROM SNAPSHOT_DESCRIPTION</td>
<td>SQLCmdCacheInfo</td>
</tr>
<tr>
<td>138</td>
<td>2.71</td>
<td>215156736</td>
<td>INSERT INTO SNAPSHOT_VALUE_SQL_VALUES (X6, 0) 1</td>
<td>SQLCmdCacheInfo</td>
</tr>
<tr>
<td>65</td>
<td>1.29</td>
<td>215074681</td>
<td>INSERT INTO SNAPSHOT_VALUE_CONFIG VALUES (X3, 0)</td>
<td>SQLCmdCacheInfo</td>
</tr>
<tr>
<td>14</td>
<td>0.28</td>
<td>215053576</td>
<td>INSERT INTO SNAPSHOT_VALUE_LOGHOLD VALUES (X5, 0)</td>
<td>SQLCmdCacheInfo</td>
</tr>
<tr>
<td>12</td>
<td>0.24</td>
<td>155315756</td>
<td>select name, owner, mimval, mimvel, atval, atvel, er</td>
<td>SQLCmdCacheInfo</td>
</tr>
<tr>
<td>12</td>
<td>0.24</td>
<td>227352678</td>
<td>update time, rrepears set sendshrink + n. sends</td>
<td>SQLCmdCacheInfo</td>
</tr>
<tr>
<td>12</td>
<td>0.24</td>
<td>226287668</td>
<td>SELECT MAKEREPLERD (DU) FROM SNAPSHOT_REPL_PEER</td>
<td>SQLCmdCacheInfo</td>
</tr>
<tr>
<td>12</td>
<td>0.24</td>
<td>215037394</td>
<td>INSERT INTO SNAPSHOT_REPL_PEER VALUES (X2, 0) 1</td>
<td>SQLCmdCacheInfo</td>
</tr>
<tr>
<td>12</td>
<td>0.24</td>
<td>215069922</td>
<td>SELECT REPELERD (DU) FROM SNAPSHOT_REPL_REPL WHERE R</td>
<td>SQLCmdCacheInfo</td>
</tr>
<tr>
<td>12</td>
<td>0.24</td>
<td>215069922</td>
<td>INSERT INTO SNAPSHOT_VALUE_REPL VALUES (X3, 0)</td>
<td>SQLCmdCacheInfo</td>
</tr>
<tr>
<td>10</td>
<td>0.22</td>
<td>215069282</td>
<td>INSERT INTO SNAPSHOT_VALUE_LSOL VALUES (X3, 0)</td>
<td>SQLCmdCacheInfo</td>
</tr>
</tbody>
</table>

SQL statistics: sort by preparations  Figure 5–5 shows an excerpt of SQL preparation metrics from the SQL Statistics section of a report. Refer to the discussion in the preceding "sort by executions" section.

Figure 5–5  ttStats report: SQL preparation statistics

SQL Sort by Preparations  Only top 30 SQL Commands are displayed

<table>
<thead>
<tr>
<th>Preparations</th>
<th>% Total</th>
<th>Cmd ID</th>
<th>CmdText</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>15.15</td>
<td>215547165</td>
<td>select owner#name_namespc.obj objtype#dline#nlist</td>
<td>SQLCmdCacheInfo</td>
</tr>
<tr>
<td>2</td>
<td>4.26</td>
<td>215064877</td>
<td>CONNIT</td>
<td>SQLCmdCacheInfo</td>
</tr>
<tr>
<td>2</td>
<td>4.26</td>
<td>215873956</td>
<td>INSERT INTO SNAPSHOT_DESCRIPTION VALUES (X2, 0)</td>
<td>SQLCmdCacheInfo</td>
</tr>
<tr>
<td>2</td>
<td>4.26</td>
<td>215835058</td>
<td>SELECT DESC_ID FROM SNAPSHOT_DESCRIPTION _WHERE TRO</td>
<td>SQLCmdCacheInfo</td>
</tr>
<tr>
<td>2</td>
<td>4.26</td>
<td>215835058</td>
<td>SELECT DESC_ID FROM SNAPSHOT_DESCRIPTION _WHERE TRO</td>
<td>SQLCmdCacheInfo</td>
</tr>
<tr>
<td>1</td>
<td>2.10</td>
<td>157941720</td>
<td>select owner#name from sys$sys where object #1</td>
<td>SQLCmdCacheInfo</td>
</tr>
<tr>
<td>1</td>
<td>2.10</td>
<td>157943267</td>
<td>select owner#name_namespc_0_timestamp_0_epoch</td>
<td>SQLCmdCacheInfo</td>
</tr>
<tr>
<td>1</td>
<td>2.13</td>
<td>147950248</td>
<td>select order#column types from sys$accessed where</td>
<td>SQLCmdCacheInfo</td>
</tr>
<tr>
<td>1</td>
<td>2.13</td>
<td>147950260</td>
<td>call tt_stats capture snapshot</td>
<td>SQLCmdCacheInfo</td>
</tr>
<tr>
<td>1</td>
<td>2.13</td>
<td>147950264</td>
<td>select piece#length piece from sys$ids from sys$ where</td>
<td>SQLCmdCacheInfo</td>
</tr>
<tr>
<td>1</td>
<td>2.13</td>
<td>147950264</td>
<td>select piece#length piece from sys$ids from sys$ where</td>
<td>SQLCmdCacheInfo</td>
</tr>
<tr>
<td>1</td>
<td>2.13</td>
<td>147950264</td>
<td>select piece#length piece from sys$ids from sys$ where</td>
<td>SQLCmdCacheInfo</td>
</tr>
<tr>
<td>1</td>
<td>2.13</td>
<td>147950264</td>
<td>select piece#length piece from sys$ids from sys$ where</td>
<td>SQLCmdCacheInfo</td>
</tr>
</tbody>
</table>

SQL statistics: command texts  Figure 5–6 shows an excerpt of SQL statements from the SQL Statistics section of a report. This report shows the complete text of each statement listed in the preceding "sort by executions" and "sort by preparations" reports, where longer statements are abbreviated.
**Figure 5–6  ttStats report: SQL command texts**

<table>
<thead>
<tr>
<th>SQL ID</th>
<th>SQL Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>220887160</td>
<td>select owner.object_name, namespace.object_name, object.status, flags from sys.obj where owner=1 and name=2 and namespace=3</td>
</tr>
<tr>
<td>230373952</td>
<td>INSERT INTO SNAPSHOTS_DESCRIPTION VALUES(28, 81)</td>
</tr>
<tr>
<td>230528504</td>
<td>SELECT DESC_ID FROM SNAPSHOTS_DESCRIPTION WHERE TRN.DESC_ID = TRN.81</td>
</tr>
<tr>
<td>230528504</td>
<td>SELECT MAX(DESC_ID) FROM SNAPSHOTS_DESCRIPTION</td>
</tr>
<tr>
<td>157913728</td>
<td>select owner.object_name from sys.obj where object=1</td>
</tr>
<tr>
<td>157942932</td>
<td>select owner.object_name, namespace.object_name, property_order, property, object.type from sys.dependency where property_order = 1 and object.type = property_order order by property_order</td>
</tr>
<tr>
<td>157942932</td>
<td>select owner.object_name, namespace.object_name, property_order, property, object.type from sys.dependency where property_order = 1 and object.type = property_order order by property_order</td>
</tr>
<tr>
<td>157942932</td>
<td>select owner.object_name, namespace.object_name, property_order, property, object.type from sys.dependency where property_order = 1 and object.type = property_order order by property_order</td>
</tr>
<tr>
<td>157942932</td>
<td>select owner.object_name, namespace.object_name, property_order, property, object.type from sys.dependency where property_order = 1 and object.type = property_order order by property_order</td>
</tr>
</tbody>
</table>

---

**PL/SQL memory statistics**  Figure 5–7 shows PL/SQL memory metrics from a report. These are metrics from the ttPLSQLMemoryStats built-in procedure. There should not be a significant difference between the start and end values of GetHitRatio or PinHitRatio.

**Figure 5–7  ttStats report: PL/SQL memory statistics**

**PL/SQL Memory Statistics**

<table>
<thead>
<tr>
<th>Statistics</th>
<th>Reg Val</th>
<th>End Val</th>
<th>Rate (Per Sec)</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>CurrentConnectionMemory</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>ttPLSQLMemoryStats</td>
</tr>
<tr>
<td>DeferredCusps</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>ttPLSQLMemoryStats</td>
</tr>
<tr>
<td>GetHitRatio</td>
<td>.84</td>
<td>.85</td>
<td>-</td>
<td>ttPLSQLMemoryStats</td>
</tr>
<tr>
<td>Gets</td>
<td>196</td>
<td>206</td>
<td>1.25</td>
<td>ttPLSQLMemoryStats</td>
</tr>
<tr>
<td>Invalidations</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>ttPLSQLMemoryStats</td>
</tr>
<tr>
<td>PinHitRatio</td>
<td>.53</td>
<td>.54</td>
<td>-</td>
<td>ttPLSQLMemoryStats</td>
</tr>
<tr>
<td>Puts</td>
<td>504</td>
<td>314</td>
<td>1.25</td>
<td>ttPLSQLMemoryStats</td>
</tr>
<tr>
<td>Pns</td>
<td>569</td>
<td>501</td>
<td>1.5</td>
<td>ttPLSQLMemoryStats</td>
</tr>
<tr>
<td>Reloads</td>
<td>25</td>
<td>28</td>
<td>13</td>
<td>ttPLSQLMemoryStats</td>
</tr>
</tbody>
</table>

---

**Replication statistics**  Figure 5–8 shows replication metrics from a report. For each transmitter (where there could be multiple transmitters per master), the metrics indicate advancement through the log, including how many records were sent to the receiver. Repl_Peer indicates the subscriber. Repl_Log_Behind and Repl_Latency are significant in indicating whether replication is keeping up with the database workload.
Parallel replication/AWT statistics  

Figure 5–9 shows an excerpt of parallel replication/AWT metrics from a report. Repl_Peer indicates the subscriber. When parallel replication/AWT is configured, if replication metrics (discussed in the previous section) indicate difficulty keeping up with the workload, parallel replication/AWT metrics may indicate why. Each value is an aggregate across all tracks, but you can click Show Details (at the end of the metrics table, not shown here) to see the data for each track. High values for track switching—"switchin" and "switchout" metrics—may indicate contention. High values for "waits" metrics are also problematic, indicating situations such as one transaction having to wait for a previous transaction to commit before it can begin or before it can commit.

Log statistics  

Figure 5–10 shows logs from a report. The report output notes that numbers in log.file.earliest and log.file.latest represent values in the begin and end snapshots. The log.buffer.waits metric is of particular interest. Log buffer waits occur when application processes cannot insert transaction data to the log buffer and must stall to wait for log buffer space to be freed. The usual reason for this is that the log flusher thread has not cleared out data fast enough. This may indicate that log buffer space is insufficient, disk bandwidth is insufficient, writing to disk is taking too long, or the log flusher is CPU-bound. (Also see "Managing transaction log buffers and files" and "Increase LogBufMB if needed" in Oracle TimesTen In-Memory Database Operations Guide.)
Figure 5–10  ttStats report: log statistics

Log Statistics

- Numbers in log_file.earliest and log_file.latest represent values in begin snapshot and end snapshot

<table>
<thead>
<tr>
<th>Statistics</th>
<th>Value</th>
<th>Rate (Per Second)</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>log_buffer_bytes_inserted</td>
<td>750560</td>
<td>57630</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>log_buffer_insertions</td>
<td>5674</td>
<td>1090.25</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>log_buffer_writes</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>log_commit_bytes_read</td>
<td>461168</td>
<td>67521</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>log_commit_file_reads</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>log_file_earliest</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>log_file_latest</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>log_file_reads</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>log_file_writes</td>
<td>17</td>
<td>2.13</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>log_files_generated</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>log_forces</td>
<td>11</td>
<td>1.38</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>log_int_data</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>log_log_bytes_per_transaction</td>
<td>87</td>
<td>10.08</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>log_recovers</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>zintelligent_log_buffer_bytes_inserted</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>zintelligent_log_buffer_insertions_zap</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>zintelligent_log_strand_switches_insertion_zap</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>zintelligent_log_strand_switches_strand_zap</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>zintelligent_log_releaser_log_last_sleeps</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>zintelligent_log_releaser_log_waits</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
</tbody>
</table>

Log holds  Figure 5–11 shows log hold information from a report. It shows bookmark positions for checkpoint log holds for each checkpoint file, and bookmark positions for replication log holds for each replication subscriber. This report may also show log hold information for backup, XLA, and long-running transactions. Where the begin and end values are the same, there have been no movements.

Ideally there will be evidence of a smooth progression through the log file. (The ttStats monitor information may be more useful in tracking this.)

Figure 5–11  ttStats report: log holds

Log Holds

<table>
<thead>
<tr>
<th>Statistics</th>
<th>Begin Value</th>
<th>End Value</th>
<th>Done</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>loghold_Checkpoint_hold_Ln</td>
<td>01794582</td>
<td>01794682</td>
<td>total</td>
<td>logholds, tbBookmark</td>
</tr>
<tr>
<td>loghold_Checkpoint_hold_Ln</td>
<td>01795192</td>
<td>01795292</td>
<td>total</td>
<td>logholds, tbBookmark</td>
</tr>
<tr>
<td>loghold_Replacation_hold_Ln</td>
<td>017687464</td>
<td>017687466</td>
<td>total</td>
<td>logholds, tbBookmark</td>
</tr>
<tr>
<td>loghold_Replacation_hold_Ln</td>
<td>017687746</td>
<td>017687748</td>
<td>total</td>
<td>logholds, tbBookmark</td>
</tr>
<tr>
<td>loghold_Replacation_hold_Ln</td>
<td>017687464</td>
<td>017687466</td>
<td>total</td>
<td>logholds, tbBookmark</td>
</tr>
<tr>
<td>loghold_Replacation_hold_Ln</td>
<td>017687746</td>
<td>017687748</td>
<td>total</td>
<td>logholds, tbBookmark</td>
</tr>
<tr>
<td>loghold_Replacation_hold_Ln</td>
<td>017687464</td>
<td>017687466</td>
<td>total</td>
<td>logholds, tbBookmark</td>
</tr>
<tr>
<td>loghold_Replacation_hold_Ln</td>
<td>017687746</td>
<td>017687748</td>
<td>total</td>
<td>logholds, tbBookmark</td>
</tr>
<tr>
<td>loghold_Replacation_hold_Ln</td>
<td>017687464</td>
<td>017687466</td>
<td>total</td>
<td>logholds, tbBookmark</td>
</tr>
</tbody>
</table>

Checkpoint statistics  Figure 5–12 shows checkpoint metrics from a report.
**Figure 5–12  ttStats report: checkpoint statistics**

### CheckPoint Statistics

<table>
<thead>
<tr>
<th>Statistics</th>
<th>Value</th>
<th>Rate (Per Second)</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>checkbytes_written (MB)</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>checkbytes_written during recovery (MB)</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>checkcompleted</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>checkcompleted_fuzzy</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>checkwrites</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
</tbody>
</table>

**Cache group statistics: AWT cache groups**  
Figure 5–13 shows AWT cache group metrics from a report. Values are aggregates across all AWT cache groups. Information includes the number of calls to the Oracle database; the number of commits, rollbacks, and retries on Oracle; and the number of rows inserted, deleted, and updated by PL/SQL operations and by SQL operations.

**Figure 5–13  ttStats report: AWT cache group statistics**

### Cache Group Statistics

<table>
<thead>
<tr>
<th>AWT Cache Group Statistics</th>
<th>Value</th>
<th>Rate (Per Second)</th>
<th>Rate (Per AWT Task)</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>cg.awt.cache_mode_in Oracle</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>sys SystemStats table</td>
</tr>
<tr>
<td>cg.awt.commit_count_oracle</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>sys SystemStats table</td>
</tr>
<tr>
<td>cg.awt.sql_mode_in Oracle</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>sys SystemStats table</td>
</tr>
<tr>
<td>cg.awt.sql_mode_inserts</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>sys SystemStats table</td>
</tr>
<tr>
<td>cg.awt.sql_mode_updates</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>sys SystemStats table</td>
</tr>
<tr>
<td>cg.awt.sql_mode_inserts</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>sys SystemStats table</td>
</tr>
<tr>
<td>cg.awt.sql_mode_updates</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>sys SystemStats table</td>
</tr>
<tr>
<td>cg.awt.sql_mode_inserts</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>sys SystemStats table</td>
</tr>
<tr>
<td>cg.awt.sql_mode_updates</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>sys SystemStats table</td>
</tr>
<tr>
<td>cg.awt.sql_mode_inserts</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>sys SystemStats table</td>
</tr>
<tr>
<td>cg.awt.sql_mode_updates</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>sys SystemStats table</td>
</tr>
<tr>
<td>cg.awt.sql_mode_exec_time</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>sys SystemStats table</td>
</tr>
</tbody>
</table>

**Cache group statistics: auto-refresh cache groups**  
Figure 5–14 shows auto-refresh cache group metrics from a report. Values are aggregates across all auto-refresh cache groups. Whether cache groups are in full or incremental refresh mode is reflected by the cg.autorefresh.full_refreshes value with respect to the cg.autorefresh.cycles.completed value (which indicates the total number of refreshes).
Figure 5–14  ttStats report: auto-refresh cache group statistics

<table>
<thead>
<tr>
<th>Statistics</th>
<th>Value</th>
<th>Rate (Per Sec)</th>
<th>Rate (Per Cycle)</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>cache_hits</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>sys.SystemStats table</td>
</tr>
<tr>
<td>db.index.hash.deletes</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>sys.SystemStats table</td>
</tr>
<tr>
<td>db.index.hash.inserts</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>sys.SystemStats table</td>
</tr>
<tr>
<td>db.index.hash.inserts.recovery rebuild</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>sys.SystemStats table</td>
</tr>
<tr>
<td>db.index.hash.revs.fetched.count</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>sys.SystemStats table</td>
</tr>
<tr>
<td>db.index.hash.revs.fetched.repl</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>sys.SystemStats table</td>
</tr>
<tr>
<td>db.index.hash.scans.count</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>sys.SystemStats table</td>
</tr>
<tr>
<td>db.table.revs.repl</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>sys.SystemStats table</td>
</tr>
<tr>
<td>db.table.revs.scans.repl</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>sys.SystemStats table</td>
</tr>
<tr>
<td>db.table.scans.repl</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>sys.SystemStats table</td>
</tr>
<tr>
<td>db.table.update</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>sys.SystemStats table</td>
</tr>
<tr>
<td>db.index.rebuilds</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>sys.SystemStats table</td>
</tr>
<tr>
<td>db.index.temp.created</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>db.index.temp.revs.fetched.count</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>db.index.temp.revs.fetched.repl</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>db.index.temp.scans.repl</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>db.index.refs</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>sys.SystemStats table</td>
</tr>
<tr>
<td>db.page.merges</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>sys.SystemStats table</td>
</tr>
<tr>
<td>db.table.idiom_loop</td>
<td>28</td>
<td>3.5</td>
<td>0</td>
<td>sys.SystemStats table</td>
</tr>
<tr>
<td>db.pass.count</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>sys.SystemStats table</td>
</tr>
<tr>
<td>db.size.pers_allocated_kb</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>db.size.pers_high_watermark_kb</td>
<td>595</td>
<td>74.38</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>db.size.pers_in_use_kb</td>
<td>595</td>
<td>74.38</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>db.size.hib_allocated_kb</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>db.size.hib_high_watermark_kb</td>
<td>30</td>
<td>3.75</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>db.size.hib_in_use_kb</td>
<td>46</td>
<td>5.95</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>db.tables.full_scans</td>
<td>17</td>
<td>2.13</td>
<td>0</td>
<td>sys.SystemStats table</td>
</tr>
</tbody>
</table>

Database activity statistics  Figure 5–15 shows an excerpt of database activity metrics from a report—index activity, memory activity, and table activity. For hash indexes and range indexes, information includes deletes, inserts, rows fetched, and scans. For memory usage, it shows size data. For tables, it shows rows read, deleted, inserted, and updated.

Figure 5–15  ttStats report: database activity statistics

DB Activity Statistics

Lock statistics  Figure 5–16 shows lock metrics from a report. This provides information about deadlocks, locks acquired, locks granted, and lock timeouts. In
particular, `lock.deadlocks`, `lock.locks_granted.wait`, and `lock.timeouts` may indicate lock contention.

**Figure 5–16** ttStats report: lock statistics

### Lock Statistics

<table>
<thead>
<tr>
<th>Statistics</th>
<th>Value</th>
<th>Rate (Per Second)</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>lock.deadlocks</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>lock.locks_acquired_dml</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>lock.locks_acquired_table_scans</td>
<td>114</td>
<td>14.15</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>lock.locks_granted_immediate</td>
<td>12000</td>
<td>100</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>lock.locks_granted_wait</td>
<td>1</td>
<td>13</td>
<td>SystemStats table</td>
</tr>
<tr>
<td>lock.timeouts</td>
<td>0</td>
<td>0</td>
<td>SystemStats table</td>
</tr>
</tbody>
</table>

**XLA information** Figure 5–17 shows XLA bookmark information from a report. For each bookmark, the begin and end values are shown for `Purge_LSN`, which indicates the position in the log file prior to which information has been purged, and for `Log_Behind`, which indicates whether there is a lag between the position of the XLA transaction and the position of the most recent log file.

**Figure 5–17** ttStats report: XLA information

### XLA Information

- `-1/-1` in `Begin Purge_LSN` means XLA is not configured in `begin_snapshot`.

<table>
<thead>
<tr>
<th>Bookmark Name</th>
<th>Purge_LSN</th>
<th>Log_Behind</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Begin Value</td>
<td>End Value</td>
</tr>
<tr>
<td>bookmark</td>
<td>02923554560</td>
<td>02923554500</td>
</tr>
</tbody>
</table>

**Configuration parameters** Figure 5–18 shows database configuration parameter settings from a report. For reference, each report shows the begin and end values of each TimesTen connection attribute.

For information about connection attributes, see Chapter 2, "Connection Attributes".

**Figure 5–18** ttStats report: configuration parameters

### Configuration Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Begin Value</th>
<th>End Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CacheAvailMethod</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>CacheAvailParallelism</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>CacheGridEnable</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>CacheGridSize</td>
<td>60</td>
<td>60</td>
</tr>
<tr>
<td>CliqFrequency</td>
<td>600</td>
<td>600</td>
</tr>
<tr>
<td>CliqLogVolume</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CliqRatio</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CommitBufferSizeMax</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>ConnectionCharacterSet</td>
<td>UTF8</td>
<td>UTF8</td>
</tr>
<tr>
<td>ConnectionName</td>
<td>repdo1</td>
<td>repdo1</td>
</tr>
<tr>
<td>Connections</td>
<td>200</td>
<td>230</td>
</tr>
<tr>
<td>ODLCommitBehavior</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ODLPropagationAction</td>
<td>INCLUDE</td>
<td>INCLUDE</td>
</tr>
<tr>
<td>ODLReplicatorLevel</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>DatabaseCharacterSet</td>
<td>ALL12UTF8</td>
<td>ALL12UTF8</td>
</tr>
<tr>
<td>DataStore</td>
<td>datastore.repdo1</td>
<td>datastore.repdo1</td>
</tr>
</tbody>
</table>
See also

ttStatsConfig
ttStatsConfigGet
This section describes how the ttStats utility works in TimesTen Scaleout.

Description

The ttStats utility monitors database metrics (statistics, states, and other information) or takes and compares snapshots of metrics. It can perform the following functions.

- Monitor and display database performance metrics in real-time, calculating rates of change during each preceding interval.

The ttStats utility gathers metrics from TimesTen system tables, views, and built-in procedures. In reports, this includes information such as a summary of memory usage, connections, and load profile, followed by metrics (as applicable) for SQL statements, transactions, PL/SQL memory, replication, logs and log holds, checkpoints, cache groups, latches, locks, and TimesTen connection attributes. Monitoring displays a smaller set of key data, as shown later in this section.

For client DSNs, use the ttStatsCS version of the utility (UNIX and Linux systems or Windows).

There are three modes of operation:

- Monitor mode (default mode): Tracks database performance in real-time by monitoring a pre-determined set of metrics, displays those metrics (primarily those whose values have changed since the last display), and calculates rates of change in the values where appropriate. Information is output to the standard output for display to the user and is not stored to disk. If the duration or number of iterations is not specified, the monitoring runs until interrupted with Ctrl-C.

  **Note:** The set of metrics displayed in monitor mode is subject to change, depending on changes to the system tables and built-in procedures from which metrics are gathered.

- Snapshot mode: In TimesTen Scaleout, the ttStats daemon automatically takes snapshots of the TimesTen Scaleout database based on the parameters of the ttStatsConfig built-in procedure.

  If you use the -snapshot option of the ttStats utility, the -description option is mandatory. When you execute ttStats -snapshot -description description, you can associate a description to the latest system generated snapshot. Provide any description or notes for the snapshot, for example to distinguish it from other snapshots.

- Report mode: Generates a report from two specified snapshots or two specified timestamps of metrics. Reports are only available in plain text format. You must specify an output file with the -outputFile option. For those familiar with Oracle Database performance analysis tools, the ttStats reports are similar in nature to Oracle Automatic Workload Repository (AWR) reports.

  In monitor mode, the overhead of reading from the database is avoided. In snapshot mode and report mode, the ttStats utility is a convenient front end to the TT_STATS PL/SQL package provided by TimesTen. Refer to "TT_STATS" in Oracle TimesTen In-Memory Database PL/SQL Packages Reference for details on that package.
Notes: The ttStats utility has the following dependencies and limitations:

- The utility cannot be used if you are connecting to TimesTen through a driver manager.

For information about built-in procedures mentioned, and the data they gather, see Chapter 3, "Built-In Procedures".

Required privilege

- **Monitor mode**: No special privilege is required to run monitor mode, but ADMIN privilege is required for the monitoring information to include data from the ttSQLCmdCacheInfo built-in procedure.

- **Snapshot and report mode**: By default, only the instance administrator has privilege to create snapshots, get snapshot information and run in report mode, due to security restrictions of the TT_STATS PL/SQL package. Any other user, including an ADMIN user, must be granted EXECUTE privilege for the TT_STATS package by the instance administrator or by an ADMIN user, such as in the following example:

  ```sql
  GRANT EXECUTE ON SYS.TT_STATS TO scott;
  ```

Syntax

```
ttStats [-h | -help]
ttStats [-V | -version]
ttStats [-monitor] [-interval seconds]
  [-duration seconds] [-iterations count]
  (DSN | -connStr connection_string)
ttStats -snapshot -description snap_desc
  (DSN | -connStr connection_string)
ttStats -report [-snap1 snapid1 -snap2 snapid2 | -timestamp1 ‘timestamp1’ -timestamp2 ‘timestamp2’]
  -outputFile filename
ttStats -snapshotInfo (DSN | -connStr connection_string)
```

**Note**: Specify only one of -monitor, -snapshot, or -report.

Options

These are the supported options for the ttStats utility in TimesTen Scaleout:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h</td>
<td>Prints the list of options and exits.</td>
</tr>
<tr>
<td>-help</td>
<td>Note: This is also the result if nothing is entered on the ttStats command line, or if options are entered without a DSN or connection string.</td>
</tr>
<tr>
<td>-V</td>
<td>Prints the TimesTen release number and exits.</td>
</tr>
<tr>
<td>-version</td>
<td></td>
</tr>
</tbody>
</table>
Option | Description
--- | ---
-`monitor` | Run in real-time monitor mode. Monitors a pre-determined set of metrics and repeatedly displays the metrics and rates of change. Unlike in snapshot mode, nothing is stored to the database.
**Note**: This is the default mode if neither `-monitor`, `-snapshot`, nor `-report` is specified.

-`interval seconds` | For monitor mode, this is the time interval between sets of metrics that are displayed, in seconds. The default is 10 seconds. Shorter intervals may negatively impact system performance.

-`duration seconds` | For monitor mode, this is the duration of how long `ttStats` runs, in seconds. After this duration, the utility exits.
Also see information for the `-iterations` option.

-`iterations count` | For monitor mode, this is the number of iterations `ttStats` performs in gathering and displaying metrics. After these iterations, the utility exits.
**Note**: If you specify both `-duration` and `-iterations`, monitoring stops when the first of the two limits is reached. If you specify neither, monitoring continues until interrupted by Ctrl-C.

-`snapshot -description snap_desc` | If you use the `-snapshot` option, the `-description` option is mandatory. When you execute `ttStats -snapshot -description description`, you can associate a description to the latest system generated snapshot. Provide any description or notes for the snapshot, for example to distinguish it from other snapshots.

-`report` | Generate a report from two specified snapshots or two specified timestamps. Use the `-snapshotInfo` option to see available snapshots for your database.

-`snap1 snapid1` | For report mode, this is the snapshot ID of the first snapshot.

-`snap2 snapid2` | For report mode, this is the snapshot ID of the second snapshot. The report period must span at least four existing snapshot ID values. Therefore, you must have at least three snapshots between `-snap1` and `-snap2`.

-`timestamp1 'timestamp1'` | For report mode, this specifies the timestamp of the first snapshot. The timestamp must use the `YYYY-MM-DD HH:MM:SS` format and be wrapped in straight single quotes.

-`timestamp2 'timestamp2'` | For report mode, this specifies the timestamp of the second snapshot. The timestamp must use the `YYYY-MM-DD HH:MM:SS` format and be wrapped in straight single quotes.

-`outputFile filename` | For report mode, optionally specify a file path and name where the report is to be written. If no file is specified, TimesTen writes the to the standard output.

-`snapshotInfo` | Prints the snapshot ID, date, time, and the description of all snapshots.

-`connStr connstring` | To specify and connect to the database from which to gather metrics, do one of the following:
- Specify an ODBC connection string, preceded by `-connStr`.
- Specify a DSN (data source name), without `-connStr`, at the end of the command line.
See “Specifying Data Source Names to identify TimesTen databases” in *Oracle TimesTen In-Memory Database Operations Guide* for information about TimesTen DSNs.
Examples

This section provides examples of ttStats monitoring and report output.

**Note:** Examples are for illustrative purposes only. Details are subject to change.

### Monitor example

This section shows sample output from monitor mode.

% ttStats samplegriddb

Connected to TimesTen Version 18.01.01.0001 Oracle TimesTen IMDB version 18.1.2.1.0.

Waiting for 10 seconds for the next snapshot

<table>
<thead>
<tr>
<th>Description</th>
<th>Current</th>
<th>Rate/Sec</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>date.2016-Mar-16 15:29:23</td>
<td>1458167363</td>
<td>1</td>
<td>sample #, not rate</td>
</tr>
<tr>
<td>connections.count</td>
<td>20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lock.locks_granted.immediate</td>
<td>124817</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>log.log_bytes_per_transaction</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>loghold.bookmark.log_force_lsn</td>
<td>0/21102592</td>
<td></td>
<td></td>
</tr>
<tr>
<td>loghold.bookmark.log_write_lsn</td>
<td>0/21102856</td>
<td></td>
<td></td>
</tr>
<tr>
<td>loghold.checkpoint_hold_lsn</td>
<td>0/21100544</td>
<td></td>
<td>samplegriddb.ds0</td>
</tr>
<tr>
<td>loghold.checkpoint_hold_lsn</td>
<td>0/21078016</td>
<td></td>
<td>samplegriddb.ds1</td>
</tr>
<tr>
<td>plsql.GetHitRatio</td>
<td>0.640</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>plsql.GetHits</td>
<td>258.000</td>
<td>0.200</td>
<td></td>
</tr>
<tr>
<td>plsql.Gets</td>
<td>403.000</td>
<td>0.200</td>
<td></td>
</tr>
<tr>
<td>plsql.PinHitRatio</td>
<td>0.557</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>plsql.PinHits</td>
<td>424.000</td>
<td>0.200</td>
<td></td>
</tr>
<tr>
<td>plsql.Pins</td>
<td>761.000</td>
<td>0.200</td>
<td></td>
</tr>
<tr>
<td>stmt.executes.count</td>
<td>24407</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>stmt.executes.selects</td>
<td>620</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** The number following the date and time is a numeric representation of the time of the snapshot and can be ignored.

The following command line example specifies that monitoring should stop after two iterations and uses a connection string to set a connection attribute value.

% ttStats -iterations 2 -connStr "DSN=samplegriddb"

### Report examples

The following example creates a report from the snapshots generated in the previous section.

**Note:** The report period must span at least four existing snapshot ID values. Therefore, you must have at least three snapshots between -snap1 and -snap2.

% ttStats -report -outputFile testreport.txt -snap1 1 -snap2 5 samplegriddb

Connected to TimesTen Version 18.01.01.0001 Oracle TimesTen IMDB version
18.1.2.1.0.
Report testreport.txt was created.

The rest of this section shows excerpts from tables of metrics for a ttStats report for a TimesTen Scaleout database and a ttStats report for an element of a TimesTen Scaleout database. This output was produced using the default plain text output format.

- TimesTen Scaleout ttStats report
- TimesTen Scaleout element ttStats report

---

**Note:** To include latch metrics, you must enable them for the database, using the ttXactAdmin utility as follows:

```
% ttXactAdmin -latchstats on DSN
```

---

**TimesTen Scaleout ttStats report**
The following sections show excerpts from tables of metrics for a ttStats report for a TimesTen Scaleout database.

**TimesTen Scaleout snapshot summary** Displays a grid snapshot summary. The TimesTen Scaleout snapshot summary shows information regarding the snapshots that you specified for the ttStats report.

**TimesTen Scaleout elements** Displays information about each database element. This section of the report shows the host name and current number of connections for each database element. See "TimesTen Scaleout element ttStats report" on page 5-157 for more information on the contents of an elements ttStats report.


**TimesTen Scaleout load profile** Displays various database metric rates. This gives you an idea of the workload, showing the rate of checkpoints, log buffer waits (delays when the log buffer fills and flushes to disk), inserts, updates, deletes, parses (such as for prepares), commits, and rollbacks. Consider whether there may be too many parses or too many durable commits (which are more expensive than non-durable commits).

**Operating system metrics summary** Displays various operating system metrics for the TimesTen Scaleout database. These metrics show the used disk space, CPU, I/O rate, and RAM.

**TimesTen Scaleout efficiency metrics** Displays various metrics that determine the efficiency of the TimesTen Scaleout. This section of the report includes the following sections:

- Target 100% - bigger is better: This shows you recommendations to improve the efficiency of the TimesTen Scaleout. It includes the following metrics:
  - Prepare exec efficiency: This shows if your SQL statements are prepared and then executed many times. If you prepare a SQL statement once for every execution, this metric goes down.
Try to minimize the number of times your SQL statements are prepared because preparing SQL statements is CPU intensive. In your applications, consider using bind variables. You can then prepare your SQL statements once and then execute your SQL statements multiple times.

- **Target 0% - smaller is better**: This shows you recommendations to improve the efficiency of the TimesTen Scaleout. It includes the following metrics:
  - **Log buffer waits**: This shows the number of log buffer waits which helps you determine how operations that use the log files are doing. It is optimal to maintain the log buffer wait low because it indicates that transactions do not need to wait before writing to the log buffer. If this number is large, try to checkpoint more frequently, increase the Log Buffer Size and/or increase the log buffer parallelism.
  - **Table data skew deviation**: This shows the percentage of table data skew deviations between the elements of the TimesTen Scaleout. Ideally the rows in tables are distributed evenly across all elements. If elements have too many rows compared to other elements, the elements with more rows use more permSize which can cause disk size and data distribution problems.
  - **Direct mode connection distribution deviation**: This shows the percentage of the direct mode connection deviation between the elements of the TimesTen Scaleout. Evenly spread the direct mode connections between the elements to achieve optimal throughput and latency.
  - **Client server connection distribution deviation**: This shows the percentage of the client/server connection deviation between the elements of the TimesTen Scaleout. Evenly spread the client/server connections between the elements to achieve optimal throughput and latency.
  - **SQL statement distribution deviation**: This shows the percentage of the SQL statement distribution deviation between the elements of the TimesTen Scaleout. It is not optimal to execute all SQL statements on a single attempt. Evenly execute the SQL statements on the elements to achieve optimal throughput and latency.
  - **Grid channel invalidation**: This shows the number of channel invalidations between the elements of the TimesTen Scaleout. Applications should cleanly disconnect and release resources to minimize channel invalidations. The cleanup process that TimesTen Scaleout performs after a channel invalidation takes time, which affects the latency and scalability of operations that want to use that channel.

**TimesTen Scaleout transactions** Displays various metrics that show information about transactions on the TimesTen Scaleout. This section of the report includes the following sections:

- **TimesTen Scaleout transactions per second**
  This table shows various transaction metrics for each element such as the rate of transactions that: only involve the local element, multiple elements, and require 2PC.

- **TimesTen Scaleout 2PC transactions**
  This table shows various 2PC transaction metrics for each element such as the percentage of transactions that: started on this element, were involved in a 2PC transaction but did not initiate it, and used durable 2PC prepares.

- **TimesTen Scaleout Durable Commits**
This table shows various transaction metrics for each element such as the percentage of transactions that are committed durably on this element.

**SQL statements: SQL statement protocol** Displays an excerpt of SQL statement protocol statistics from the SQL Statements section of a report. These statistics show you the percentage of SQL statements: executed for that element, executed on their local element, that required execution on a remote element, and that required a broadcast to all elements to execute.

**SQL statements: SQL statements type** Displays an excerpt of SQL statement type statistics from the SQL Statements section of a report. These statistics show you various statistics of SQL statements executed for that element.

**DB connections** Displays various connection statistics for each element of the TimesTen Scaleout. These statistics show you the type of connections, connections and disconnections per minute, and client server failover for every element of the TimesTen Scaleout.

**TimesTen Scaleout data distribution: table data skew - worst three tables** Displays the three tables with the highest data skew percentage of the TimesTen Scaleout. For more information on the row distribution table, see "TimesTen Scaleout data distribution: row distribution for table" on page 5-156. These statistics show you the percentage of deviation, the table distribution type, and the distribution keys for the three tables with the highest data skew percentage.

**TimesTen Scaleout data distribution: TimesTen Scaleout PermSize usage** Displays statistics related to the PermSize attribute for each element of the TimesTen Scaleout. These statistics show you the proportional percentage of used PermSize for each element in the TimesTen Scaleout, percentage of used PermSize, the percentage of used high water of PermSize, and the size of the PermSize for each element of the TimesTen Scaleout. For more information about the PermSize attribute, see "PermSize" on page 2-48.

**TimesTen Scaleout data distribution: TimesTen Scaleout TempSize usage** Displays statistics related to the TempSize attribute for each element of the TimesTen Scaleout. These statistics show you the percentage of used TempSize, the percentage of used high water of TempSize, and the size of the TempSize for each element of the TimesTen Scaleout. For more information about the TempSize attribute, see "TempSize" on page 2-50.

**TimesTen Scaleout data distribution: row distribution for table** There are three of these tables in your ttStats report, which show row distribution statistics for the tables with the highest data skew percentage of the TimesTen Scaleout. These statistics show you the number of rows that are stored on each element for that specific table. For more information about the three tables with the highest data skew percentage of the TimesTen Scaleout, see "TimesTen Scaleout data distribution: table data skew - worst three tables" on page 5-156.

**TimesTen Scaleout channel: TimesTen Scaleout messages per second** Displays statistics related to message rates over TimesTen Scaleout channels. These messages can be requests for data or data result sets. These statistics show you the number of sent, received, and invalidated messages for each element of the TimesTen Scaleout.

**TimesTen Scaleout channel: TimesTen Scaleout channel data rate** Displays statistics related to channel data rates for each element of the TimesTen Scaleout. These
data rates are based on the size of messages that TimesTen Scaleout sends over the channels. The messages can be requests for data or data result sets. Larger messages tend to have better throughput than smaller messages. These statistics show you the data rates of sent and received messages for each element of the TimesTen Scaleout.

**Checkpoint: checkpoint data rate** Displays statistics related to checkpoint data rates for each element of the TimesTen Scaleout database.

**Transaction Log: transaction log data rate** Displays statistics related to transaction log data rates for each element of the TimesTen Scaleout database.

**Top SQL: top SQL attributes** Displays statistics related to the attributes of the most executed SQL statements on the TimesTen Scaleout database.

**Top SQL: top SQL text** Displays information related to the SQL text of the most executed SQL statements on the TimesTen Scaleout database.

**OS Disk Space** Displays the statistics related to the disk space of every element that is part of the grid.

**CPU Utilization** Displays the statistics related to the CPU of every element that is part of the grid.

**Operating System IO Rates: disk IO rates** Displays the statistics related to the disk I/O rates of every element that is part of the grid. This table enables you to determine if a host has a lot of disk I/O operations that are unrelated to TimesTen Scaleout.

**Operating System IO Rates: network IO rates** Displays the statistics related to the network I/O rates of every element that is part of the grid. This table enables you to determine if a host has a lot of network I/O operations that are unrelated to TimesTen Scaleout.

**OS Memory** Displays the statistics related to the memory usage of every element that is part of the grid. This table enables you to determine if a host is using swap space or is having performance issues due to a lack of available RAM.

**TimesTen Scaleout element ttStats report**
The following sections show excerpts from tables of metrics for a ttStats report for an element of a TimesTen Scaleout database.

**Snapshot Summary** Displays the statistics related to the snapshots that ttStats uses to create the ttStats report for the element.

**Element Summary** Displays a summary of critical statistics for the element. The summary includes statistics of your element related to transaction rates, SQL statements, database connections, checkpoint rates, transaction log rates, and other critical statistics.

**Load Profile** Displays various database metric rates. This gives you an idea of the workload, showing the rate of checkpoints, log buffer waits (delays when the log buffer fills and flushes to disk), inserts, updates, deletes, parses (such as for prepares), commits, and rollbacks. Consider whether there may be too many parses or too many durable commits (which are more expensive than non-durable commits).
Operating System Metrics Summary  Displays various operating system metrics for the element. These metrics show the used disk space, CPU, I/O rate, and RAM.

Efficiency Metrics  Displays various metrics that determine the efficiency of the element. This section of the report includes the following sections:

- Target 100% - bigger is better: This shows you recommendations to improve the efficiency of the element. It includes the following metrics:
  - Prepare exec efficiency: This shows if your SQL statements are prepared and then executed many times. If you prepare a SQL statement once for every execution, this metric goes down.

  Try to minimize the number of times your SQL statements are prepared because preparing SQL statements is CPU intensive. In your applications, consider using bind variables. You can then prepare your SQL statements once and then execute your SQL statements multiple times.

- Target 0% - smaller is better: This shows you recommendations to improve the efficiency of the element. It includes the following metrics:
  - Log buffer waits: This shows the percentage of log buffer waits which helps you determine how operations that use the log files are doing. It is optimal to maintain the log buffer wait low because it indicates that transactions do not need to wait before writing to the log buffer.

    If this percentage is high, try to checkpoint more frequently, increase the Log Buffer Size and/or increase the log buffer parallelism

  - Table data skew deviation: This shows the percentage of table data skew deviations between the elements of the TimesTen Scaleout. Ideally the rows in tables are distributed evenly across all elements. If elements have too many rows compared to other elements, the elements with more rows use more permSize which can cause disk size and data distribution problems.

  - Direct mode connection distribution deviation: This shows the percentage of the direct mode connection deviation between the elements of the TimesTen Scaleout. Evenly spread the direct mode connections between the elements to achieve optimal throughput and latency.

  - Client server connection distribution deviation: This shows the percentage of the client/server connection deviation between the elements of the TimesTen Scaleout. Evenly spread the client/server connections between the elements to achieve optimal throughput and latency.

  - SQL statement distribution deviation: This shows the percentage of the SQL statement distribution deviation between the elements of the TimesTen Scaleout. It is not optimal to execute all SQL statements on a single attempt. Evenly execute the SQL statements on the elements to achieve optimal throughput and latency.

  - Grid channel invalidation: This shows the percentage of channel invalidations between the elements of the TimesTen Scaleout. Applications should cleanly disconnect and release resources to minimize channel invalidations. The cleanup process that the TimesTen Scaleout performs after a channel invalidation takes time, which affects the latency and scalability of operations that want to use that channel.

Transactions  Displays various metrics that show information about transactions on the element. This section of the report includes the following sections:
- **Transaction Type**
  
  This table shows various transaction metrics for your element such as the percentage of transactions that: only involve the local element, use remote transactions, and require 2PC.

- **2PC transactions**
  
  This table shows various 2PC transaction metrics for each element such as the percentage of transactions that: started on this element, were involved in a 2PC transaction but did not initiate it, and used durable 2PC prepares.

**SQL statements: SQL statement protocol** Displays an excerpt of SQL statement protocol statistics for the element from the SQL Statements section of a report. These statistics show you the percentage of SQL statements: executed, executed locally, that required execution on a remote element, and that required a broadcast to all elements to execute.

**SQL statements: SQL statement type** Displays an excerpt of SQL statement type statistics for the element from the SQL Statements section of a report. These statistics show you various statistics for SQL statements executed on your element.

**Database connections** Displays various connection statistics for the element. These statistics show you the type of connections, connections and disconnections per minute, and client server failover for your element.

**Table data skew** Displays the three tables with the highest data skew percentage of the element. For more information on the row distribution table, see "TimesTen Scaleout data distribution: row distribution for table" on page 5-156. These statistics show you the percentage of deviation, the table distribution type, and the distribution keys for the three tables with the highest data skew percentage.

**Grid channel usage** Displays message statistics over grid channels. These messages can be requests for data or data result sets. These statistics show you the number of sent, received, and invalidated messages for the element.

**Log holds** Displays log hold information from a report. It shows bookmark positions for checkpoint log holds for each checkpoint file. This report may also show log hold information for backup, XLA, and long-running transactions. Where the begin and end values are the same, there have been no movements.

Ideally there will be evidence of a smooth progression through the log file. (The ttStats monitor information may be more useful in tracking this.)

**Checkpoint usage** Displays checkpoint usage metrics from a report.

**Transaction log usage** Displays transaction log usage statistics for the element. This provides information about the rate of I/O operations for the transaction log, log buffer waits, log file reads, and log reads for commits.

**Top SQL: top SQL attributes** Displays statistics related to the attributes of the most executed SQL statements on the element.

**Top SQL: top SQL text** Displays information related to the SQL text of the most executed SQL statements on the element.
See also

- `ttStatsConfig`
- `ttStatsConfigGet`
ttStatus

Description
Displays information that describes the current state of TimesTen. The command displays:

- State of the TimesTen daemon process and all subdaemon processes.
- Names of all existing TimesTen databases.
- Number of connections currently connected to each TimesTen database.
- The RAM, cache agent and replication policies.
- TimesTen cache agent status.
- The status of PL/SQL.
- The key and address of the shared memory segment used by TimesTen.
- The address, key and ID of the shared memory segment used by PL/SQL.
- Whether the TimesTen instance is accessible by a specified operating system group or accessible by anyone. For more details, see the daemon options in the "Managing TimesTen daemon attributes" in Oracle TimesTen In-Memory Database Operations Guide.
- Miscellaneous status information.

If you specify a connection string or DSN, ttStatus outputs only the information for the specified database.

Required privilege
This utility requires no privileges.

Syntax

```
ttStatus {-h | -help | -?}  
ttStatus {-V | -version}  
ttStatus [-v] [-r secs] [-[no]pretty] [-gridbrief]  
ttStatus [-r secs] [-[no]pretty] (DSN | -connStr connection_string)
```

Options

ttStatus has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h</td>
<td>Prints usage information and exits.</td>
</tr>
<tr>
<td>-help</td>
<td></td>
</tr>
<tr>
<td>-?</td>
<td></td>
</tr>
<tr>
<td>-V</td>
<td>-version</td>
</tr>
<tr>
<td>-connStr connection_string</td>
<td>An ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.</td>
</tr>
<tr>
<td>DSN</td>
<td>An ODBC data source name of the database for which to get status.</td>
</tr>
</tbody>
</table>
When you call the procedure, a report that describes the current state of the system is displayed to stdout. To get the status for the cachedb1_18110 DSN:

```
ttstatus cachedb1_18110
```

TimesTen status report as of Thu May 02 19:45:43 2013
Daemon pid 5280 port 53392 instance tt1811
TimesTen server pid 3940 started on port 53393

```
Data store cachedb1_18110
There are 12 connections to the data store
Shared Memory KEY Global\cachedb1_18110.c| . . HANDLE 0x254
PL/SQL Memory KEY Global\cachedb1_18110.c| . . HANDLE 0x258 Address 0x5B8C0000

<table>
<thead>
<tr>
<th>Type</th>
<th>PID</th>
<th>Context</th>
<th>Connection Name</th>
<th>ConnID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process</td>
<td>5196</td>
<td>0x01066a58</td>
<td>cachedb1_18110</td>
<td>1</td>
</tr>
<tr>
<td>Subdaemon</td>
<td>3912</td>
<td>0x00b2c398</td>
<td>Manager</td>
<td>2047</td>
</tr>
<tr>
<td>Subdaemon</td>
<td>3912</td>
<td>0x00b7e4a0</td>
<td>Rollback</td>
<td>2046</td>
</tr>
<tr>
<td>Subdaemon</td>
<td>3912</td>
<td>0x015d25e8</td>
<td>Flusher</td>
<td>2045</td>
</tr>
<tr>
<td>Subdaemon</td>
<td>3912</td>
<td>0x015e46b0</td>
<td>Monitor</td>
<td>2044</td>
</tr>
<tr>
<td>Subdaemon</td>
<td>3912</td>
<td>0x016767f8</td>
<td>Deadlock Detector</td>
<td>2043</td>
</tr>
<tr>
<td>Subdaemon</td>
<td>3912</td>
<td>0x016888c0</td>
<td>Checkpoint</td>
<td>2041</td>
</tr>
<tr>
<td>Subdaemon</td>
<td>3912</td>
<td>0x0d350578</td>
<td>Aging</td>
<td>2042</td>
</tr>
<tr>
<td>Subdaemon</td>
<td>3912</td>
<td>0x0d362640</td>
<td>Log Marker</td>
<td>2040</td>
</tr>
<tr>
<td>Subdaemon</td>
<td>3912</td>
<td>0x0d347c8</td>
<td>AsyncMV</td>
<td>2039</td>
</tr>
<tr>
<td>Subdaemon</td>
<td>3912</td>
<td>0x0d446890</td>
<td>HistGC</td>
<td>2038</td>
</tr>
<tr>
<td>Subdaemon</td>
<td>3912</td>
<td>0x0d458958</td>
<td>IndexGC</td>
<td>2037</td>
</tr>
</tbody>
</table>
```

Replication policy : Manual
Cache Agent policy : Manual
PL/SQL enabled.

```
Accessible by group . . .
End of report
```

Show a summary of the (only) TimesTen Scaleout database on the current instance:

```
% ttstatus -gridbrief
TimesTen status report as of Thu Feb  7 11:57:32 2019

Daemon pid 12767 port 6624 instance grid1_mgmt
TimesTen server pid 12819 started on port 6625
Grid GUID: C19E1FE5-2E58-4C3F-A66A-81661763B65F
Grid Membership Service is zookeeper
Grid Membership:
  6 members
```
Notes

While primarily intended for use by TimesTen customer support, this information may be useful to system administrators and developers.

This utility is supported only where the TimesTen Data Manager is installed.

The `ttStatus` utility only reports the RAM policy if it is not `inUse`.

See also

`ttAdmin`
ttSyslogCheck (UNIX/Linux)

Description
Determines if the system’s /etc/syslog.conf file is properly configured for TimesTen. The TimesTen Data Manager uses syslog to log a variety of progress messages. It is highly desirable to configure syslog so that TimesTen writes all messages to disk in a single disk file. The ttSyslogCheck utility examines the syslog configuration (in /etc/syslog.conf) to verify that it is properly configured for TimesTen.

If syslog is properly configured, ttSyslogCheck displays the name of the file that TimesTen messages are logged to and exits with exit code 0. If syslog is not properly configured, ttSyslogCheck displays an error message and exits with code 1.

Required privilege
This utility requires no privileges.

Syntax

```
ttSyslogCheck {-h | -help | -?}
ttSyslogCheck {-V | -version}
ttSyslogCheck [-facility name]
```

Options

```
ttSyslogCheck has the options:
```

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h</td>
<td>Prints a usage message and exits.</td>
</tr>
<tr>
<td>-help</td>
<td></td>
</tr>
<tr>
<td>-?</td>
<td></td>
</tr>
<tr>
<td>-facility name</td>
<td>Specifies the syslog facility name being used for message logging.</td>
</tr>
<tr>
<td>-V</td>
<td>-version</td>
</tr>
</tbody>
</table>

Notes
This utility is supported only where the TimesTen Data Manager is installed.
ttTail

Description
Fetches TimesTen internal trace information from a database and displays it to stdout. By default, TimesTen generates no tracing information. See "ttTraceMon" on page 5-166 for more information.

Required privilege
This utility requires the ADMIN privilege.

Syntax
```
ttTail { -h | -help | -? }
ttTail { -V | -version }
ttTail [-f] { -connStr connection_string | DSN }
```

Options
The ttTail utility supports the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-connStr connection_string</td>
<td>An ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.</td>
</tr>
<tr>
<td>DSN</td>
<td>Indicates the ODBC data source name of the database from which to get a trace.</td>
</tr>
<tr>
<td>-f</td>
<td>When the end of the trace is reached, ttTail does not terminate but continues to execute, periodically polling the database's trace buffer to retrieve and display additional TimesTen trace records. For example, this is useful for generating a display of trace data that is updated in real time.</td>
</tr>
<tr>
<td>-h-help</td>
<td>Prints a usage message and exits.</td>
</tr>
<tr>
<td>-?</td>
<td></td>
</tr>
<tr>
<td>-V</td>
<td>-version</td>
</tr>
</tbody>
</table>

Examples
```
ttTail MyDatastore
```

Notes
While primarily intended for use by TimesTen customer support, this information may be useful to system administrators and developers.

This utility is supported only where the TimesTen Data Manager is installed.
ttTraceMon

Description

The ttTraceMon utility lets you enable and disable the TimesTen internal tracing facilities.

Tracing options can be enabled and disabled separately for each database. Each database contains a trace buffer into which messages describing TimesTen internal operations can be written. By default, tracing is disabled. However, it can be enabled using this utility.

The ttTraceMon utility provides subcommands to enable, disable, dump and manipulate trace information. ttTraceMon can be executed interactively (multiple subcommands can be entered at a prompt) or not interactively (one subcommand can be specified on the ttTraceMon command line).

When executed interactively, ttTraceMon prompts for lines of text from standard input and interprets the lines as trace commands. You can provide multiple trace commands on the same line by separating them with semicolons. To exit ttTraceMon, enter a blank line.

In interactive mode, you can redirect ttTraceMon command output to a file:

```
 ttTraceMon connection_string >filename
```

Component names are case-insensitive. Some commands (dump, show and flush) allow you to list many components and operate on each one. For each subcommand, if you do not list components, the utility operates on all components.

For a description of the components available through this utility and a description of the information that ttTraceMon returns for each, see "Using the ttTraceMon utility" in Oracle TimesTen In-Memory Database Troubleshooting Guide.

Required privilege

This utility requires the ADMIN privilege.

Syntax

```
 ttTraceMon {-h | -help | -?}
 ttTraceMon {-V | -version}
 ttTraceMon [-e subcommand] {-connStr connection_string | DSN}
```

Options

ttTraceMon has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-connStr connection_string</td>
<td>An ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.</td>
</tr>
<tr>
<td>DSN</td>
<td>Indicates the ODBC data source name of the database from which to get trace information.</td>
</tr>
</tbody>
</table>
-e subcommand

Causes the subcommand to be executed against the specified database. If the subcommand consists of more than one word, enclose it in double quotes. For example:

```
traceMon -e "show err" database1
```

Once the subcommand is complete, `traceMon` exits. If `-e` is not specified, `traceMon` starts in interactive mode, reading commands from stdin and displaying results to stdout.

-h

Prints a usage message and exits.

-help

-?

Prints the release number of `ttraceMon` and exits.

### Subcommands

`ttraceMon` can be called with the following subcommands:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>components</td>
<td>List the names and internal identifiers of all components.</td>
</tr>
<tr>
<td></td>
<td>For a description of the components available through this utility and a description of the information that <code>ttraceMon</code> returns for each, see &quot;Using the <code>ttraceMon</code> utility&quot; in Oracle TimesTen In-Memory Database Troubleshooting Guide.</td>
</tr>
<tr>
<td>connection {all</td>
<td>self</td>
</tr>
<tr>
<td>dump</td>
<td>Prints all trace records currently buffered. Requires <code>SELECT</code> privileges or database object ownership.</td>
</tr>
<tr>
<td>dump comp</td>
<td>Prints all trace records for component <code>comp</code>. Requires <code>SELECT</code> privileges or database object ownership.</td>
</tr>
<tr>
<td>flush</td>
<td>Discards all buffered trace records.</td>
</tr>
<tr>
<td>flush comp</td>
<td>Discards all buffered trace records for component <code>comp</code>.</td>
</tr>
<tr>
<td>help</td>
<td>Prints a summary of the trace commands.</td>
</tr>
<tr>
<td>level comp n</td>
<td>Sets the trace level for component <code>comp</code> to <code>n</code>. Requires <code>ADMIN</code> privileges or database object ownership.</td>
</tr>
<tr>
<td>outfile file</td>
<td>Prints trace output to the specified file. The file may be any of <code>0</code>, <code>stdout</code>, <code>stderr</code>, or a file name. On Windows, the file name must be in short 8.3 format. Printing is turned off when file is 0. TimesTen continues to buffer traces as usual, and they are accessible through other utilities like <code>ttTail</code>. If no file is specified, prints the current outfile setting.</td>
</tr>
<tr>
<td>show</td>
<td>Shows all the trace levels in force.</td>
</tr>
<tr>
<td>show comp</td>
<td>Shows the trace level for component <code>comp</code>.</td>
</tr>
<tr>
<td>tracefiles n</td>
<td>Sets the maximum number of output files.</td>
</tr>
</tbody>
</table>
Notes

Because tracing can degrade performance significantly, we recommend that you enable tracing only to debug problems. While primarily intended for use by TimesTen customer support, this information may be useful to system administrators and developers.

This utility is supported only where the TimesTen Data Manager is installed.
**ttUser**

**Description**

Prompts for a password and returns an encrypted password. You can then include the output in a connection string or as the value for the PWDCrypt connection attribute in an ODBCINI file.

**Required privilege**

This utility requires no privileges.

**Syntax**

```
ttUser {-h | -help | -?}
ttUser {-V | -version}
ttUser -pwdcrypt
```

**Options**

The `ttuser` utility supports the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h</td>
<td>Prints a usage message and exits.</td>
</tr>
<tr>
<td>-help</td>
<td></td>
</tr>
<tr>
<td>-?</td>
<td></td>
</tr>
<tr>
<td>-pwdcrypt</td>
<td>Generates an encrypted password value for the PWDCrypt connection attribute.</td>
</tr>
<tr>
<td>-V</td>
<td>-version</td>
</tr>
</tbody>
</table>
ttVersion

Description

The `ttVersion` utility lists the TimesTen release information, including: number, platform, instance name, instance administrator, instance home directory, daemon home directory, port number and build timestamp. You can specify various levels of output:

- You can specify `ttVersion` with no options to list abbreviated output.
- You can specify the `-m` option to list enhanced output.
- You can specify an attribute to list output only for a specific attribute.

Required privilege

This utility requires no privileges.

Syntax

```
ttVersion [-m] [attribute] [...]```

Options

`ttVersion` has the option:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-m</code></td>
<td>Generates computer-readable enhanced output. If not specified and no attribute is specified, abbreviated information is output.</td>
</tr>
<tr>
<td><code>attribute</code></td>
<td>Generates information only about the specified attribute. You can specify multiple attributes. When you specify more than one attribute, the output is displayed with an equal sign after the attribute name.</td>
</tr>
</tbody>
</table>

Attributes

`ttVersion` has these attributes:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>patched</td>
<td>Lists <code>yes</code> or <code>no</code>, indicating whether the release has been patched.</td>
</tr>
<tr>
<td>config_found</td>
<td>Lists <code>yes</code> or <code>no</code>, indicating whether the configuration file is found.</td>
</tr>
<tr>
<td>product</td>
<td>Lists the name of the product.</td>
</tr>
<tr>
<td>major1</td>
<td>The first part of the five-part release number (18 for release 18.1.2.1.0). A change in <code>major1</code> indicates major infrastructure and functionality changes.</td>
</tr>
<tr>
<td>major2</td>
<td>The second part of the five-part release number (1 for release 18.1.2.1.0). A change in only <code>major2</code> indicates a version with new functionality changes, but no infrastructure changes.</td>
</tr>
<tr>
<td>Attribute</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>patchset</td>
<td>The third part of the five-part release number (2 for release 18.1.2.1.0).</td>
</tr>
<tr>
<td></td>
<td>A change in only patchset indicates a release that contains all bug fixes since the previous maintenance release.</td>
</tr>
<tr>
<td>patch</td>
<td>The fourth part of the five-part release number (1 for release 18.1.2.1.0).</td>
</tr>
<tr>
<td></td>
<td>A change in only patch indicates a release with minor bug fixes.</td>
</tr>
<tr>
<td>reserved</td>
<td>The fifth part of the five-part release number (0 for release 18.1.2.1.0).</td>
</tr>
<tr>
<td></td>
<td>Reserved for future use.</td>
</tr>
<tr>
<td>major3</td>
<td>No longer needed, but maintained for backward compatibility. Same as patchset.</td>
</tr>
<tr>
<td>portpatch</td>
<td>No longer needed, but maintained for backward compatibility. Same as reserved.</td>
</tr>
<tr>
<td>version</td>
<td>All five parts of the release number, separated by periods (such as 18.1.2.1.0).</td>
</tr>
<tr>
<td>shortversion</td>
<td>The first two parts of the five-part release number, without periods (181 for release 18.1.2.1.0).</td>
</tr>
<tr>
<td>numversion</td>
<td>All five parts of the release number, without periods, in the format %02d%02d%04d%02d%02d, compatible with ODBC version number format (180100020100 for release 18.1.2.1.0).</td>
</tr>
<tr>
<td>bits</td>
<td>Lists 64 to indicate the 64-bit-level of the operating system for which this release is intended.</td>
</tr>
<tr>
<td>os</td>
<td>The operating system for which this release is intended.</td>
</tr>
<tr>
<td>buildstamp</td>
<td>A number indicating the specific build.</td>
</tr>
<tr>
<td>buildtime</td>
<td>The UTC time the release was built, for example: 2013-03-19T17:21:59Z</td>
</tr>
<tr>
<td>clientonly</td>
<td>Lists yes or no to indicate if the release is a client-only release.</td>
</tr>
<tr>
<td>instance</td>
<td>The name of the instance, for example: tt1811.</td>
</tr>
<tr>
<td>effective_port</td>
<td>The number of the port on which the main daemon listens.</td>
</tr>
<tr>
<td>orig_port</td>
<td>The original number of the port on which the main daemon listened.</td>
</tr>
<tr>
<td>instance_admin</td>
<td>The user name of the instance administrator.</td>
</tr>
<tr>
<td>effective_insthome</td>
<td>The path that indicates the location of the instance.</td>
</tr>
<tr>
<td>orig_insthome</td>
<td>The path that indicates the location of the instance.</td>
</tr>
<tr>
<td>effective_daemonhome</td>
<td>The path to the home of the daemon for the specific instance.</td>
</tr>
<tr>
<td>effective_daemonhome_long</td>
<td>On Windows, the path to the home of the daemon for the specific instance, including a bit extension on the instance name.</td>
</tr>
<tr>
<td>orig_daemonhome</td>
<td>The path to the original home of the daemon.</td>
</tr>
<tr>
<td>plsql</td>
<td>Indicates if PL/SQL is configured for this instance. 0 indicates that PL/SQL is not configured. 1 indicates that PL/SQL is configured. The value corresponds with the setting of the PLSQL connection attribute.</td>
</tr>
<tr>
<td>grid</td>
<td>Indicates if the instance is configured for grid distribution.</td>
</tr>
<tr>
<td>group_name</td>
<td>The name of the instance group.</td>
</tr>
</tbody>
</table>
Following is `ttVersion` output without the `-m` option:

TimesTen Release 18.1.2.1.0 (64 bit Linux/x86_64) (grid1_mgmt:6624)
2019-01-15T06:25:33Z
   Instance admin: ttuser1
   Instance home directory: /sw/ttinstances/grid1_mgmt
   Group owner: timesten
   Daemon home directory: /sw/ttinstances/grid1_mgmt/info
   PL/SQL enabled.

And with the `-m` option:

```
patched=yes
config_found=yes
product=TimesTen
major1=18
major2=1
patchset=2
patch=1
reserved=0
major3=2
portpatch=0
version=18.1.2.1.0
shortversion=181
numversion=180100020100
bits=64
os=Linux/x86_64
buildstamp=1547533533
builtime=2019-01-15T06:25:33Z
clientonly=no
instance=grid1_mgmt
effective_port=6624
orig_port=6624
instance_admin=ttuser1
effective_insthome=/sw/ttinstances/grid1_mgmt
orig_insthome=/sw/ttinstances/grid1_mgmt
effective_daemonhome=/sw/ttinstances/grid1_mgmt/info
orig_daemonhome=/sw/ttinstances/grid1_mgmt/info
plsql=1
grid=0
```

group_name=timesten
**Description**

The `ttXactAdmin` utility lists ownership, status, log and lock information for each outstanding transaction. The `ttXactAdmin` utility also enables you to heuristically commit, terminate or forget an XA transaction branch.

Applications should monitor log holds and the accumulation of log files. For more information, see "Monitoring accumulation of transaction log files" in the *Oracle TimesTen In-Memory Database Operations Guide*.

**Required privilege**

This utility requires various privileges depending on which options are entered on the command line. See the description of the options to determine which privilege is needed, if any.

**Syntax**

```
ttXactAdmin {-h | -help | -?}  
ttXactAdmin {-V | -version}  

ttXactAdmin [-v verbosity] [-mt maxTrans] [-ml maxLocks] [-pid pid]  
[-xact xid] [-tbl [owner.]tableName] [-interval seconds]  
[-count iterations] [DSN | -connstr connection_string]  

ttXactAdmin -latch [-interval seconds] [-count iterations]  
(DSN | -connstr connStr)  

ttXactAdmin -connections [-pid pid] [-interval seconds]  
[-count iterations] [DSN | -connstr connStr]  

ttXactAdmin -xactIdRollback xid (DSN | -connstr connStr)  

ttXactAdmin -XactIdCommit xid  

```

**Options**

`ttXactAdmin` has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-connections</code></td>
<td>Shows all current connections to the database. When run with the <code>-connections</code> option, <code>ttXactAdmin</code> itself does not establish a true connection to the database, and requires no latches. This can be useful when diagnosing frozen systems. This option requires <code>ADMIN</code> privileges.</td>
</tr>
<tr>
<td><code>-connStr connection_string</code></td>
<td>An ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.</td>
</tr>
<tr>
<td><code>-count iterations</code></td>
<td>Generate the report iterations times. If no <code>-interval</code> option is specified, an interval of 1 second is used.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>DSN</strong></td>
<td>Indicates the ODBC data source name of the database to be administered.</td>
</tr>
<tr>
<td></td>
<td>This option requires <strong>ADMIN</strong> privileges.</td>
</tr>
<tr>
<td><strong>-h</strong></td>
<td>Prints a usage message and exits.</td>
</tr>
<tr>
<td><strong>-help</strong></td>
<td></td>
</tr>
<tr>
<td><strong>-?</strong></td>
<td></td>
</tr>
<tr>
<td><strong>-HAbort xid</strong></td>
<td>Available for TimesTen Scaleout. Heuristically terminates an XA transaction branch in TimesTen. The specified transaction ID must be the local TimesTen TransID. This option requires <strong>ADMIN</strong> privileges or ownership of the specified transactions. This option is not supported in TimesTen Scaleout.</td>
</tr>
<tr>
<td><strong>-HCommit xid</strong></td>
<td>Available for TimesTen Scaleout. Heuristically commit an XA transaction branch in TimesTen. The specified transaction ID must be the local TimesTen TransID. This option requires <strong>ADMIN</strong> privileges or ownership of the specified transactions. This option is not supported in TimesTen Scaleout.</td>
</tr>
<tr>
<td><strong>-HForget xid</strong></td>
<td>Available for TimesTen Scaleout. Heuristically forget an XA transaction branch in TimesTen. The specified transaction ID must be the local TimesTen TransID. This option requires <strong>ADMIN</strong> privileges or ownership of the specified transactions. This option is not supported in TimesTen Scaleout.</td>
</tr>
<tr>
<td><strong>-interval seconds</strong></td>
<td>Repeat the generation of the report, pausing the indicated number of seconds between each generation. If no <strong>-count</strong> option is specified, repeat forever.</td>
</tr>
<tr>
<td><strong>-latch</strong></td>
<td>This option is to be used by TimesTen Customer Support only. Shows only the latch information for the database specified.</td>
</tr>
<tr>
<td><strong>-ml maxLocks</strong></td>
<td>Maximum number of locks per transaction. Default is 6000.</td>
</tr>
<tr>
<td><strong>-mt maxTrans</strong></td>
<td>Specifies the maximum number of transactions to be displayed. The default is all outstanding transactions.</td>
</tr>
<tr>
<td><strong>-pid pid</strong></td>
<td>Displays only transactions started by the process with the specified <strong>pid</strong>. On Linux, it is the pid of the thread that opens the connection. This option requires <strong>ADMIN</strong> privileges or ownership of the specified transactions.</td>
</tr>
<tr>
<td><strong>-tbl [owner.]tableName</strong></td>
<td>Displays lock information for the specified table. This option requires <strong>ADMIN</strong> privileges or ownership of the specified table.</td>
</tr>
<tr>
<td>**-V</td>
<td>-version**</td>
</tr>
<tr>
<td><strong>-v verbosity</strong></td>
<td>Specifies the verbosity level. One of:</td>
</tr>
<tr>
<td></td>
<td>0 - Does not display the names of the tables for row locks. In this case, <strong>ttXactAdmin</strong> runs faster.</td>
</tr>
<tr>
<td></td>
<td>1 (default) - Displays the names of the tables for row locks.</td>
</tr>
</tbody>
</table>
ttXactAdmin produces the following output:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program File Name</td>
<td>The executable file name of the process that owns the transaction.</td>
</tr>
<tr>
<td>PID</td>
<td>The process ID of the application that owns the transaction. On Linux, the PID of the thread that opens the connection.</td>
</tr>
<tr>
<td>Context</td>
<td>The internal identifier that distinguishes between multiple connections to the database made by a single multithreaded process.</td>
</tr>
</tbody>
</table>
The unique identifier for the transaction used internally by TimesTen. For TimesTen Classic, the identifier has two parts. For TimesTen Scaleout, the identifier is prefixed by the Element ID. The first part is a relatively small value (less than 2048), used to identify the connection of the program executing the transaction. The second part is a potentially large value (an unsigned integer), that distinguishes between successive uses of the same first part. (The value wraps around if necessary.) Thus, identifiers 4.100 and 4.200 cannot be present at the same time. If 4.100 is seen, and then 4.200, this indicates that transaction 4.100 has completed (committed or rolled back).

Current state of the transaction, one of:

- **Active** - Active transaction.
- **Abort**ing - A transaction is in the process of terminating. See Notes for more information.
- **Commit**ing - Committing transaction, locks are being released.
- **Ckpoint**ing - A transaction doing checkpoint.
- **Idle** - A transaction branch currently not accessing data.
- **Prepared** - Prepared transaction branch.
- Heur-Committed - Heuristically committed transaction branch.
- Heur-Aborted - Heuristically terminated transaction branch.
- Propagating - TimesTen transaction waiting for Oracle to commit.

When using TimesTen Scaleout, the current status of the transaction, one of:

- **Active** - Active transaction.
- **Abort**ing - A transaction is in the process of terminating. See Notes for more information.
- **Commit**ing - Committing transaction, locks are being released.
- **Ckpoint**ing - A transaction doing checkpoint.
- **Idle** - A transaction branch currently not accessing data.
- Grid-Doubtful-Yes - The grid transaction prepared and voted yes for commit on this element, and is now doubtful.
- Grid-Doubtful-No - The grid transaction prepared and voted no for commit on this element, and is now doubtful.
- Grid-Err - The grid transaction returned an error on this element.

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XactID</td>
<td>The unique identifier for the transaction used internally by TimesTen. For TimesTen Classic, the identifier has two parts. For TimesTen Scaleout, the identifier is prefixed by the Element ID. The first part is a relatively small value (less than 2048), used to identify the connection of the program executing the transaction. The second part is a potentially large value (an unsigned integer), that distinguishes between successive uses of the same first part. (The value wraps around if necessary.) Thus, identifiers 4.100 and 4.200 cannot be present at the same time. If 4.100 is seen, and then 4.200, this indicates that transaction 4.100 has completed (committed or rolled back).</td>
</tr>
</tbody>
</table>
### Column Description

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ResourceId</strong></td>
<td>A unique identifier of each unique resource. The identifier is displayed in hexadecimal format with a few exception. <strong>Table</strong> and <strong>CompCmd</strong> are shown as decimal values. Row locks are shown in the <strong>ROWID</strong> character format.</td>
</tr>
<tr>
<td><strong>Mode</strong></td>
<td>A value used to determine the level of concurrency that the lock provides:</td>
</tr>
<tr>
<td></td>
<td><strong>S</strong> - Shared lock in serializable isolation.</td>
</tr>
<tr>
<td></td>
<td><strong>Sn</strong> - Shared lock in non-serializable isolation.</td>
</tr>
<tr>
<td></td>
<td><strong>U</strong> - Update lock in serializable isolation.</td>
</tr>
<tr>
<td></td>
<td><strong>Un</strong> - Update lock in non-serializable isolation.</td>
</tr>
<tr>
<td></td>
<td><strong>En</strong> - End-of-scan lock for non-serializable isolation.</td>
</tr>
<tr>
<td></td>
<td><strong>IRC</strong> - Intention shared lock in non-serializable isolation.</td>
</tr>
<tr>
<td></td>
<td><strong>IS</strong> - Intention shared lock in serializable isolation.</td>
</tr>
<tr>
<td></td>
<td><strong>IU</strong> - Intention update lock in serializable isolation.</td>
</tr>
<tr>
<td></td>
<td><strong>IUn</strong> - Intention update lock in non-serializable isolation.</td>
</tr>
<tr>
<td></td>
<td><strong>IX</strong> - Intention exclusive lock in serializable isolation.</td>
</tr>
<tr>
<td></td>
<td><strong>IXn</strong> - Intention exclusive lock non-serializable isolation.</td>
</tr>
<tr>
<td></td>
<td><strong>SIX</strong> - Shared lock with intent to set an exclusive lock in serializable isolation.</td>
</tr>
<tr>
<td></td>
<td><strong>SIXn</strong> - Shared lock with intent to set an exclusive lock non-serializable isolation.</td>
</tr>
<tr>
<td></td>
<td><strong>X</strong> - Exclusive lock.</td>
</tr>
<tr>
<td></td>
<td><strong>Xn</strong> - Exclusive lock in non-serializable isolation.</td>
</tr>
<tr>
<td></td>
<td><strong>W</strong> - Update, insert or delete table lock.</td>
</tr>
<tr>
<td></td>
<td><strong>XNi</strong> - Next lock for inserting into tables or non-unique index.</td>
</tr>
<tr>
<td></td>
<td><strong>NS</strong> - Table lock in read-committed isolation that conflicts with all table locks in serializable isolation. Lock &quot;0&quot; means the blocker is still in the waiting list.</td>
</tr>
<tr>
<td><strong>HMode</strong></td>
<td>The mode in which the competing transaction is holding the lock which the waiting transaction is requesting.</td>
</tr>
<tr>
<td></td>
<td>See &quot;Mode&quot; in this table for concurrency level descriptions.</td>
</tr>
<tr>
<td><strong>RMode</strong></td>
<td>Shows the mode in which the waiting transaction has requested to hold the lock. See &quot;Mode&quot; in this table for concurrency level descriptions.</td>
</tr>
</tbody>
</table>
Examples

The following command displays all locks in the database:

ttXactAdmin -connstr DSN=demodata
2018-03-20 13:02:54.760
/timesten/jsmith/demo/demodata
TimesTen Release 18.1.1.1.0
ElementID 1

Program File Name: _ttIsql

<table>
<thead>
<tr>
<th>XactID</th>
<th>PID</th>
<th>Context</th>
<th>State</th>
<th>Loghold Last ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1.195</td>
<td>115640</td>
<td>0x859570</td>
<td>Active</td>
<td>391.15355904 [1666839:6]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource</th>
<th>ResourceID</th>
<th>Mode</th>
<th>SqlCmdID</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>0x01312d0001312d00</td>
<td>IX</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Table</td>
<td>2367528</td>
<td>IXn</td>
<td>275642480</td>
<td>JSMITH.T</td>
</tr>
<tr>
<td>Row</td>
<td>AAAVVUAADXQAANje</td>
<td>Xn</td>
<td>275642480</td>
<td>JSMITH.T</td>
</tr>
</tbody>
</table>

Begin Time: 13:01:43.108

1 outstanding transaction found

Notes

If the transaction specified in the command is not an XA transaction branch but a TimesTen local transaction, no XA-XID are displayed. The XA-XID is a C structure that contains a format identifier, two length fields and a data field. The data field consists of at most two contiguous components: a global transaction identifier (gtrid) and a branch qualifier (bqual). The two length fields specify the number of bytes (1-64) in gtrid and bqual respectively. For more details, refer to the X/Open publication: Distributed Transaction Processing: The XA Specification (c193).

For databases, TimesTen only holds S locks when the isolation mode is serializable. For commands, S only means “shared” lock, and can be held in either serializable or read-committed isolation modes.

Under RMode, awaiting transactions are sorted by PID and Context. The listing does not reflect the order of the lock requests.

A lock request with an RMode compatible with the HMode of the lock holder can be waiting because there is another lock request with an incompatible mode ahead of the compatible request in the lock request queue.

A transaction can have the status Aborting for one of these reasons:

- A user application requested rollback after doing a large amount of work.
- An application with autocommit tried a statement that could not be completed and it is being undone.
- Another call to ttXactAdmin caused a transaction to rollback.
- A process died with work in progress and that work is being undone.
**ttXactLog**

**Description**
Displays a formatted dump of the contents of a TimesTen transaction log. It is designed to be used by TimesTen customer support to diagnose problems in the log or database.

A loss of data can occur with certain options such as `-tr`, therefore only use this tool if you have been asked to do so by a TimesTen customer support representative.

**Required privilege**
This utility requires the `ADMIN` privilege.

**Syntax**

```
 ttXactLog (-h | -help | -?)
 ttXactLog (-V | -version)
 ttXactLog [-v verbosity] [-m maxChars] [-s] [-t] [-b blkID]
 [-l1 fn.ifo [-l2 fn.ifo]] [-r recType] [...] [-tr dir]
 [-l1b] [-headers recs] [-logdir dir]
 (-connStr connection_string | DSN | dspath)
```

```
 ttXactLog [-v verbosity] -logAnalyze
 [-s subscriberName -host hostname]
 [-xid xid] (-connStr connection_string | DSN | dspath)
```

**Options**

`ttXactLog` has the options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-b blkID</code></td>
<td>Restricts log records to those accessing this block, plus any transaction records.</td>
</tr>
<tr>
<td><code>-connStr connection_string</code></td>
<td>An ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.</td>
</tr>
<tr>
<td><code>DSN</code></td>
<td>The ODBC source name of the database for which to display the transaction log.</td>
</tr>
<tr>
<td><code>dspath</code></td>
<td>The fully qualified name of the database. This is not the DSN associated with the connection but the fully qualified database path name associated with the database as specified in the <code>DataStore=</code> parameter of the database's ODBC definition. For example, for a database consisting of files/home/payroll/2011.ds0, /home/payroll/2011.dsl and several transaction log files /home/payroll/2011.logn,dspath is /home/payroll/2011.</td>
</tr>
<tr>
<td><code>-h</code></td>
<td>Prints a usage message and exits.</td>
</tr>
<tr>
<td><code>-help</code></td>
<td></td>
</tr>
<tr>
<td><code>-?</code></td>
<td></td>
</tr>
<tr>
<td><code>-headers records</code></td>
<td>Prints one header for every records records. A value of 0 disables headers.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>-host hostName</td>
<td>Specifies the name of the host on which the subscriber resides. Use this option with the -subscriber option, if the name of the subscriber is ambiguous.</td>
</tr>
<tr>
<td>-lb</td>
<td>Connects to the database and prints out the log buffer. Contents of the transaction log files are not printed. Requires SELECT privileges or database object ownership.</td>
</tr>
<tr>
<td>lfn.1fo</td>
<td>Transaction log file number (lfn) and transaction log file offset (1fo) for a log record.</td>
</tr>
<tr>
<td>-11</td>
<td>Considers this log record only (unless an -12 argument is present).</td>
</tr>
<tr>
<td>-12</td>
<td>Considers records between -11 and -12, inclusive.</td>
</tr>
<tr>
<td>-logAnalyze</td>
<td>Determines the remaining amount of a database to be replicated for one or all of the subscribers. Use with the -v option to print: 1 - A summary for every track (default). 2 - Level 1 plus a track analysis. 3 - Level 2 plus an in-depth transaction analysis. Use with -subscriber and -host to get information for a specific subscriber. Use with -xid to look for a specific transaction ID.</td>
</tr>
<tr>
<td>-logdir dir</td>
<td>Specifies the directory where the database's transaction log files reside. If -logdir is not specified, ttXactLog uses the directory path portion of the value supplied in dspath.</td>
</tr>
<tr>
<td>-m maxChars</td>
<td>Maximum number of characters printed for binary items (for -v 3) only (defaults to 4000).</td>
</tr>
<tr>
<td>-r recType</td>
<td>Considers only records of the specified type. This option may be used multiple times to specify a list of desired log record types. recType is case-sensitive.</td>
</tr>
<tr>
<td>-s</td>
<td>Prints summary information. Requires SELECT privileges or database object ownership.</td>
</tr>
<tr>
<td>-subscriber subscriberName</td>
<td>Specifies the name of the subscriber. To qualify the name of the subscriber, use -host hostname.</td>
</tr>
<tr>
<td>-t</td>
<td>Only reads transaction log file tail (from start of last checkpoint transaction log file or, if no checkpoint, the most recent transaction log file).</td>
</tr>
<tr>
<td>-tr dir</td>
<td>Truncates all log records in the directory at the LBN boundary. The original transaction log files are moved to the directory dir.</td>
</tr>
<tr>
<td>-V</td>
<td>-version</td>
</tr>
<tr>
<td>-v verbosity</td>
<td>Specifies the verbosity level. One of: 0 - Print only summary log information (if -s specified). 1 (default) - Print log record headers too. 2 - Print log record bodies too, except long data. 3 - Print full log records (see -m option).</td>
</tr>
<tr>
<td>-x xid</td>
<td>Specifies the transaction ID.</td>
</tr>
</tbody>
</table>

**Examples**

```
ttXactLog -v 3 -m 100 /users/pat/TimesTen/Daily/F112697SS
```
Notes

This utility is supported only where the TimesTen Data Manager is installed.

See also

"Analyze outstanding transactions in the replication log" in the Oracle TimesTen In-Memory Database Replication Guide.
The following sections list all TimesTen system limits and defaults.

- System limits and defaults
- Limits on number of open files
- Path names

System limits and defaults

Specific operating system limits may take precedence over these values.

For more information, see "Operating system prerequisites" in Oracle TimesTen In-Memory Database Installation, Migration, and Upgrade Guide or "Operating system prerequisites" in Oracle TimesTen In-Memory Database Scaleout User's Guide.

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum number of subscriber databases in a replication scheme that is</td>
<td>128</td>
</tr>
<tr>
<td>not an active standby pair.</td>
<td></td>
</tr>
<tr>
<td>Maximum number of propagators in a replication scheme. Each propagator</td>
<td>128</td>
</tr>
<tr>
<td>can have the maximum number of subscribers.</td>
<td></td>
</tr>
<tr>
<td>Maximum number of subscriber databases in an active standby pair.</td>
<td>127</td>
</tr>
<tr>
<td>Minimum database size (bytes). Size includes both the permanent and</td>
<td>32 MB</td>
</tr>
<tr>
<td>temporary space required to perform operations on the database.</td>
<td></td>
</tr>
<tr>
<td>Maximum length for a fixed-length column (bytes).</td>
<td>8,300</td>
</tr>
<tr>
<td>Maximum number of columns in a table.</td>
<td>1,000</td>
</tr>
<tr>
<td>Maximum number of columns in an ORDER BY clause.</td>
<td>1,000</td>
</tr>
<tr>
<td>Maximum number of columns in a GROUP BY clause.</td>
<td>1,000</td>
</tr>
<tr>
<td>Maximum cumulative length of a row’s fixed-length columns (bytes).</td>
<td>32,768</td>
</tr>
<tr>
<td>Maximum length for a varying-length column (bytes).</td>
<td>$2^{22} = 4,194,304$</td>
</tr>
<tr>
<td>Maximum length for a replicated column.</td>
<td>4 MB</td>
</tr>
<tr>
<td>Maximum number of concurrent connections to a database (including system</td>
<td>32,047</td>
</tr>
<tr>
<td>connections).</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Value</td>
</tr>
<tr>
<td>---------------------------------------------------------------------------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>Maximum number of concurrent application connections to a database</td>
<td>32,000</td>
</tr>
<tr>
<td>(may be limited by semaphore configuration or Connections DSN attribute</td>
<td>2,000 (default)</td>
</tr>
<tr>
<td>or both).</td>
<td></td>
</tr>
<tr>
<td>Maximum number of connections (system and application) across all</td>
<td>32,048</td>
</tr>
<tr>
<td>databases in an instance.</td>
<td></td>
</tr>
<tr>
<td>Maximum number of concurrent client connections to a TimesTen instance.</td>
<td>2048</td>
</tr>
<tr>
<td><strong>Note</strong>: Some instances may support a slightly smaller maximum number of</td>
<td></td>
</tr>
<tr>
<td>connections depending on such things as whether the database is shared</td>
<td></td>
</tr>
<tr>
<td>or replicated and operating system limits. Most configurations support no</td>
<td></td>
</tr>
<tr>
<td>less than 2,000 connections.</td>
<td></td>
</tr>
<tr>
<td>Maximum length of database names.</td>
<td>32</td>
</tr>
<tr>
<td>Maximum length of the path name for a database in an asynchronous</td>
<td>248</td>
</tr>
<tr>
<td>writethrough cache group</td>
<td></td>
</tr>
<tr>
<td>Maximum number of projected expressions in a SELECT statement.</td>
<td>32,767</td>
</tr>
<tr>
<td>Maximum length of string specifying a join order.</td>
<td>1,024</td>
</tr>
<tr>
<td>Maximum number of columns in an index (or primary key).</td>
<td>32</td>
</tr>
<tr>
<td>Maximum length of basic names.</td>
<td>30</td>
</tr>
<tr>
<td>Maximum length of displayed predicate string in the SYS.PLAN table.</td>
<td>1,024</td>
</tr>
<tr>
<td>Maximum length of SQL statement, including the NULL terminator.</td>
<td>409,600</td>
</tr>
<tr>
<td>Maximum number of table references in an SQL query.</td>
<td>24</td>
</tr>
<tr>
<td>Maximum number of indexes on a table.</td>
<td>500</td>
</tr>
<tr>
<td>Maximum number of partitions in a table.</td>
<td>999</td>
</tr>
<tr>
<td>Maximum number of prepared PL/SQL statements per connection.</td>
<td>5000</td>
</tr>
<tr>
<td>Maximum number of recently-used PL/SQL blocks that can be cached per</td>
<td>5000</td>
</tr>
<tr>
<td>session.</td>
<td></td>
</tr>
<tr>
<td>Maximum number of concurrent segment client/server connections per</td>
<td>Unlimited (up to the limits of the</td>
</tr>
<tr>
<td>TimesTen instance</td>
<td>operating system)</td>
</tr>
<tr>
<td>Maximum number of concurrent shared memory segment client/server</td>
<td>Unlimited (up to the limits of the</td>
</tr>
<tr>
<td>connections per TimesTen instance.</td>
<td>operating system)</td>
</tr>
<tr>
<td>Maximum size of IPC shared memory segment for client/server</td>
<td>4 GB</td>
</tr>
<tr>
<td>connections</td>
<td></td>
</tr>
<tr>
<td>Maximum number of allocated statement handles per shared memory segment</td>
<td>512</td>
</tr>
<tr>
<td>client/server connection.</td>
<td></td>
</tr>
<tr>
<td>Maximum depth of nesting subqueries.</td>
<td>Equal to the maximum number of table</td>
</tr>
<tr>
<td>references in a SQL query.</td>
<td></td>
</tr>
<tr>
<td>Maximum error message length for applications that specify an error</td>
<td>512</td>
</tr>
<tr>
<td>message length (for example, through a call to SQLError).</td>
<td></td>
</tr>
<tr>
<td>Maximum number of replicated XLA bookmarks.</td>
<td>64</td>
</tr>
</tbody>
</table>
Limits on number of open files

Each process connected to a TimesTen database keeps at least one operating-system file descriptor open from the time of the first connection until the process terminates. Additional file descriptors may be opened for each database connection:

- Connections to databases that have logging to disk enabled require an additional two file descriptors for the duration of the connection.
- An additional file descriptor is needed for the duration of database checkpoints issued by the process.
- Additional file descriptors may be opened during transaction commit or rollback operations.

For multithreaded applications that maintain many concurrent TimesTen database connections, the default number of open files permitted to each process by the operating system may be too low.

- On Solaris, the default limit is 256 open files and may be raised for a session with the `ulimit` command (limit for `csh` users). You can also set the per-process limit programmatically with `setrlimit`.
- On AIX, the limit is 2,048 open files, so you are not likely to encounter problems.
- On Linux, the default limit is 1,024 open files, so you are not likely to encounter problems.
- On Windows, the default limit is at least 2,000 open files, so you are not likely to encounter problems.

Most of the open file descriptors are used for reading and writing database recovery log files. If a process fails to open a log file, the database is marked as requiring recovery and all current connections to the database are terminated.

Path names

TimesTen does not support file path names that contain multibyte characters. Ensure the installation path, database path, transaction log path, and temporary file path do not contain any multibyte characters.
Symbols, 2-113

A
access control
collection attributes, 2-1
utilities, 2-73, 4-1, 5-1
administration
daemon, ttDaemonAdmin utility, 5-48
database, ttAdmin utility, 5-5
AllFlags optimizer flag, 3-163
attributes—see connection attributes
authentication, utilities, 2-73
AutoCreate connection attribute, 2-27

B
backup
ttBackup utility, 5-34
ttBackupStatus built-in procedure, 3-16
BranchAndBound optimizer flag, 3-161
built-in procedures
calling, 3-1
ttAgingLRUCfg, 3-10
ttAgingScheduleNow, 3-13
ttApplicationContext, 3-15
ttBackupStatus, 3-16
ttBlockInfo, 3-18
ttBookmark, 3-19
ttCacheADGStandbyStateGet, 3-21
ttCacheADGStandbyStateSet, 3-23
ttCacheADGStandbyTimeoutGet, 3-25
ttCacheADGStandbyTimeoutSet, 3-26
ttCacheAllowFlushAwtSet, 3-28
ltCacheAutorefIntervalStatsGet, 3-30
ttCacheAutorenew, 3-33
ltCacheAutorefreshLogDefrag, 3-35
ltCacheAutorefreshSelectLimit, 3-41
ltCacheAutorefreshStatsSetGet, 3-37
ltCacheAutorefreshXactLimit, 3-43
ltCacheAWTMonitorConfig, 3-46
ltCacheAWTThresholdGet, 3-48
ltCacheAWTThresholdSet, 3-49
ltCacheCheck, 3-51

Index
attribute, 2-123
TTC_TCP_KEEPALIVE_PROBES connection
attribute, 2-124
TTC_TCP_Keepalive_Time_MS connection
attribute, 2-125
TTC_Timeout connection attribute, 2-111, 2-126
UID and PWD connection attributes, 2-73
clusterware, ttCWAdmin utility, 5-42
COL_STATS system table, 3-177
column ID, XLA, ttSetUserColumnID built-in
procedure, 3-228
columns
fixed-length column, maximum length, 6-1
maximum length, varying-length column, 6-1
maximum number in index, 6-2
maximum per table, 6-1
maximum, GROUP BY clause, 6-1
maximum, ORDER BY clause, 6-1
replicated column, maximum length, 6-1
command cache, SQL
query plans, ttSQLCmdQueryPlan built-in
procedure, 3-238
ttSQLCmdCacheInfo built-in procedure, 3-233
ttSQLCmdCacheInfoGet built-in
procedure, 3-236
ttSQLCmdQueryPlan built-in procedure, 3-238
commit
DurableCommits connection attribute, 2-59
CommitBufferSizeMax connection attribute, 2-52
compacting database
ttCompact built-in procedure, 3-91
concurrency
concurrent application connections,
maximum, 6-2
concurrent client connections, maximum, 6-2
concurrent connections, maximum, 6-1
concurrent connections, system plus applications,
maximum, 6-2
LockLevel connection attribute, 2-63
LockWait connection attribute, 2-64
configuration file attributes, 1-1
connection
concurrent application connections,
maximum, 6-2
concurrent client connections, maximum, 6-2
concurrent connections, maximum, 6-1
concurrent connections, system plus applications,
maximum, 6-2
conflicts, 2-74
ConnectionCharacterSet connection
attribute, 2-76
MaxConnsPerServer connection attribute, 2-128
PL/SQL statements, prepared, maximum, 6-2
UID and PWD connection attributes, 2-73
WaitForConnect connection attribute, 2-74
connection attributes
access control, 2-1
AutoCreate, 2-27
CacheAWTMethod, 2-95
CacheAWTParallelism, 2-98
CkptFrequency, 2-28
CkptLogVolume, 2-30
CkptRate, 2-32
CkptReadThreads, 2-34
client connection attributes, 2-8, 2-108
CommitBufferSizeMax, 2-52
ConnectionCharacterSet, 2-76
ConnectionName, 2-53
Connections, 2-35
CreateEpochAtCommit, 2-54
Data Source Name, 2-11
data store attributes, 2-2, 2-10
DatabaseCharacterSet, 2-13
DataStore, 2-12
DDLReplicationAction, 2-55
DDLReplicationLevel, 2-56
Description, 2-16
Diagnostics, 2-58
Driver, 2-17
Durability, 2-18
DurableCommits, 2-59
DynamicLoadEnable, 2-100
DynamicLoadErrorMode, 2-101
EpochInterval, 2-36
first connection attributes, 2-3, 2-26
ForceConnect, 2-37
ForceDisconnectEnabled, 2-38
general connection attributes, 2-4, 2-51
GridLogRecoveryThreshold, 2-19
IncludeInCore, 2-60
Isolation, 2-61
LockLevel, 2-63
LockWait, 2-64
LogAutoTruncate, 2-39
LogBufMB, 2-40
LogBufParallelism, 2-41
LogDir, 2-19, 2-20
LogFileSize, 2-42
LogFlushMethod, 2-43
LogPurge, 2-44
MaxConnsPerServer, 2-128
MemoryLock, 2-45
NLS general connection attributes, 2-6, 2-75
NLS_LENGTH_SEMANTICS, 2-77
NLS_NCHAR_CONV_EXCP, 2-78
NLS_SORT, 2-79
OptimizerHint, 2-65
OracleNetServiceName, 2-102
OraclePWD, 2-103
Overwrite, 2-47
PassThrough, 2-104
PermSize, 2-48
PermWarnThreshold, 2-66
PLSCOPE_SETTINGS, 2-88
PL/SQL first connection attributes, 2-6, 2-83
PL/SQL general connection attributes, 2-6, 2-87
PLSQL_CCFLAGS, 2-89
PLSQL_CONN_MEM_LIMIT, 2-90
PLSQL_MEMORY_ADDRESS, 2-84
PLSQL_MEMORY_SIZE, 2-86
PLSQL_OPTIMIZE_LEVEL, 2-91
PLSQL_SESSION_CACHED_CURSORS, 2-92
PLSQL_TIMEOUT, 2-93
Preallocate, 2-21
PrivateCommands, 2-67
privileges, 2-1
PWD, 2-73
PWDCrypt, 2-68
QueryThreshold, 2-69
RACCallback, 2-107
RecoveryThreads, 2-49
ReplicationApplyOrdering, 2-22
ReplicationParallelism, 2-24
ReplicationTrack, 2-70
server connection attributes, 2-9, 2-127
ServersPerDSN, 2-129
ServerStackSize, 2-130
SQLQueryTimeout, 2-71
TCP_Port, 2-109
TCP_Port2, 2-110
Temporary, 2-25
TempSize, 2-50
TempWarnThreshold, 2-72
TimesTen Cache database attributes, 2-7, 2-97
TimesTen Cache first connection attributes, 2-7, 2-94
TimesTen Cache general connection attributes, 2-7, 2-99
TTC_ConnectTimeout, 2-111
TTC_FailoverPortRange, 2-112
TTC_Keepalive_Probes, 2-124
TTC_NoReconnectOnFailover, 2-113
TTC_Random_Selection, 2-114
TTC_Redirect, 2-115
TTC_Redirect_Limit, 2-116, 2-117
TTC_RollbackRequiredOnFailover, 2-117
TTC_Server, 2-118
TTC_Server_DSN, 2-121
TTC_Server_DSN2, 2-122
TTC_Server2, 2-120
TTC_TCP_Keepalive_Interval_MS, 2-123
TTC_TCP_KEEPALIVE_PROBES, 2-124
TTC_TCP_Keepalive_Time_MS, 2-125
TTC_Timeout, 2-111, 2-126
ttConfiguration built-in procedure (check values), 3-94
UID, 2-73
WaitForConnect, 2-74
ConnectionCharacterSet connection attribute, 2-76
ConnectionName connection attribute, 2-53
Connections connection attribute, 2-35
consistency checking, ttCheck utility, 5-39
correlation name, optimizer, 3-168
CreateEpochAtCommit connection attribute, 2-54
creation, database
  AutoCreate connection attribute, 2-27
  Overwrite connection attribute, 2-47
  PermSize connection attribute, 2-48
  Preallocate connection attribute, 2-21
  RecoveryThreads connection attribute, 2-49
TempSize connection attribute, 2-50
creation, grid
ttGridRollout utility, 4-112

d daemon
  administration, ttDaemonAdmin utility, 5-48
  log, ttDaemonLog utility, 5-50
Data Source Name connection attribute, 2-11
data store attributes, 2-2, 2-10
DatabaseCharacterSet connection attribute, 2-13
DataStore connection attribute, 2-12
DDLReplicationAction connection attribute, 2-55
DDLReplicationLevel connection attribute, 2-56
Default optimizer flag, 3-163
description connection attribute, 2-16
destroy, ttDestroy utility, 5-54
Diagnostics connection attribute, 2-58
disk space, file system, Preallocate connection attribute, 2-21
Driver connection attribute, 2-17
Durability connection attribute, 2-18
durable commits, ttDurableCommit built-in procedure, 3-112
DurableCommits connection attribute, 2-59
DynamicLoadEnable connection attribute, 2-100
DynamicLoadEnable optimizer flag, 3-162
DynamicLoadErrorMode connection attribute, 2-101
DynamicLoadErrorMode optimizer flag, 3-162

E
EpochInterval connection attribute, 2-36
error message, maximum length, 6-2
expressions, SELECT statement, maximum, 6-2

F
file system disk space, Preallocate connection attribute, 2-21
files, open, maximum, 6-3
first connection attributes, 2-9, 2-127
fixed-length column, maximum length, 6-1
force Compile optimizer flag, 3-162
force Connect connection attribute, 2-37
force DisconnectEnabled connection attribute, 2-38
fragmentation
  eliminating with ttCompact built-in procedure, 3-91

G
general connection attributes, 2-4, 2-51
GenPlan optimizer flag, 3-162
grid
  administration -- See ttGridRollout
  creation--See ttGridRollout
GridLogRecoveryThreshold connection attribute, 2-19
| H | Hash optimizer flag, 3-162 |
| - | HashGb optimizer flag, 3-162 |
|   | Host name |
|   | ttHostNameGet built-in procedure, 3-117 |
|   | ttHostNameSet built-in procedure, 3-118 |
| I | IF-THEN-ELSE command, ttIsql utility, 5-79 |
|   | IncludeInCore connection attribute, 2-60 |
|   | Index Advisor--see ttIndexAdviceXXX built-in procedures |
|   | IndexedOR optimizer flag, 3-162 |
|   | indexes |
|   | columns, maximum, 6-2 |
|   | maximum per table, 6-2 |
|   | redundant indexes check, 3-190 |
|   | isolation connection attribute, 2-61 |
| J | Join order, maximum length, 6-2 |
| L | Latch stats, 3-130 |
|   | Limits, system, 6-1 |
|   | Limits--also see Maximum |
|   | Load from Oracle |
|   | ttIsql createandloadfromoraquery command, 5-69 |
|   | ttLoadFromOracle built-in procedure, 3-132 |
|   | ttTableSchemaFromOraQueryGet built-in procedure, 3-251 |
|   | Locking, database |
|   | LockLevel connection attribute, 2-63 |
|   | LockWait connection attribute, 2-64 |
|   | ttLockLevel built-in procedure, 3-136 |
|   | ttLockWait built-in procedure, 3-138 |
|   | LockLevel connection attribute, 2-63 |
|   | LockWait connection attribute, 2-64 |
|   | Log buffer size, LogBufMB connection attribute, 2-40 |
|   | LogDaemon connection attribute, 5-50 |
|   | Directory, LogDir connection attribute, 2-19, 2-20 |
|   | File size, LogFileSize connection attribute, 2-42 |
|   | flush, LogFlushMethod connection attribute, 2-43 |
|   | Holds, ttLogHolds built-in procedure, 3-140 |
|   | parallelism, LogBufParallelism connection attribute, 2-41 |
|   | purge, LogPurge connection attribute, 2-44 |
|   | LogAutoTruncate connection attribute, 2-39 |
|   | LogBufMB connection attribute, 2-40 |
|   | LogBufParallelism connection attribute, 2-41 |
|   | LogDir connection attribute, 2-19, 2-20 |
|   | LogFileSize connection attribute, 2-42 |
|   | LogFlushMethod connection attribute, 2-43 |
|   | LogPurge connection attribute, 2-44 |
| M | MaxConnsPerServer connection attribute, 2-128 |
|   | Maximum |
|   | basic name, length, 6-2 |
|   | columns per table, 6-1 |
|   | columns, GROUP BY clause, 6-1 |
|   | columns, ORDER BY clause, 6-1 |
|   | Concurrent application connections, 6-2 |
|   | Concurrent client connections, 6-2 |
|   | Concurrent connections, 6-1 |
|   | Concurrent connections system plus applications, 6-2 |
|   | Concurrent shared memory segment client connections per TimesTen instance, 6-2 |
|   | Database name, length, 6-2 |
|   | Error message length, 6-2 |
|   | Expressions, SELECT statement, 6-2 |
|   | Files open, 6-3 |
|   | Fixed-length column, length, 6-1 |
|   | Index, number of columns, 6-2 |
|   | Indexes per table, 6-2 |
|   | Join order length, 6-2 |
|   | Nesting subqueries, depth, 6-2 |
|   | Partitions per table, 6-2 |
|   | Path name, length, database in AWT cache group, 6-2 |
|   | PL/SQL blocks, recently used, cached, 6-2 |
|   | PL/SQL statements, prepared, per connection, 6-2 |
|   | Predicate string, SYS.PLAN table, length, 6-2 |
|   | Propagator databases, 6-1 |
|   | Replicated column, length, 6-1 |
|   | Row, cumulative length of fixed-length columns, 6-1 |
|   | Shared memory segment size, client connections, 6-2 |
|   | SQL statement, length, 6-2 |
|   | Statement handles per shared memory segment client connections, 6-2 |
|   | Subscriber databases, 6-1 |
|   | Subscriber databases, active standby pair, 6-1 |
|   | Table references, SQL query, 6-2 |
|   | Varying-length column, length, 6-1 |
|   | Memory |
|   | low memory warning, ttWarnOnLowMemory built-in procedure, 3-255 |
|   | MemoryLock connection attribute, 2-45 |
|   | out-of-memory warnings, PermWarnThreshold connection attribute, 2-66 |
|   | out-of-memory warnings, TempWarnThreshold connection attribute, 2-72 |
|   | Permanent, PermSize connection attribute, 2-48 |
|   | Temporary, TempSize connection attribute, 2-50 |
|   | MemoryLock connection attribute, 2-45 |
|   | MergeJoin optimizer flag, 3-162 |
|   | Metrics--see ttStats |
|   | Migrate, ttMigrate utility, 5-97 |
minimum database size, 6-1
multibyte characters, path name, non-support, 6-3

N
name
  basic, maximum length, 6-2
  database, maximum length, 6-2
NestedLoop optimizer flag, 3-162
nesting subqueries, maximum depth, 6-2
NLS general connection attributes, 2-6, 2-75
NLS_LENGTH_SEMANTICS connection attribute, 2-77
NLS_NCHAR_CONV_EXCP connection attribute, 2-78
NLS_SORT connection attribute, 2-79
NoRemRowIdOpt optimizer flag, 3-162

O
optimizer
  correlation name, 3-168
  execution plans, generation, ttOptUseIndex built-in procedure, 3-180
ttOptSetOrder built-in procedure, 3-168
optimizer flags
  AllFlags, 3-163
  BranchAndBound, 3-161
  Default, 3-163
  DynamicLoadEnable, 3-162
  DynamicLoadErrorMode, 3-162
  FirstRow, 3-162
  ForceCompile, 3-162
  GenPlan, 3-162
  Hash, 3-162
  HashGb, 3-162
  IndexedOR, 3-162
  MergeJoin, 3-162
  NestedLoop, 3-162
  NoRemRowIdOpt, 3-162
  PassThrough, 3-163
  Range, 3-163
  Rowid, 3-163
  RowLock, 3-163, 3-164
  Scan, 3-163
  settings, ttOptGetFlag built-in procedure, 3-153
  settings, ttOptSetFlag built-in procedure, 3-161
  ShowJoinOrder, 3-163
  TblLock, 3-163, 3-164
  TmpHash, 3-163
  TmpRange, 3-163
  TmpTable, 3-163
  UseBoyerMooreStringSearch, 3-163
OptimizerHint connection attribute, 2-65
OracelNetServiceName connection attribute, 2-102
OraclePWD connection attribute, 2-103
ORDER BY clause, maximum columns, 6-1
Overwrite connection attribute, 2-47

P
parallel checkpoint read, 2-34
parallel replication
  ReplicationApplyOrdering connection attribute, 2-22
  ReplicationParallelism connection attribute, 2-24
  partitions, maximum per table, 6-2
  PassThrough connection attribute, 2-104
  PassThrough optimizer flag, 3-163
password
  OraclePWD connection attribute, 2-103
  PWD connection attribute, 2-73
  PWDCrypt connection attribute, 2-68
  ttUser utility, 5-169
passwordless SSH setup, TimesTen Scaleout, 4-61
path name
  maximum length, database in AWT cache group, 6-2
  multibyte characters, non-support, 6-3
PermSize connection attribute, 2-48
PermWarnThreshold connection attribute, 2-66
PLSCOPE_SETTINGS connection attribute, 2-88
PL/SQL blocks, recently used, maximum cached per session, 6-2
PL/SQL first connection attributes, 2-6, 2-83
PL/SQL general connection attributes, 2-6, 2-87
PL/SQL statements, prepared, maximum per connection, 6-2
PLSQL_CCFLAGS connection attribute, 2-89
PLSQL_CONN_MEM_LIMIT connection attribute, 2-90
PLSQL_MEMORY_ADDRESS connection attribute, 2-84
PLSQL_MEMORY_SIZE connection attribute, 2-86
PLSQL_OPTIMIZE_LEVEL connection attribute, 2-91
PLSQL_SESSION_CACHED_CURSORS connection attribute, 2-92
PLSQL_TIMEOUT connection attribute, 2-93
Preallocate connection attribute, 2-21
predicate string, SYS.PLAN table, maximum length, 6-2
PrivateCommands connection attribute, 2-67
propagator databases, maximum, 6-1
PWD connection attribute, 2-73
PWDCrypt connection attribute, 2-68

Q
query
  nesting subqueries, maximum depth, 6-2
  plans, ttSQLCmdQueryPlan built-in procedure, 3-238
  table references, maximum, 6-2
QueryThreshold connection attribute, 2-69

R
RACCallback connection attribute, 2-107
ram policy, ttAdmin utility, 5-9
Index-7

ram policy—see ttRamPolicyXXX built-in procedures
Range optimizer flag, 3-163
RecoveryThreads connection attribute, 2-49
replicated column, maximum length, 6-1
replication
  DDLReplicationAction connection attribute, 2-55
  DDLReplicationLevel connection attribute, 2-56
  policy, ttAdmin utility, 5-14
  ttCWAdmin utility, 5-42
  ttRepAdmin utility, 5-110
replication—also see ttRepXXX built-in procedures
ReplicationApplyOrdering connection attribute, 2-22
ReplicationParallelism connection attribute, 2-24
ReplicationTrack connection attribute, 2-70
restore, ttRestore utility, 5-124
row, maximum cumulative length of fixed-length columns, 6-1
Rowid optimizer flag, 3-163
row-level locking, optimizer flag, 3-163, 3-164
RowLock optimizer flag, 3-163, 3-164

S
Scan optimizer flag, 3-163
schema, ttSchema utility, 5-126
SELECT statement, expressions, maximum, 6-2
server connection attributes, 2-9, 2-127
ServersPerDSN connection attribute, 2-129
ServerStackSize connection attribute, 2-130
session, PL/SQL blocks, recently used, maximum cached, 6-2
shared memory segment
  concurrent client connections per TimesTen instance, maximum, 6-2
  size, maximum, client connections, 6-2
ShowJoinOrder optimizer flag, 3-163
size, database, minimum, 6-1
size, table
  ttSize built-in procedure, 3-230
  ttSize utility, 5-131
SQL command cache
query plans, ttSQLCmdQueryPlan built-in procedure, 3-238
  ttSQLCmdCacheInfo built-in procedure, 3-233
  ttSQLCmdCacheInfoGet built-in procedure, 3-236
  ttSQLCmdQueryPlan built-in procedure, 3-238
SQL query, table references, maximum, 6-2
SQL statement, maximum length, 6-2
SQLQueryTimeout connection attribute, 2-71
SSH, passwordless, setup, TimesTen Scaleout, 4-61
statement handles, maximum per shared memory segment, client connections, 6-2
statement, SQL, maximum length, 6-2
statistics
  clearing stats, ttOptClearStats built-in procedure, 3-144
  configuration, ttStatsConfig built-in procedure, 3-243, 3-248
  estimating, ttOptEstimateStats built-in procedure, 3-148
  getting, ttOptGetColStats built-in procedure, 3-151
  modifying explicitly (column), ttOptSetColIntvlStats built-in procedure, 3-156
  modifying explicitly (column), ttOptSetColStats built-in procedure, 3-159
  modifying explicitly (table), ttOptSetTblStats built-in procedure, 3-171
PL/SQL memory, ttPLSQLMemoryStats built-in procedure, 3-182
TBL_STATS and COL_STATS system tables, 3-177
  ttStats utility, 5-133
  updating explicitly, ttOptUpdateStats built-in procedure, 3-177
status
  ttDataStoreStatus built-in procedure, 3-98
  ttStatus utility, 5-161
subqueries, nesting, maximum depth, 6-2
subscriber databases
  maximum, 6-1
  maximum, active standby pair, 6-1
syslog (UNIX), 5-164
SYS.PLAN table, predicate string, maximum length, 6-2
system table ID, XLA, ttSetUserTableID built-in procedure, 3-229

T
  table
    columns, maximum, 6-1
    indexes, maximum, 6-2
    partitions, maximum, 6-2
    references, SQL query, maximum, 6-2
    size, ttSize built-in procedure, 3-230
    size, ttSize utility, 5-131
    TBL_STATS system table, 3-177
    TblLock optimizer flag, 3-163, 3-164
    TCP_Port connection attribute, 2-109
    TCP_Port2 connection attribute, 2-110
    Temporary connection attribute, 2-25
    temporary database, Temporary connection attribute, 2-25
    TempSize connection attribute, 2-50
    TempWarnThreshold connection attribute, 2-72
    threads
      RecoveryThreads connection attribute, 2-49
      timeout, 2-92
      PLSQL_TIMEOUT connection attribute, 2-93
      SQLQueryTimeout connection attribute, 2-71
      TTC_ConnectTimeout connection attribute, 2-111
      TTC_Keepalive_Probes connection attribute, 2-124
      TTC_RollbackRequiredOnFailover connection attribute, 2-117
      TTC_TCP_Keepalive_Intvl_MS connection

Index-7
attribute, 2-123
TTC_TCP_Keepalive_Time_MS connection attribute, 2-125
TTC_Timeout connection attribute, 2-111, 2-126

tGridAdmin utility operations, 4-10

TimesTen Cache database attributes, 2-7, 2-97
TimesTen Cache first connection attributes, 2-7, 2-94
TimesTen Cache general connection attributes, 2-7, 2-99
timesten.conf, 1-1
TmplHash optimizer flag, 3-163
TmplRange optimizer flag, 3-163
TmplTable optimizer flag, 3-163

tracing
enabling/disabling, ttTraceMon utility, 5-166
information display, ttTail utility, 5-165

transaction
administration, ttXactAdmin utility, 5-173
commit, DurableCommits connection attribute, 2-59
ID, ttXactIdGet built-in procedure, 3-257
log, ttXactLog utility, 5-179

transaction log API—see XLA
ttAdmin utility, 5-5
ttAdoptStores utility, 5-18
ttAgingLRUCfg built-in procedure, 3-10
ttAgingScheduleNow built-in procedure, 3-13
ttApplicationContext built-in procedure, 3-15
tBackup utility, 5-34
ttBackupStatus built-in procedure, 3-16
ttBlockInfo built-in procedure, 3-18
ttBookmark built-in procedure, 3-19
ttBulkCp utility, 5-20
TTC_ConnectTimeout, 2-111
TTC_ConnectTimeout connection attribute, 2-111
TTC_FailoverPortRange connection attribute, 2-112
TTC_Keepalive_Probes connection attribute, 2-124
TTC_Random_Selection connection attribute, 2-114
TTC_REDIRECT connection attribute, 2-115
TTC_REDIRECT_LIMIT connection attribute, 2-116, 2-117
TTC_RollbackRequiredOnFailover, 2-117
TTC_Server connection attribute, 2-118
TTC_Server_DSN connection attribute, 2-121
TTC_Server_DSN2 connection attribute, 2-122
TTC_Server2 connection attribute, 2-120
TTC_TCP_Keepalive_Intvl_MS connection attribute, 2-123
TTC_TCP_KEEPALIVE_PROBES, 2-124
TTC_TCP_Keepalive_Time_MS connection attribute, 2-125
TTC_Timeout connection attribute, 2-111, 2-126
ttCacheADC_GetStandbyStateGet built-in procedure, 3-21
ttCacheADC_GetStandbyStateSet built-in procedure, 3-23
ttCacheADC_GetTimeoutGet built-in procedure, 3-25
ttCacheADC_GetTimeoutSet built-in procedure, 3-26

TTC_AWTMonitorCfg built-in procedure, 3-28
TTC_AutoFlushAwtSet built-in procedure, 3-30
TTC_AutoRefreshIntervalStatsGet built-in procedure, 3-30
TTC_AutoRefresh built-in procedure, 3-33
TTC_AutorefreshLogDefrag built-in procedure, 3-35
TTC_AutorefreshSelectLimit built-in procedure, 3-41
TTC_AutorefreshStatsGet built-in procedure, 3-37
TTC_AutorefreshXactLimit built-in procedure, 3-43
TTC_AWTMonitorConfig built-in procedure, 3-46
TTC_AWTThresholdGet built-in procedure, 3-48
TTC_AWTThresholdSet built-in procedure, 3-49
TTC_Backup built-in procedure, 3-51
TTC_BackupStatus built-in procedure, 3-54
TTC_DbCgStatus built-in procedure, 3-58
TTC_PolicyGet built-in procedure, 3-61
TTC_PolicySet built-in procedure, 3-63
TTC_PropagateFlagSet built-in procedure, 3-65
TTC_SqlGet built-in procedure, 3-67
TTC_Start built-in procedure, 3-69
TTC_Stop built-in procedure, 3-71
TTC_UidGet built-in procedure, 3-73
TTC_UidPwdSet built-in procedure, 3-74
ttCapture utility, 5-37
ttCheck utility, 5-39
TTC_Ckp built-in procedure, 3-76
TTC_CkpBlocking built-in procedure, 3-78
TTC_CkpConfig built-in procedure, 3-80
TTC_CkpHistory built-in procedure, 3-83
TTC_CommitBufferStats built-in procedure, 3-88
TTC_CommitBufferStatsReset built-in procedure, 3-90
TTC_Compact built-in procedure, 3-91
TTC_ComputeTabSizes built-in procedure, 3-92
TTC_Configuration built-in procedure, 3-94
TTC_Context built-in procedure, 3-97
TTC_CWAdmin utility, 5-42
TTC_DaemonAdmin utility, 5-48
TTC_DaemonLog utility, 5-50
TTC_DataStoreStatus built-in procedure, 3-98
TTC_DBcompactConfig built-in procedure, 3-100
TTC_DBConfig built-in procedure, 3-103
TTC_DBWriteConcurrencyModeGet built-in procedure, 3-106
TTC_DBWriteConcurrencyModeSet built-in procedure, 3-108
TTC_Destroy utility, 5-54
TTC_DistributeProgress built-in procedure, 3-110
TTC_DurableCommit built-in procedure, 3-112
TTC_EpochCreate built-in procedure, 3-113
TTC_EpochSessionGet built-in procedure, 3-114
TTC_GridAdmin utility, 4-3

connectable operations, 4-17
data space group operations, 4-22
database backup and restore operations, 4-12
database definition operations, 4-24
database import and export operations, 4-71
database operations, 4-30
grid operations, 4-55
ttXlaSubscribe built-in procedure, 3-262
XLA

UID connection attribute, 2-73
UseBoyerMooreStringSearch optimizer flag, 3-163
user table ID, XLA, ttSetUserTableID built-in procedure, 3-229
utilities
  access control, 4-1, 5-1
  authentication, 2-73, 4-1, 5-1
  privileges, 4-1, 5-1
  TimesTen Scaleout, 4-2, 5-2
  TimesTen Scaleout or TimesTen Classic, 4-2, 5-2
  ttAdmin, 5-5
  ttAdoptStores, 5-18
  ttBackup, 5-34
  ttBulkCp, 5-20
  ttCapture, 5-37
  ttCheck, 5-39
  ttCWAdmin, 5-42
  ttDaemonAdmin, 5-48
  ttDaemonLog, 5-50
  ttDestroy, 5-54
  ttGridAdmin, 4-3
  ttGridRollout, 4-112
  ttInstallationCheck, 5-56
  ttInstallDSN, 5-57
  ttInstanceCreate, 5-59
  ttInstanceDestroy, 5-61
  ttInstanceModify, 5-62
  ttIsql, 5-64
  ttMigrate, 5-97
  ttRepAdmin, 5-110
  ttRestore, 5-124
  ttSchema, 5-126
  ttSize, 5-131
  ttStats, 5-133
  ttStatus, 5-161
  ttSysLogCheck, 5-164
  ttTail, 5-165
  ttTraceMon, 5-166
  ttUser, 5-169
  ttVersion, 5-170
  ttXactAdmin, 5-173
  ttXactLog, 5-179

V

varying-length column, maximum length, 6-1
version, TimesTen
  ttVersion built-in procedure, 3-253
  ttVersion utility, 5-170

W

WaitForConnect connection attribute, 2-74
WHENEVER SQLERROR command, ttIsql utility, 5-80