Oracle® TimesTen In-Memory Database

Database Reference

Release 26.1
F54460-01
March 2026

Oracle TimesTen In-Memory Database Database Reference, Release 26.1

F54460-01

Copyright © 1996, 2026, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

1 TimesTen Instance Configuration File

2 Connection Attributes

List of Connection Attributes 1
General Use Connection Attributes 8
Data Store Attributes 8
Data Source Name 8
DatabaseCharacterSet 9
DataStore 12
DbEncrypted 12
DbKeyWallet 13
Description 14
Driver 14
LogDir 15
Preallocate 15
ReplicationApplyOrdering 16
ReplicationParallelism 17
First Connection Attributes 18
AutoCreate 18
BackupFailThreshold 19
CkptFrequency 20
CkptLogVolume 20
CkptRate 21
CkptReadThreads 22
Connections 23
DefragThreshold 24
DefragSpeed 25
DefragCycle 25
EnableParallelConnect 26
EnableParallelRedo 27
ForceConnect 27
ForceDisconnectEnabled 28
LogAutoTruncate 29

Copyright © 1996, 2026, Oracle and/or its affiliates.

LogBufMB 29
LogBufParallelism 30
LogFileSize 31
LogFlushMethod 32
LogPurge 32
MemoryLock 33
Overwrite 34
PermSize 35
ShmPagesNoTouch 36
RecoveryThreads 37
RedoParallelism 38
RedoSegmentSize 38
TempSize 39
General Connection Attributes 39
CommitBufferSizeMax 40
ConnectionName 40
DDLReplicationAction 41
DDLReplicationLevel 42
Diagnostics 44
DurableCommits 44
IncludeInCore 45
Isolation 46
LockLevel 47
LockWait 48
OptimizerHint 49
PermWarnThreshold 50
PrivateCommands 50
PWDCrypt 51
PwdWallet 52
QueryThreshold 53
ReplicationTrack 53
SQLQueryTimeout 54
SQLQueryTimeoutMSec 55
TempWarnThreshold 56
TransactionTimeout 56
UID and PWD 57
WaitForConnect 58
Globalization Connection Attributes 59
NLS General Connection Attributes 59
ConnectionCharacterSet 59
NLS_LENGTH_SEMANTICS 60
NLS_NCHAR_CONV_EXCP 61

Copyright © 1996, 2026, Oracle and/or its affiliates.

NLS_SORT 61
PL/SQL Connection Attributes 65
PL/SQL First Connection Attributes 65
PLSQL_MEMORY_ADDRESS 65
PLSQL_MEMORY_SIZE 66
PLSQL_OPEN_CURSORS 67
PL/SQL General Connection Attributes 68
PLSQL_CCFLAGS 68
PLSQL_CONN_MEM_LIMIT 69
PLSQL_OPTIMIZE_LEVEL 70
PLSQL_SESSION_CACHED_CURSORS 70
PLSQL_TIMEOUT 71
Cache Connection Attributes 72
Cache First Connection Attributes 72
CacheAdminWallet 72
CacheAWTMethod 73
Cache Database Attributes 74
CacheAWTParallelism 74
UseCacheConnPool 75
Cache General Connection Attributes 76
DynamicLoadEnable 76
DynamicLoadErrorMode 77
OracleNetServiceName 78
OraclePWD 78
PassThrough 79
RACCallback 81
StandbyNetServiceName 82
Client and Server Connection Attributes 82
Client General Connection Attributes 82
TTC_REDIRECT 83
CipherSuites 84
ChildServer 85
Encryption 86
SSLClientAuthentication 86
TCP_Port 87
TCP_Port2, TCP_PortN 88
TTC_ConnectTimeout 89
TTC_FailoverPortRange 89
TTC_NetMsgMaxBytes 90
TTC_NetMsgMaxRows 91
TTC_NoReconnectOnFailover 92
TTC_Random_Selection 92

Copyright © 1996, 2026, Oracle and/or its affiliates.

TTC_Server or TTC_Server1 93
TTC_Server2, TTC_ServerN 94
TTC_Server_DSN 95
TTC_Server_DSN2, TTC_Server_DSNn 96
TTC_SqlTimeoutMS 96
TTC_TCP_KEEPALIVE_INTVL_MS 97
TTC_TCP_KEEPALIVE_PROBES 97
TTC_TCP_KEEPALIVE_TIME_MS 98
TTC_Timeout 98
Wallet 99
Server First Connection Attributes 100
CipherSuites 100
Encryption 101
SSLClientAuthentication 102
MaxConnsPerServer 103
ServersPerDSN 104
ServerStackSize 105
SSLRenegotiationPeriod 105
SSLRenegotiationSize 106
Wallet 107

3 Built-In Procedures

List of Built-In Procedures 2
ttAgingLRUConfig 7
ttAgingTableLRUConfig 9
ttAgingScheduleNow 11
ttApplicationContext 13
ttBackupStatus 14
ttBlockInfo 16
ttBookmark 16
ttCacheADGStandbyStateGet 17
ttCacheADGStandbyStateSet 18
ttCacheADGStandbyTimeoutGet 20
ttCacheADGStandbyTimeoutSet 21
ttCacheAllowFlushAwtSet 22
ttCacheAutorefIntervalStatsGet 23
ttCacheAutorefresh 25
ttCacheAutorefreshLogDefrag 26
ttCacheAutorefreshStatsGet 27
ttCacheAutorefreshSelectLimit 31
ttCacheAutorefreshXactLimit 33

Copyright © 1996, 2026, Oracle and/or its affiliates.

ttCacheAWTMonitorConfig 35
ttCacheAWTThresholdGet 37
ttCacheAWTThresholdSet 38
ttCacheCheck 39
ttCacheConfig 41
ttCacheConnPoolApply 46
ttCacheConnPoolGet 47
ttCacheConnPoolSet 49
ttCacheDbCgStatus 50
ttCacheDDLTrackingConfig 52
ttCacheLoadSCNGet 53
ttCacheLogStats 54
ttCachePolicyGet 55
ttCachePolicySet 56
ttCachePropagateFlagSet 57
ttCacheSqlGet 59
ttCacheStart 61
ttCacheStop 62
ttCacheUidGet 63
ttCacheUidPwdSet 63
ttCkpt 65
ttCkptBlocking 66
ttCkptConfig 67
ttCkptConfigGet 69
ttCkptHistory 70
ttCommitBufferStats 74
ttCommitBufferStatsReset 76
ttCompact 76
ttComputeTabSizes 77
ttConfiguration 78
ttContext 80
ttDataStoreStatus 80
ttDBCompactConfig 82
ttDBConfig 84
ttDBWriteConcurrencyModeGet 88
ttDBWriteConcurrencyModeSet 89
ttDurableCommit 91
ttHeapInfo 91
ttHostNameGet 92
ttHostNameSet 93
ttIndexAdviceCaptureDrop 94
ttIndexAdviceCaptureEnd 95

Copyright © 1996, 2026, Oracle and/or its affiliates.

ttIndexAdviceCaptureInfoGet 96
ttIndexAdviceCaptureOutput 98
ttIndexAdviceCaptureStart 99
ttLatchStatsGet 100
ttLoadFromOracle 102
ttLockLevel 105
ttLockWait 106
ttLogHolds 107
ttMonitorHighWaterReset 110
ttOptClearStats 110
ttOptCmdCacheInvalidate 111
ttOptEstimateStats 113
ttOptGetColStats 115
ttOptGetFlag 116
ttOptGetMaxCmdFreeListCnt 117
ttOptGetOrder 117
ttOptSetColIntvlStats 118
ttOptSetColStats 120
ttOptSetFlag 121
ttOptSetMaxCmdFreeListCnt 126
ttOptSetMaxPriCmdFreeListCnt 126
ttOptSetOrder 127
ttOptSetTblStats 129
ttOptShowJoinOrder 130
ttOptStatsExport 131
ttOptUpdateStats 132
ttOptUseIndex 135
ttPageLevelTableInfo 136
ttPLSQLMemoryStats 138
ttRamPolicyAutoReloadGet 140
ttRamPolicyAutoReloadSet 141
ttRamPolicyGet 141
ttRamPolicySet 143
ttRedundantIndexCheck 144
ttRekeyStatus 145
ttRepDeactivate 146
ttReplicationStatus 146
ttRepPolicyGet 148
ttRepPolicySet 149
ttRepQueryThresholdGet 150
ttRepQueryThresholdSet 151
ttRepStart 153

Copyright © 1996, 2026, Oracle and/or its affiliates.

ttRepStateGet 154
ttRepStateSave 155
ttRepStateSet 156
ttRepStop 157
ttRepSubscriberStateSet 158
ttRepSubscriberWait 159
ttRepSyncGet 161
ttRepSyncSet 163
ttRepSyncSubscriberStatus 165
ttRepTransmitGet 166
ttRepTransmitSet 167
ttRepXactStatus 168
ttRepXactTokenGet 169
ttSetUserColumnID 170
ttSetUserTableID 171
ttSize 172
ttSQLCmdCacheInfo 174
ttSQLCmdCacheInfoGet 177
ttSQLCmdQueryPlan 178
ttSQLExecutionTimeHistogram 180
ttStatsConfig 182
ttStatsConfigGet 185
ttTableSchemaFromOraQueryGet 187
ttVersion 188
ttWarnOnLowMemory 189
ttXactIdGet 190
ttXlaBookmarkCreate 191
ttXlaBookmarkDelete 193
ttXlaSubscribe 193
ttXlaUnsubscribe 194

4 Utilities

Overview 1
Utilities List 1
ttAdmin 3
- Help, Version, and Query Options 4
- Perform RAM Operations 5
- Open or Close a Database 9
- Transport Layer Security 9
- Encryption at Rest 10
- Force Disconnect 11

Copyright © 1996, 2026, Oracle and/or its affiliates.

Set Cache Policies 13
Set Replication Policies 15
ttAdoptStores 17
ttBackup 19
ttBulkCp 21
ttCapture 34
ttCheck 36
ttClientImport 38
ttCreateCerts 39
ttCacheInfo 45
ttCWAdmin 46
ttDaemonAdmin 52
ttDaemonLog 53
ttDestroy 57
ttExporter 58
ttInstallationCheck 61
ttInstallDSN 62
ttInstanceCreate 63
ttInstanceDestroy 68
ttInstanceModify 69
ttIsql 72
ttMigrate 74
ttRepAdmin 86
Help and Version Information 87
Database Information 87
Subscriber Database Operations 87
Duplicate a Database 89
Wait for Updates to Complete 93
Replication Status 94
ttRestore 98
ttSchema 100
ttShmSize 106
ttSize 107
ttStats 109
ttStatus 129
ttTail 131
ttTraceMon 132
ttUser 134
ttVersion 137
ttXactAdmin 139
ttXactLog 144

Copyright © 1996, 2026, Oracle and/or its affiliates.

5 System Limits

System Limits and Defaults 1
Limits on Number of Open Files 2
Path Names 3

Copyright © 1996, 2026, Oracle and/or its affiliates.

About This Content

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

Audience

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

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 (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:

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

In addition, TimesTen documentation uses the following special conventions.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Convention	Meaning
installation_dir	The path that represents the directory where TimesTen is installed.
timesten_home	The path that represents the home directory of a TimesTen instance.
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, 221 or 22.1 represents TimesTen Release 22.1.
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 details 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. The documentation may not be up to date. You can confirm the current release number by looking at the Release Notes or running the `ttVersion` utility.

Copyright © 1996, 2026, Oracle and/or its affiliates.

1

TimesTen Instance Configuration File

The TimesTen instance configuration file contains TimesTen instance configuration attributes 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.
The following table contains all the TimesTen configuration attributes, descriptions, default values, and data types.

Note that some configuration attributes have a corresponding connection attribute. In these cases, you can override the setting of the configuration attribute in the server `timesten.conf` file by setting the connection attribute in the database DSN definition. Similarly, you can override the setting of the configuration attribute in the client `timesten.conf` file by setting the connection attribute in either the client DSN definition or the connection string.

Name	Description	Value or Data Type
`admin_uid` *Required*	The OS `uid` number of the instance administrator. This entry is added by the ttInstanceCreate utility and must not be changed. *Default:* None	Integer
`admin_user` *Required*	The OS user name of the instance administrator that matches the OS owner of the instance home directory. This entry is added by the ttInstanceCreate utility and must not be changed. *Default:* None	String
`allow_network_files` *Optional*	Indicates if data access on NFS-mounted systems is allowed. On Linux x86 and Solaris, you can access checkpoint and transaction log files on NFS-mounted systems. TimesTen ignores this attribute on platforms other than Linux x86 and Solaris. • `1` indicates that TimesTen systems can access data across NFS-mounted systems. • `0` indicates that TimesTen systems cannot access data across NFS-mounted systems. *Default:* `0` For more information, see Using NFS-Mounted Systems for Checkpoint and Transaction Log Files in the *Oracle TimesTen In-Memory Database Operations Guide*.	`1` or `0`

Copyright © 1996, 2026, Oracle and/or its affiliates.

Name	Description	Value or Data Type
`client_cipher_suites` and `server_cipher_suites` *Optional*	Lists the cipher suite or suites that can be used for Transport Layer Security. To use TLS for client/server connections, set `client_cipher_suites` in the client `timesten.conf` file and `server_cipher_suites` in the server `timesten.conf` file. Specify one or more TLS version 1.3 or 1.2 cipher suites in a comma-separated list from strongest to weakest. For TLS to be used, the server and client settings must include at least one common cipher suite. See Cipher Suites in TimesTen in *Oracle TimesTen In-Memory Database Security Guide* to learn more about the supported cipher suites and how TimesTen negotiates cipher suite selection. *Default:* None `server_cipher_suites` can also be set through: • The `-serverCipherSuites` option of the `ttInstanceCreate` command for TimesTen. **Note** You can also use the `CipherSuites` connection attribute for TimesTen to set `client_cipher_suites` and `server_cipher_suites`.	String
`client_only` *Optional*	Indicates if the instance is client-only or not. This entry is added by the ttInstanceCreate utility and must not be changed. *Default:* `No`	`Yes` or `No`

Copyright © 1996, 2026, Oracle and/or its affiliates.

Name	Description	Value or Data Type
`client_wallet` and `server_wallet` *Optional*	Specifies the fully qualified path to the wallet directory—the directory where generated certificates are located. *Default:* None For more information, see Transport Layer Security for TimesTen Client/Server in the *Oracle TimesTen In-Memory Database Security Guide*.	Pathname
`daemon_log_snippet_interval_in_mins` *Optional*	The duration of the daemon log collection for a critical event. The granularity of the duration is in minutes. The minimum value is 1 minute. *Default:* 10	Integer
`daemon_port` *Required*	The valid TCP port number on which the TimesTen daemon for this instance listens. TimesTen ignores this attribute for client-only instances. *Default:* 6624	Number
`db_fips_140` *Optional*	Specifies whether FIPS 140-2 mode for symmetric encryption of TimesTen network communications is enabled. If specified, it uses a FIPS-validated version of the encryption libraries available to provide the encryption. • 1 (FIPS) indicates that FIPS 140-2 mode for symmetric encryption is enabled. • `0` (non-FIPS) indicates that FIPS 140-2 mode for symmetric encryption is disabled. **Notes** • If you specify FIPS mode and the libraries do not exist, the daemon does not start. • The `db_fips_140` value is read when each process starts. • If you change the value while a daemon is running, the change does not take effect on the daemon until it is restarted. • If you change the value while a daemon is running, any new process started after this change is affected by it (for instance, running a `ttIsql` operation) *Default:* `0`	1 or `0`

Copyright © 1996, 2026, Oracle and/or its affiliates.

Name	Description	Value or Data Type
`enable_policy_inactive` *Optional*	`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`. *Default:* 1 For more information, see Specifying a RAM Policy in the *Oracle TimesTen In-Memory Database Operations Guide*.	`1` or `0`
`enableipv6` *Optional*	Indicates whether to support IPv6 or not. 1 means supported, and `0` means not supported. *Default:* 1	`1` or `0`
`facility` *Optional*	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`. *Default:* None	String
`group_name` *Required*	The primary group name of the instance administrator. This entry is added by the ttInstanceCreate utility, and must not be changed. *Default:* None	String
`hostname` *Required*	The OS hostname of the instance. This entry is added by the ttInstanceCreate utility, and must not be changed. *Default:* None	String
`instance_guid` *Required*	A globally unique ID, guaranteed to be different than the ID of any other instance. TimesTen adds this ID during instance creation. *Default:* None	String
`instance_name` *Required*	The name of the TimesTen instance. An ASCII name from 1 to 255 characters long. This attribute is optional for client-only instances. *Default:* `instance1`	String

Copyright © 1996, 2026, Oracle and/or its affiliates.

Name	Description	Value or Data Type
`listen_addr` and `listen_6_addr` *Optional*	`listen_addr` indicates the IPv4 address where the TimesTen daemon and servers listen. `listen_6_addr` indicates the IPv6 address where the TimesTen daemon and servers listen. By default, TimesTen supports IPv6. *Default:* None	String
`max_conns_per_server` *Optional*	The maximum number of client/server connections for the TimesTen server. You can also set this with the MaxConnsPerServer connection attribute. *Default:* 1	Integer
`max_subs` *Optional*	The maximum number of subdaemons TimesTen spawns. The main TimesTen daemon spawns subdaemons dynamically as they are needed. *Default:* 50	Integer
`max_support_log_files` *Optional*	The TimesTen main daemon automatically rotates the files once they get to a specific size. This attribute specifies the maximum number of daemon log files to keep. *Default:* 10	Integer
`max_support_log_size_mb` *Optional*	The maximum size of a TimesTen daemon log file in megabytes. *Default:* 100 MB	Integer
`max_user_log_files` *Optional*	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. *Default:* 10	Integer
`max_user_log_size_mb` *Optional*	The maximum size of a TimesTen daemon user log file in megabytes. *Default:* 10 MB	Integer
`min_subs` *Optional*	The minimum number of subdaemons TimesTen spawns. The main TimesTen daemon spawns subdaemons dynamically as they are needed. *Default:* 4	Integer
`noserverlog` *Optional*	Turns logging on or off for connects to and disconnects from client applications. `1` indicates logging is on. `0` indicates that logging is off. *Default:* 1	`1` or `0`

Copyright © 1996, 2026, Oracle and/or its affiliates.

Name	Description	Value or Data Type
`openssl_path` *Optional*	Specifies the fully qualified directory path where the OpenSSL library resides. If you do not provide a value for this attribute, TimesTen uses the OpenSSL library, if any, that the system uses by default. *Default:* None	String
`replication_cipher_suite` *Optional*	The cipher suite to be used in encrypting communications to and from the replication agents. This setting is required if you are using TLS for replication. See see Cipher Suites in TimesTen and Transport Layer Security for TimesTen Replication in *Oracle TimesTen In-Memory Database Security Guide* for more information about supported cipher suites and replication TLS attributes. *Default:* None	String
`replication_ssl_mandatory` *Optional*	Specifies whether it is mandatory to have consistent TLS configuration between TimesTen instances—specifically, whether TLS is configured through `replication_cipher_suite` and `replication_wallet` settings, and what cipher suite is specified. If there is a mismatch between the current instance and the replication peer, then TimesTen behavior is determined as follows: • `1` indicates that replication cannot proceed unless settings are the same on all instances. • `0` indicates that TLS is not used for communication between replication agents. *Default:* 1	`1` or `0`
`replication_wallet` *Optional*	Specifies the path to the wallet directory—the directory where you placed the certificates that you generated. This setting is required if you are using TLS for replication. Oracle recommends using the same location and directory name on each TimesTen instance. *Default:* None	Pathname

Copyright © 1996, 2026, Oracle and/or its affiliates.

Name	Description	Value or Data Type
`server_encryption` *Optional*	This is the encryption flag. Possible values are: • `accepted`: Enable an encrypted session if required or requested by the client; use an unencrypted session otherwise (default). • `rejected`: Demand an unencrypted session. (If the server does not support encryption, TimesTen behaves as if this is the setting on the server.) The connection is rejected if the client requires encryption. • `requested`: Request an encrypted session if the client allows it (if the client has any setting other than `rejected`); use an unencrypted session otherwise. • `required`: Demand an encrypted session. Reject the connection if the client rejects encryption. *Default:* Accepted `server_encryption` can also be set through: • The `-serverEncryption` option of the `ttInstanceCreate` command for TimesTen. • The `Encryption` connection attribute for TimesTen. **Note** To use Transport Layer Security, settings from the server and the client must be compatible. For more information about settings compatibility, see Transport Layer Security for TimesTen Client/ Server in the *Oracle*	String

Copyright © 1996, 2026, Oracle and/or its affiliates.

Name	Description	Value or Data Type
	TimesTen In-Memory Database Security Guide.	
`server_pool` *Optional*	The number of processes that the TimesTen server should prespawn and keep in a reserve pool. If not specified, no processes are prespawned. *Default:* None	Integer
`server_port` *Optional*	The valid TCP port number on which the TimesTen server for this instance listens if you want to run a server in the instance. *Default:* 6625	Number
`servers_per_dsn` *Optional*	The number of servers per DSN for client /server implementations. The value is set by and can be changed by the ServersPerDSN attribute in the server DSN. *Default:* 1	Integer
`server_stack_size` *Optional*	The client/server stack size in kilobytes. You can also set this with the ServerStackSize connection attribute. *Default:* 768 KB	Integer
`show_date` *Optional*	Indicates whether the date is prepended on all daemon and user log entries. 1 indicates that the date is prepended to every record in the user and daemon log files. 0 indicates that the date is not prepended to every record in the user and daemon log files. *Default:* 1	1 or 0
`snmp_conf_path` *Optional*	Lists the fully qualified directory paths, separated by a colon, where the `snmp.conf` and `snmp.local.conf` files may reside. *Default:* None	Pathname
`snmp_trap` *Optional*	Specifies the fully qualified directory path where the `snmptrap` utility resides. If you do not provide a value for this attribute, TimesTen does not attempt to generate SNMP traps. *Default:* None	Pathname

Copyright © 1996, 2026, Oracle and/or its affiliates.

Name	Description	Value or Data Type
`snmp_trap_dests` *Optional*	Lists the destinations for the SNMP traps, separated by a space. Each destination must include the transport specifier and transport address. If the transport specifier or port number are not provided, `snmptrap` uses the defaults: `udp` and `161`, respectively. A destination can be specified as follows: • `[udp:]hostname[:port]` or `[udp:]IPv4-address[:port]` • `udp6:hostname[:port]`, `udp6:IPv6-address:port`, or `upd6:'['IPv6-address']'[:port]` • `dtlsudp:hostname[:port]` or `dtlsudp:IPv4-address[:port]` • `tcp:hostname[:port]` or `tcp:IPv4-address[:port]` • `tcp6:hostname[:port]`, `tcp6:IPv6-address:port`, or `tcp6:'['IPv6-address']'[:port]` • `tlstcp:hostname[:port]` or `tlstcp:IPv4-address[:port]` • `unix:pathname` *Default:* None	String
`snmp_trap_opts` *Optional*	Lists the options to be passed to the `snmptrap` utility. For information on the options supported by the `snmptrap` utility, see the man pages for `snmpcmd` and `snmptrap`. *Default:* None	String
`snmp_version` *Optional*	Specifies the SNMP version to be used. *Default:* 3	1, 2, or 3

Copyright © 1996, 2026, Oracle and/or its affiliates.

Name	Description	Value or Data Type
`ssl_client_authentication` *Optional*	Specifies whether TLS client authentication is required (setting of 1) or not (setting of 0). *Default:* `0` With client authentication, the server validates an identity presented by the client, and requires an identity (public/private key) in the client wallet. Regardless of the client authentication setting, server authentication is performed, where the client validates the server. For more information, see Transport Layer Security for TimesTen Client/Server in the *Oracle TimesTen In-Memory Database Security Guide*.	`1` or `0`
`ssl_fips_140` *Optional*	Specifies whether FIPS 140-2 mode encryption for both client/server and replication connections are enabled. If specified, it uses a FIPS-validated version of the encryption libraries available to provide the encryption. • `1` indicates that FIPS 140-2 mode encryption is enabled for client/server and replication connections. • `0` indicates that FIPS 140-2 mode encryption is disabled for client/server and replication connections. **Notes** • To use the FIPS 140-2 mode encryption, you must enable encryption on both the client and server side of the connection. • Both ends of a connection need to be either encrypted or unencrypted. However, you can mix encrypted and unencrypted client/server connections to a single database. *Default:* `0`	`1` or `0`
`supportlog` *Optional*	The location of the TimesTen daemon log file. *Default:* *`timesten_home`*`/diag/ttmesg.log`	String

Copyright © 1996, 2026, Oracle and/or its affiliates.

Name	Description	Value or Data Type
`supportlog_readable` *Optional*	Specifies the file permissions of the TimesTen daemon log file, `ttmesg.log`. Possible values are: • `owner`: Sets read and write permissions for the owner and no permissions for the group and others (`660`). • `admin`: Same as `owner`. • `group`: Sets read and write permissions for the owner, read-only permission for the group, and no permissions for others (`640`). • `all`: Sets read and write permissions for the owner and read-only permissions for the group and others (`644`). • `world`: Same as `all`. TimesTen ignores this attribute for existing daemon log files; only new files are affected. *Default:* `owner`	String
`timesten_release` *Optional*	The number of the TimesTen release that created the instance home and that can use the instance home. *Default:* The installed TimesTen release number	Number
`tns_admin` *Optional*	In a TimesTen cache environment, the directory that contains the `TNS_ADMIN` settings to allow TimesTen to communicate with the Oracle database. This attribute is required in a TimesTen cache environment only. This entry is added by the ttInstanceCreate utility and can be changed using the ttInstanceModify utility. *Default:* None	Pathname
`userlog` *Optional*	The location of the TimesTen daemon user log file. *Default:* *`timesten_home`*`/diag/tterrors.log`	String

Copyright © 1996, 2026, Oracle and/or its affiliates.

Name	Description	Value or Data Type
`userlog_readable` *Optional*	Specifies the file permissions of the TimesTen daemon user log file, `tterrors.log`. Possible values are: • `owner`: Sets read and write permissions for the owner and no permissions for the group and others (`660`). • `admin`: Same as `owner`. • `group`: Sets read and write permissions for the owner, read-only permission for the group, and no permissions for others (`640`). • `all`: Sets read and write permissions for the owner and read-only permissions for the group and others (`644`). • `world`: Same as `all`. TimesTen ignores this attribute for existing daemon user log files; only new files are affected. *Default:* `owner`	String

In TimesTen, 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 Daemon in the *Oracle TimesTen In-Memory Database Operations Guide*.

Copyright © 1996, 2026, Oracle and/or its affiliates.

2

Connection Attributes

You specify the connection attributes defined for TimesTen in the connection string.

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*

To view the names and values of most attributes specified in the connection string, use the ttConfiguration built-in procedure with your application.

> **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, `False` means the attribute value is set to `0` and `True` means the attribute value is set to `1`.

List of Connection Attributes

These are the connection attributes defined for TimesTen, with their types, and descriptions.

Name and Type	Description
AutoCreate First connection attribute	Specifies that the first connection creates the database if it does not exist.
BackupFailThreshold First connection attribute	Controls the number of log files that can accumulate before a backup fails.
CacheAdminWallet Cache first connection attribute	Specifies that credentials for the Oracle cache administration user that are specified with the `ttCacheUidPwdSet` built-in procedure are stored in a system managed Oracle Wallet.
CacheAWTMethod Cache first connection attribute	Enables the AWT propagation method to be used on Oracle database tables.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Name and Type	Description
CacheAWTParallelism Cache database attribute	Indicates the number of threads necessary to apply changes to the Oracle database.
ChildServer Client general connection attribute	Identifies the client server process.
CipherSuites Client general connection attribute Server first connection attribute	Lists the cipher suite or suites that can be used by TLS for client/server connections, depending also on the client setting.
CkptFrequency First connection attribute	Controls the frequency in seconds that TimesTen performs a background checkpoint.
CkptLogVolume First connection attribute	Controls the amount of data in megabytes (MB) that collects in the log between background checkpoints.
CkptRate First connection attribute	Controls the maximum rate at which data should be written to disk during a checkpoint operation.
CkptReadThreads First connection attribute	Controls the number of threads used to read a checkpoint file when loading the database into memory.
CommitBufferSizeMax General connection attribute	Specifies the maximum size of the commit buffer in the transaction control block.
ConnectionCharacterSet NLS general connection attribute	The character encoding for the connection, which can be different from the database character set.
ConnectionName General connection attribute	Specifies whether there is a symbolic name for the data source.
Connections First connection attribute	Indicates the upper bound on the number of user-specified concurrent connections to the database.
Data Source Name Data store attribute	A name that identifies the specific attributes of a connection to the database.
DatabaseCharacterSet NLS general connection attribute	Identifies the character set used by the database. This attribute is required at database creation time.
DataStore Data store attribute	Identifies the physical database.
DbEncrypted Data store attribute	Determines whether encryption at rest is enabled and specifies the preferred AES encryption algorithm.
DbKeyWallet Data store attribute	The directory where the wallet for encryption at rest is stored.
DDLReplicationAction General connection attribute	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.
DDLReplicationLevel General connection attribute	Enables replication of data definition language (DDL) statements in an active standby replication scheme.
DefragThreshold First connection attribute	Specifies the level of online defragmentation. A higher value leads to more space being freed.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Name and Type	Description
DefragSpeed First connection attribute	Specifies the speed at which the online defragmentation should occur. A higher value uses more CPU resources. The default value indicates that the online defragmentation is disabled.
DefragCycle First connection attribute	Specifies the number of seconds to wait until the next cycle of the online defragmentation.
Description Data store attribute	A statement that identifies the use of the data source name.
Diagnostics General connection attribute	Specifies whether diagnostic messages are generated.
Driver Data store attribute	Specifies the TimesTen ODBC Driver Manager.
DurableCommits General connection attribute	Specifies that commit operations should write log records to disk.
DynamicLoadEnable Cache general connection attribute	Enables or disables transparent load of data from an Oracle database to dynamic cache groups.
DynamicLoadErrorMode Cache general connection attribute	Determines if an error message is returned upon a transparent load failure.
EnableParallelConnect First connection attribute	Specifies whether TimesTen enables parallel database connections at the process level, allowing multiple processes to connect simultaneously.
EnableParallelRedo First connection attribute	Specifies whether TimesTen enables parallel redo for faster database recovery.
Encryption Client general connection attribute Server first connection attribute	Specifies whether encryption is required or not for a client/server connection for the client and whether encryption is accepted, rejected, requested, or required for the server.
ForceConnect First connection attribute	Specifies whether a connection is allowed to a failed database if it is not properly restored from the corresponding subscriber database.
ForceDisconnectEnabled First connection attribute	Enables forced disconnections through `ttAdmin -disconnect`.
IncludeInCore General connection attribute	Specifies which parts of shared memory are included in a core dump.
Isolation General connection attribute	Specifies whether the isolation level is read committed or serializable.
LockLevel General connection attribute	Specifies whether the connection should use row-level locking (`0`) or database-level locking (1).
LockWait General connection attribute	Enables an application to configure the lock wait interval for the connection.
LogAutoTruncate First connection attribute	Determines whether the first connection to a database should proceed if TimesTen recovery encounters a defective log record.
LogBufMB First connection attribute	The size of the internal log buffer in MB.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Name and Type	Description
LogBufParallelism First connection attribute	The number of log buffer strands.
LogDir Data store attribute	The directory where transaction log files are stored.
LogFileSize First connection attribute	The transaction log file size in MB.
LogFlushMethod First connection attribute	Controls the method used by TimesTen to write and sync log data to transaction log files.
LogPurge First connection attribute	Specifies that unneeded transaction log files are deleted during a checkpoint operation.
MaxConnsPerServer Server first connection attribute	The maximum number of concurrent connections a child TimesTen server process can handle.
MemoryLock First connection attribute	Enables applications that connect to a shared database to specify whether the real memory should be locked during database loading.
NLS_LENGTH_SEMANTICS NLS general connection attribute	The default length semantics configuration.
NLS_NCHAR_CONV_EXCP NLS general connection attribute	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.
NLS_SORT NLS general connection attribute	The collating sequence to use for linguistic comparisons.
OptimizerHint General connection attribute	Sets optimizer hints at the connection level.
OracleNetServiceName Cache general connection attribute	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.
OraclePWD Cache general connection attribute	Identifies the password for the Oracle database that is being cached in TimesTen.
Overwrite First connection attribute	Specifies that the existing database should be overwritten with a new one when a connection is attempted.
PassThrough Cache general connection attribute	Specifies which SQL statements are executed locally in TimesTen and which SQL statements are passed through to the Oracle database for execution.
PermSize First connection attribute	The size in MB for the permanent partition of the database.
PermWarnThreshold General connection attribute	The threshold at which TimesTen returns a warning when the permanent partition of the database is low in memory.
PLSQL_CCFLAGS PL/SQL general connection attribute	Controls conditional compilation of PL/SQL units.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Name and Type	Description
PLSQL_CONN_MEM_LIMIT PL/SQL general connection attribute	Specifies the maximum amount of process heap memory in MB that PL/SQL can use for this connection.
PLSQL_MEMORY_ADDRESS PL/SQL first connection attribute	The virtual address at which the shared memory segment is loaded into each process that uses the TimesTen direct drivers.
PLSQL_MEMORY_SIZE PL/SQL first connection attribute	The size in MB of the shared memory segment used by PL/SQL.
PLSQL_OPEN_CURSORS PL/SQL first connection attribute	The number of open cursors.
PLSQL_OPTIMIZE_LEVEL PL/SQL general connection attribute	The optimization level that the PL/SQL compiler uses to compile PL/SQL library units.
PLSQL_SESSION_CACHED_CURSORS PL/SQL general connection attribute	The number of session cursors to cache.
PLSQL_TIMEOUT PL/SQL general connection attribute	The number of seconds a PL/SQL procedure can run before being automatically terminated.
Preallocate Data store attribute	Specifies that disk space for the checkpoint files should be preallocated when creating the database.
PrivateCommands General connection attribute	Determines if commands are shared between connections.
PWD See UID and PWD General connection attribute	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.
PWDCrypt General connection attribute	The value of the encrypted user password.
PwdWallet General connection attribute	Provides the location and name of the wallet that contains stored user credentials needed for a connection.
QueryThreshold General connection attribute	Determines whether TimesTen returns a warning if a query times out before executing.
RACCallback Cache general connection attribute	Specifies whether to enable or disable the installation of Application Failover (TAF) and Fast Application Notification (FAN) callbacks.
RecoveryThreads First connection attribute	The number of threads used to rebuild indexes during recovery.
RedoParallelism First connection attribute	Specifies the number of threads used for database redo during recovery.
RedoSegmentSize First connection attribute	Specifies the size, in MB, of each segment into which the transaction log is divided.
ReplicationApplyOrdering Data store attribute	Enables automatic parallel replication.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Name and Type	Description
ReplicationParallelism Data store attribute	Specifies the number of tracks available for automatic parallel replication.
ReplicationTrack General connection attribute	Assigns a connection to a replication track.
ShmPagesNoTouch First connection attribute	Determines whether TimesTen touches all pages in the shared memory segment when loading the database.
ServersPerDSN Server first connection attribute	The desired number of TimesTen server processes for the DSN.
ServerStackSize Server first connection attribute	The size in kilobytes (KB) of the thread stack for each connection.
SQLQueryTimeout General connection attribute	Specifies the time limit in seconds within which the database should execute SQL statements.
SQLQueryTimeoutMSec General connection attribute	Specifies the time limit in milliseconds within which the database should execute SQL statements.
SSLClientAuthentication Client general connection attribute	Specifies whether SSL client authentication is required
SSLClientAuthentication Server first connection attribute	Specifies whether SSL client authentication is required
SSLRenegotiationPeriod Server first connection attribute	The time in minutes before which the client/server session is renegotiated.
SSLRenegotiationSize Server first connection attribute	The data transfer size in MB before which the client/server session is renegotiated.
StandbyNetServiceName Cache general connection attribute	The Oracle Service Name of the standby Oracle database instance from which data is to be loaded into a TimesTen database. This attribute is only used by the cache agent in an Oracle Active Data Guard configuration. Set the `StandbyNetServiceName` attribute to the standby Oracle Service Name.
TCP_Port Client general connection attribute	The port number on which the TimesTen server is listening.
TCP_Port2, TCP_PortN Client general connection attributes	For TimesTen, the port number on which the TimesTen server should listen if an automatic failover occurs.
TempSize First connection attribute	The size in MB for the temporary partition of the database.
TempWarnThreshold General connection attribute	The threshold at which TimesTen returns a warning when the temporary partition of the database is low in memory.
TransactionTimeout General connection attribute	Specifies the time limit in seconds for user transactions to complete.
TTC_ConnectTimeout Client general connection attribute	Number of seconds for the client to wait for a connect or disconnect call.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Name and Type	Description
TTC_FailoverPortRange Client general connection attribute	A range for the failover port numbers.
TTC_NetMsgMaxBytes Client general connection attribute	Specifies the maximum size in bytes of a client result set buffer.
TTC_NetMsgMaxRows Client general connection attribute	Specifies the maximum number of rows of a client result set buffer.
TTC_NoReconnectOnFailover Client general connection attribute	Specifies if TimesTen performs an automatic reconnect after all other failover procedures are completed.
TTC_Random_Selection Client general connection attribute	Specifies that the client randomly selects an alternate server from the list provided or selects the server according to `TTC_ServerN` settings.
TTC_Server or TTC_Server1 Client general connection attributes	Name of the computer where the TimesTen Server is running or a logical TimesTen server name.
TTC_Server_DSN Client general connection attribute	Server DSN corresponding to the TimesTen database.
TTC_Server_DSN2, TTC_Server_DSNn Client general connection attributes	For TimesTen, server DSN corresponding to the TimesTen database, if an automatic failover occurs.
TTC_Server2, TTC_ServerN Client general connection attributes	If an automatic failover occurs, the name of the system where the TimesTen Server should be running or a logical TimesTen server name.
TTC_SqlTimeoutMS Client general connection attribute	The time (in milliseconds) to wait for a response from the client.
TTC_TCP_KEEPALIVE_INTVL_MS Client general connection attribute	The time interval (in milliseconds) between subsequential probes.
TTC_TCP_KEEPALIVE_PROBES Client general connection attribute	The number of unacknowledged probes to send before considering the connection as failed and notifying the client.
TTC_TCP_KEEPALIVE_TIME_MS Client general connection attribute	The duration time (in milliseconds) between the last data packet sent and the first probe.
TTC_Timeout Client general connection attribute	Timeout period, in seconds, for completion of a TimesTen client/server operation.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Name and Type	Description
UID See UID and PWD. General connection attribute	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.
UseCacheConnPool Cache database attribute	Enables or disables the cache connection pool.
WaitForConnect General connection attribute	Specifies that the connection attempt should wait if an immediate connection is not possible.
Wallet Client general connection attribute Server first connection attribute	The fully qualified path to the directory where the wallet is stored.

General Use Connection Attributes

These are the TimesTen attributes related to essential specifications and behaviors of a connection.

In this guide, the general use connection attributes are grouped into three categories:

- Data Store Attributes
- First Connection Attributes
- General Connection Attributes

See the List of Connection Attributes for a comprehensive list of the connection attributes in TimesTen, their types, descriptions, and default values.

Data Store Attributes

Data store attributes define the database's connection name, character set, and other essential specifications. These attributes can be assigned values only during database creation by the instance administrator.

The data store attributes are described in detail next.

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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Required Privilege

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

Setting

Set `Data Source Name` as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`DSN`	A name that describes the DSN.
Windows ODBC Data Source Administrator	Not applicable	

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.

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:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`DatabaseCharacterSet`	Specify the preferred character set.
Windows ODBC Data Source Administrator	Not applicable	

Supported Character Sets

The following tables describe the character sets supported in TimesTen.

Asian Character Sets

Name	Description
`JA16EUC`	`EUC` 24-bit Japanese
`JA16EUCTILDE`	The same as `JA16EUC` except for the way that the wave dash and the tilde are mapped to and from Unicode
`JA16SJIS`	`Shift-JIS` 16-bit Japanese
`JA16SJISTILDE`	The same as `JA16SJIS` except for the way that the wave dash and the tilde are mapped to and from Unicode
`KO16KSC5601`	`KSC5601` 16-bit Korean
`KO16MSWIN949`	Microsoft Windows Code Page 949 Korean
`TH8TISASCII`	Thai Industrial Standard 620-2533 - ASCII 8-bit
`VN8MSWIN1258`	Microsoft Windows Code Page 1258 8-bit Vietnamese
`ZHS16CGB231280`	`CGB2312-80` 16-bit Simplified Chinese
`ZHS16GBK`	`GBK` 16-bit Simplified Chinese
`ZHS32GB18030`	`GB18030-2000`
`ZHT16BIG5`	`BIG5` 16-bit Traditional Chinese
`ZHT16HKSCS`	Microsoft Windows Code Page 950 with Hong Kong Supplementary Character Set `HKSCS-2001`. Character set conversion to and from Unicode is based on Unicode 3.0.
`ZHT16MSWIN950`	Microsoft Windows Code Page 950 Traditional Chinese

European Character Sets

Name	Description
`BLT8CP921`	Latvian Standard `LVS8-92(1)` Windows/UNIX/Linux 8-bit Baltic
`BLT8ISO8859P13`	ISO 8859-13 Baltic
`BLT8MSWIN1257`	Microsoft Windows Code Page 1257 8-bit Baltic
`BLT8PC775`	IBM-PC Code Page 775 8-bit Baltic
`CEL8ISO8859P14`	ISO 8859-13 Celtic
`CL8ISO8859P5`	ISO 8859-5 Latin/Cyrillic
`CL8KOI8R`	RELCOM Internet Standard 8-bit Latin/Cyrillic
`CL8KOI8U`	KOI8 Ukrainian Cyrillic

Copyright © 1996, 2026, Oracle and/or its affiliates.

Name	Description
CL8MSWIN1251	Microsoft Windows Code Page 1251 8-bit Latin/Cyrillic
EE8ISO8859P2	ISO 8859-2 East European
EL8ISO8859P7	ISO 8859-7 Latin/Greek
ET8MSWIN923	Microsoft Windows Code Page 923 8-bit Estonian
EE8MSWIN1250	Microsoft Windows Code Page 1250 8-bit East European
EL8MSWIN1253	Microsoft Windows Code Page 1253 8-bit Latin/Greek
EL8PC737	IBM-PC Code Page 737 8-bit Greek/Latin
EE8PC852	IBM-PC Code Page 852 8-bit East European
LT8MSWIN921	Microsoft Windows Code Page 921 8-bit Lithuanian
NE8ISO8859P10	ISO 8859-10 North European
NEE8ISO8859P4	ISO 8859-4 North and North-East European
RU8PC866	IBM-PC Code Page 866 8-bit Latin/Cyrillic
SE8ISO8859P3	ISO 8859-3 South European
US7ASCII	ASCII 7-bit American
US8PC437	IBM-PC Code Page 437 8-bit American
WE8ISO8859P1	ISO 8859-1 West European
WE8ISO8859P15	ISO 8859-15 West European
WE8MSWIN1252	Microsoft Windows Code Page 1252 8-bit West European
WE8PC850	IBM-PC Code Page 850 8-bit West European
WE8PC858	IBM-PC Code Page 858 8-bit West European

Middle Eastern Character Sets

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

Copyright © 1996, 2026, Oracle and/or its affiliates.

Universal Character Sets

Name	Description
AL16UTF16	Unicode 4.0 UTF-16 Universal character set. This is the implicit TimesTen national character set. This character set cannot be specified as a value to the DatabaseCharacterSet or ConnectionCharacterSet attributes.
AL32UTF8	Unicode 4.0 UTF-8 Universal character set
UTF8	Unicode 3.0 UTF-8 Universal character set, CESU-8 compliant

DataStore

The database path name uniquely identifies the physical database. It is the fully qualified directory path name of the database and the file name prefix, for example: `/disk1/databases/database1`. This name is not a file name. The actual database file names have suffixes, such as `.ds0` and `.log0`, for example `/disk1/databases/database1.ds0` and `/disk1/databases/database1.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:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`DataStore`	Full path to the physical database that the data source name references.
Windows ODBC Data Source Administrator	Not applicable	

DbEncrypted

The `DBEncrypted` attribute determines whether encryption at rest is enabled for checkpoint and transaction log files and specifies the preferred AES encryption algorithm to use for the master and data encryption keys.

TimesTen respects the setting for the `DbEncrypted` attribute for all operations that create a new checkpoint file, such as database creation, `ttRepAdmin -duplicate`, and `ttRestore`.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Required Privilege

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

Setting

Set `DbEncrypted` as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file	`DbEncrypted`	`none` (Default) - Encryption at rest is disabled. `AES128` - Enables encryption at rest and specifies AES-128 as the preferred encryption algorithm. `AES192` - Enables encryption at rest and specifies AES-192 as the preferred encryption algorithm. `AES256` - Enables encryption at rest and specifies AES-256 as the preferred encryption algorithm.
Windows ODBC Data Source Administrator	Not applicable	

DbKeyWallet

The `DbKeyWallet` attribute specifies the fully qualified directory path name where the Oracle wallet for the master key for encryption at rest resides. The wallet is stored in the `.wallet-`*`database_UUID`* directory within the specified path, where *`database_UUID`* is a unique, database-scoped string. The Oracle wallet is an auto-login wallet.

Required Privilege

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

Setting

Set `DbKeyWallet` as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file	`DbKeyWallet`	Specifies the directory where the wallet for encryption at rest resides. The default value is *`instance_home`*`/info/wallets`.
Windows ODBC Data Source Administrator	Not applicable	

Copyright © 1996, 2026, Oracle and/or its affiliates.

> **Note**
>
> Since the wallet is stored in a unique directory within the path specified by `DbKeyWallet`, more than one database can use the same `DbKeyWallet` path without issues.

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:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`Description`	Text description of the Data Source Name. This attribute is optional.
Windows ODBC Data Source Administrator	Not applicable	

Driver

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

For example, on Windows systems the value can be `TimesTen Client 22.1`.

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 Creating a DSN on Linux and UNIX for TimesTen in *Oracle TimesTen In-Memory Database Operations Guide*.

For general usage scenarios, refer to standard ODBC reference documentation, such as:

`https://docs.microsoft.com/en-us/sql/odbc/reference/syntax/odbc-api-reference`

Required Privilege

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

Setting

Set `Driver` as follows:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`Driver`	Specifies the path name for the TimesTen ODBC Driver shared library file, if using a Driver Manager.
Windows ODBC Data Source Administrator	Select a driver from the **Create New Data Source** dialog.	Specifies the client driver for TimesTen and the release.

LogDir

The `LogDir` attribute specifies the fully qualified directory path name 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:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`LogDir`	Specifies the directory where transaction log files reside.
Windows ODBC Data Source Administrator	Not applicable	

Preallocate

The `Preallocate` attribute determines whether TimesTen preallocates disk space for the database checkpoint files 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.

TimesTen respects the setting for the `Preallocate` attribute for all operations that create a new checkpoint file, such as database creation, `ttRepAdmin -duplicate`, and `ttRestore`.

Required Privilege

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

Setting

Set `Preallocate` as follows:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`Preallocate`	`0` - Does not preallocate disk space for checkpoint files when creating the database. 1 (Default) - Preallocates disk space for the checkpoint files.
Windows ODBC Data Source Administrator	Not applicable	

> **Note**
>
> Reallocating disk space for a large database is very time consuming.

ReplicationApplyOrdering

`ReplicationApplyOrdering` enables parallel replication.

When used with the ReplicationParallelism attribute, 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 *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 Improving AWT Throughput with Parallel Propagation to the Oracle Database in *Oracle TimesTen In-Memory Database Cache Guide*.

Required Privilege

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

Setting

Set `ReplicationApplyOrdering` as follows:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`ReplicationApplyOrdering`	`0` - Specifies automatic parallel replication. Automatic parallel replication is available for both classic and active standby pair replication schemes. (default) 2 - Specifies automatic parallel replication with disabled commit dependencies.
Windows ODBC Data Source Administrator	Not applicable	

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.

Set `ReplicationParallelism` to a number from 2 to 32. This number indicates the number of transmitter threads on the source database and the number of receiver threads on the target database. However, if you are using single-threaded replication, set `ReplicationParallelism` to 1, which is the default.

The LogBufParallelism and `ReplicationParallelism` connection attributes are related. `LogBufParallelism` specifies the number of strands that are mapped to the threads that are specified by `ReplicationParallelism`. For example, if `LogBufParallelism` = 4 and `ReplicationParallelism` = 4, then one strand is mapped to one thread. If `LogBufParallelism` = 8 and `ReplicationParallelism` = 4, then two strands are mapped to one thread.

Thus, if `ReplicationParallelism` is greater than 1, the `LogBufParallelism` connection attribute must be equal to or greater than the value of `ReplicationParallelism`. The `ReplicationParallelism` connection attribute cannot exceed the value of `LogBufParallelism`. In order for the number of strands to be equally distributed across the number of threads, you may want to make `LogBufParallelism` a multiple of the number of threads specified in `ReplicationParallelism`.

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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Setting

Set `ReplicationParallelism` as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`ReplicationParallelism`	*n* - A value between 1 and 32, indicating the number of tracks to replicate in parallel. The default is 1.
Windows ODBC Data Source Administrator	Not applicable	

Restrictions

When parallel replication is enabled, the Description column of the ttLogHolds built-in procedure displays one row per track per subscriber node.

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.

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.

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 .

The first connection attributes are described in detail next.

AutoCreate

When you connect to an nonexistent database, the `AutoCreate` connection attribute automatically creates that database.

With `AutoCreate` set, TimesTen creates the database, but not the path to the database. 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. 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.

Required Privilege

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

Copyright © 1996, 2026, Oracle and/or its affiliates.

Setting

Set `AutoCreate` as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`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	Not applicable	

BackupFailThreshold

The `BackupFailThreshold` connection attribute enables you to ensure the safe completion of your backup.

This connection attribute controls the number of transaction log files that accumulate in the directory defined by the value of the LogDir attribute after a backup starts before TimesTen releases the hold on checkpoint operations. If a checkpoint is initiated before the completion of a backup, the backup is invalidated.

TimesTen temporarily ignores the CkptFrequency and CkptLogVolume attributes (controlling background checkpoints) while a backup is in progress if this attribute is not set or is set to `0`.

Set the attribute to a value that is high enough to ensure the safe completion of your backup. For example, if a backup typically takes *n* seconds to complete and your database creates *m* transaction log files per second, set `BackupFailThreshold` to a value greater than *n*m*.

The number of log files generated by your database per any given unit of time is directly proportional to your write workload and inversely proportional to the value set for the LogFileSize attribute.

Required Privilege

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

Setting

Set `BackupFailThreshold` as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`BackupFailThreshold`	Enter a non-zero integer value that indicates the number of transaction log files that are allowed to accumulate before the backup fails. The default is `0`, indicating that transaction log files continue to accumulate with the backup operation is in process.
Windows ODBC Data Source Administrator	Not applicable	

Copyright © 1996, 2026, Oracle and/or its affiliates.

CkptFrequency

The `CkptFrequency` connection attribute 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.

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:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`CkptFrequency`	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.
Windows ODBC Data Source Administrator	Not applicable	

CkptLogVolume

The `CkptLogVolume` connection attribute 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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

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.

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:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`CkptLogVolume`	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.
Windows ODBC Data Source Administrator	Not applicable	

CkptRate

The `CkptRate` connection attribute 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

Copyright © 1996, 2026, Oracle and/or its affiliates.

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:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`CkptRate`	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.
Windows ODBC Data Source Administrator	Not applicable	

CkptReadThreads

The `CkptReadThreads` connection attribute 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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Setting

Set `CkptReadThreads` as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`CkptReadThreads`	*n* - The number of threads to use when reading the checkpoint files during the loading of the database into memory. It takes an integer value from `0` to `64`. Default is `1`.
Windows ODBC Data Source Administrator	Not applicable	

> **Note**
>
> For a progress report on a recovery process, see the rebuild messages in the daemon log file. Set the number of threads low enough to leave sufficient resources on the TimesTen server for other services/processes.

Connections

This attribute 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.

> **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*.

Required Privilege

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

Copyright © 1996, 2026, Oracle and/or its affiliates.

Setting

Set `Connections` as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`Connections`	The default value is the lesser of `2000` or the number of semaphores specified in the `SEMMNS` kernel parameter minus `155`. Supported value is an integer from 1 through `32000`. The value represents the maximum number of connections.
Windows ODBC Data Source Administrator	Not applicable	

DefragThreshold

The `DefragThreshold` attribute defines the level of online defragmentation. Higher values trigger more intense defragmentation, targeting row pages even when they contain a relatively small number of rows. It leads to more space being freed.

The acceptable values range from 1 to 3 with each value corresponding to a specific level of defragmentation sensitivity. For details, see Online Defragmentation of TimesTen Databases in the *Oracle TimesTen In-Memory Database Operations Guide*.

The defragmentation process obtains table locks and row locks while executing defragmentation. Higher `DefragThreshold` values cause more pages to be defragmented, as more row pages become eligible for defragmentation. As a result, table and row locks are acquired more often, which increases the likelihood of lock contention or conflicts with concurrent transactions.

Required Privilege

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

Setting

Set `DefragThreshold` as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`DefragThreshold`	• 1 - low • 2 - medium (default) • 3 - high

> **Note**
>
> You can view or modify the setting using `ttDBConfig` built-in. See ttDBConfig.

Copyright © 1996, 2026, Oracle and/or its affiliates.

DefragSpeed

The DefragSpeed attribute enables you to ensure how fast the online defragmentation is done.

For lower DefragSpeed values, the defragmentation process runs slowly using less CPU time. When the DefragSpeed value is set to 0, online defragmentation is disabled.

Higher values lead to faster defragmentation and use more CPU resources, which leads to a greater chance of lock contention. This is due to defragmentation running on a single thread. For details, see Online Defragmentation of TimesTen Databases in the *Oracle TimesTen In-Memory Database Operations Guide*.

Required Privilege

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

Setting

Set DefragSpeed as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems odbc.ini file in TimesTen	DefragSpeed	• 0 - disabled (default) • 1 - low • 2 - medium • 3 - high

> **Note**
>
> You can view or modify the setting using ttDBConfig built-in. See ttDBConfig.

DefragCycle

The DefragCycle specifies the number of seconds to wait before the next cycle of the online defragmentation.

If DefragCycle=0, online defragmentation keeps scanning tables for fragmentation, even when no fragmentation is present in the tables. This leads to unnecessary consumption of CPU resources.

If you set DefragCycle to a non-zero value, it can help reduce the CPU usage.

If you change the DefragCycle value while the defragmentation is waiting for the next DefragCycle, the next round of defragmentation is performed after the number of seconds specified by the new DefragCycle. For example, if you set the DefragCycle value to 60 and update to 120 after 30 seconds have passed since the last defragmentation, TimesTen schedules the next defragmentation round in 120 seconds instead of the remaining 30 seconds. For details, see Online Defragmentation of TimesTen Databases in the *Oracle TimesTen In-Memory Database Operations Guide*.

Required Privilege

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

Copyright © 1996, 2026, Oracle and/or its affiliates.

Setting

Set `DefragCycle` as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`DefragCycle`	Default value: `0` Maximum value: `86400` seconds

> **Note**
>
> You can view or modify the setting using `ttDBConfig` built-in. See ttDBConfig.

EnableParallelConnect

The `EnableParallelConnect` attribute enables connections to a TimesTen database to be created in parallel. However, multiple threads within the same process connect sequentially. Only separate processes can establish connections concurrently.

This behavior matters most in client/server mode when you use the `MaxConnsPerServer` attribute. `MaxConnsPerServer` has a default setting of `1`, so each client connection uses its own TimesTen server process. When you increase this value, a single server process handles multiple client connections and creates a separate thread for each one.
With `EnableParallelConnect` set to `1`, the TimesTen database accepts parallel connections. Even when you enable `EnableParallelConnect`, threads within the same server process connect one at a time. Parallel connections occur only across different server processes, not across threads in a single process.

If you set the value `0`, connections to the TimesTen database will be sequential. Sequential connections may result in multiple processes trying to connect at the same time, resulting in processeses waiting to connect.

Required Privilege

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

Setting

Set `EnableParallelConnect` as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen Classic	`EnableParallelConnect`	`0` - Do not allow parallel connections to the database. 1 (default) - Allow parallel connections to the database, so that multiple processes can connect at the same time.
Windows ODBC Data Source Administrator	Not applicable	

Copyright © 1996, 2026, Oracle and/or its affiliates.

EnableParallelRedo

The parallel redo feature improves database recovery by using multiple CPU threads to apply redo transaction logs simultaneously instead of sequentially.

If you set the `EnableParallelRedo` attribute to `1`, parallel redo is enabled. Setting `EnableParallelRedo=0` disables parallel redo, and the database uses the legacy redo method instead. If the new default parallel redo method fails or lengthens the recovery time, you can disable it by setting the connection attribute `EnableParallelRedo=0`. The default value is `1`.

When parallel redo is enabled:

- The transaction log is divided into segments based on the `RedoSegmentSize` connection attribute. See RedoSegmentSize.
- Within each segment, multiple redo recovery threads (as defined by `RedoParallelism`) reapply the transaction log records concurrently. See RedoParallelism.
- Once all redo records in one segment are processed, parallel redo proceeds to read and process the next segment until recovery completes.

For details on how to configure this attribute, see Parallel Redo Recovery in TimesTen Database in the *Oracle TimesTen In-Memory Database Operations Guide*.

Required Privilege

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

Setting

Set `EnableParallelRedo` as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`EnableParallelRedo`	`0` - Disable `1` - Enable (default)
Windows ODBC Data Source Administrator	Not applicable	Not applicable

ForceConnect

Specifies whether a connection is allowed to a failed database if it is not properly restored from the corresponding subscriber database.

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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Required Privilege

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

Setting

Set `ForceConnect` as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen Classic	`ForceConnect`	`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.
Windows ODBC Data Source Administrator	Not applicable	

ForceDisconnectEnabled

Allows you to force disconnection from the database using `ttAdmin -disconnect`.

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

> **Note**
>
> - 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. See Force Disconnect.
> - 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:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen Classic	`ForceDisconnectEnabled`	`0` (default) - Do not allow forced disconnections. 1 - Allow forced disconnections.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Where to set the attribute	How the attribute is represented	Setting
Windows ODBC Data Source Administrator	N/A	

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:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`LogAutoTruncate`	`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.
Windows ODBC Data Source Administrator	Not applicable	

LogBufMB

Specifies the size of the internal transaction log buffer in megabytes.

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

The recommendation is to set `LogBufMB` to a value between 256 MB and 4 GB. If memory space is a concern, start with 256 MB; otherwise, start with 1 GB.

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`).

Copyright © 1996, 2026, Oracle and/or its affiliates.

Required Privilege

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

Setting

Set `LogBufMB` as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`LogBufMB`	*n* - Size of log buffer in megabytes. 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).
Windows ODBC Data Source Administrator	Not applicable	

Examples

The following shows how to see the value of `LogBufMB`.

```
Command> CALL ttConfiguration('LogBufMB');
< LogBufMB, 64 >
1 row found.
```

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.

Strands divide the transaction log buffer available memory into a number of different regions, which can be accessed concurrently by different connections. Each connection can execute data-independent DML statements in parallel using those strands as if each has its own transaction log buffer.

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:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`LogBufParallelism`	An integer value between 1 and 64. The default is 4.
Windows ODBC Data Source Administrator	Not applicable	

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 >= LogBufMB`).

Required Privilege

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

Setting

Set `LogFileSize` as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`LogFileSize`	*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).
Windows ODBC Data Source Administrator	Not applicable	

Copyright © 1996, 2026, Oracle and/or its affiliates.

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 1 if most of your transactions commit durably and use the value 0 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:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`LogFlushMethod`	`0` Write data to transaction log files using synchronous writes such that explicit sync operations are not needed. `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).
Windows ODBC Data Source Administrator	Not applicable	

Also see 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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

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:

Where to set the attributes	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`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	Not applicable	

MemoryLock

Enables applications that connect to a shared database to specify whether the real memory should be locked during database loading.

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.

> **Note**
>
> If you are using systemd, set the `LimitMEMLOCK` setting in the `#Service Limit Settings` section of the service unit configuration file as specified in Using systemd to Manage a TimesTen Service in *Oracle TimesTen In-Memory Database Installation, Migration, and Upgrade Guide* to control the amount of memory a user can lock.

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

Copyright © 1996, 2026, Oracle and/or its affiliates.

```
% ttDaemonadmin -restart
```

Setting

Set MemoryLock as follows.

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems odbc.ini file in TimesTen	MemoryLock	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.
Windows ODBC Data Source Administrator	Not applicable	

Overwrite

Specifies that the existing database should be overwritten with a new one when a connection is attempted.

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). 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:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`Overwrite`	`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.
Windows ODBC Data Source Administrator	N/A	

PermSize

Indicates the size in MB for the permanent partition of the database.

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

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. 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 Memory Region Sizes 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:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`PermSize`	*n* - Size of permanent partition in megabytes; default is 32 MB.
Windows ODBC Data Source Administrator	Not applicable	

Copyright © 1996, 2026, Oracle and/or its affiliates.

ShmPagesNoTouch

The ShmPagesNoTouch connection attribute prevents TimesTen from "touching pages" when creating a shared segment. In the database context, touching pages means a query or operation accesses or modifies specific data pages in memory.

By default, TimesTen pre-touches all pages in the shared memory segment during database initialization. In this context, touching a page means the database reads and writes to it, so the operating system immediately assigns physical memory to it. This process commits the entire memory footprint at startup and ensures the database has dedicated RAM before it starts processing requests. It may increase initialization time for large databases, but it does not increase total memory usage; it validates and commits memory earlier rather than during runtime.
Linux allocates physical memory only when a process first accesses a page. If you turn off page touching, TimesTen allocates memory on demand when it references those pages, rather than committing memory during initialization. This allocation reduces startup time and lowers immediate memory usage, but it also delays memory validation until the database is operating under load.

The ShmPagesNoTouch connection attribute turns off the pre-touch phase and enables demand-based allocation. This configuration reduces the chance of running out of memory during startup, but it increases the risk of runtime failures if the system runs out of physical memory. In that situation, the Linux Out-of-Memory (OOM) killer may terminate the TimesTen process, potentially invalidating the database. Because of this risk, configure ShmPagesNoTouch connection attribute to 1 only when the system can guarantee sufficient free memory throughout the database's lifecycle.
Before changing the value of the attribute to 1 to turn off page touching when you are experiencing an issue with the OOM Killer, Oracle recommends trying these steps first:

1. Configure the Linux kernel parameters for shared memory settings. See Configure shmmax and shmall in *Oracle TimesTen In-Memory Database Installation, Migration, and Upgrade Guide*.
2. Reduce OOM priority for the processes connecting to the database, such as timestend, timestensubd, timestenrepd, and timestenorad. See How to Configure the Linux Out-of-Memory Killer.

Required Privilege

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

Setting

Set ShmPagesNoTouch as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems odbc.ini file in TimesTen Classic	ShmPagesNoTouch	0 (default) - Allows TimesTen to touch all pages in the shared memory segment when loading the database. 1 - Prevents TimesTen from touching all pages in the shared memory segment when loading the database. Do not set 1 on non- Linux systems.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Where to set the attribute	How the attribute is represented	Setting
Windows ODBC Data Source Administrator	Not applicable	

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:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`RecoveryThreads`	*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.
Windows ODBC Data Source Administrator	Not applicable	

> **Note**
>
> For a progress report on the recovery process, see the rebuild messages in the daemon log.
>
> Set the number of threads low enough to leave sufficient resources on the TimesTen server for other services/processes.

Copyright © 1996, 2026, Oracle and/or its affiliates.

RedoParallelism

The `RedoParallelism` attribute controls the degree of parallelism. It means how many RedoTracks (threads) TimesTen uses to process redo logs.

This attribute takes effect only when you enable `EnableParallelRedo` (see EnableParallelRedo).

The values range from `1` to `64` and the default value is `4`. If you set `RedoParallelism` to `2`, it means two redo threads (RedoTracks) process redo records within each segment concurrently (see RedoSegmentSize). Set `RedoParallelism=1` to run the parallel redo algorithm in serial mode. The higher the value, the more CPU threads are utilized for recovery.

For details on how to configure this attribute, see Parallel Redo Recovery in TimesTen Database in the *Oracle TimesTen In-Memory Database Operations Guide*.

Required Privilege

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

Setting

Set `RedoParallelism` as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`RedoParallelism`	`1` - Minimum `4` - Default `64` - Maximum
Windows ODBC Data Source Administrator	Not applicable	Not applicable

RedoSegmentSize

The `RedoSegmentSize` attribute determines the size, in MB, of the segments used to divide the transaction log for parallel redo processing.

Each segment is independent, allowing the parallel redo transaction log cursor to read and process one segment at a time. You can specify the segment size using the `RedoSegmentSize` connection attribute, which sets the size of each segment in megabytes (MB). The transaction log is divided into segments irrespective of parallel redo is enabled or disabled (see EnableParallelRedo). The `RedoSegmentSize` value becomes effective once the database is up and running. If the database becomes invalid and you change the value of `RedoSegmentSize`, the new value will take effect after the recovery process is complete.

For details on how to configure this attribute, see Parallel Redo Recovery in TimesTen Database in the *Oracle TimesTen In-Memory Database Operations Guide*.

Required Privilege

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

Setting

Set `RedoSegmentSize` as follows:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`RedoSegmentSize`	`8MB` - Minimum `LogBufMB` - Maximum `20MB` - Default
Windows ODBC Data Source Administrator	Not applicable	Not applicable

TempSize

Indicates the size in MB for the temporary partition of the database.

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

`TempSize` has no predefined value.

If left unspecified, the `TempSize` 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.

In TimesTen, the minimum `TempSize` is 32 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.

Required Privilege

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

Setting

Set `TempSize` as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`TempSize`	*n* - Size of the temporary partition, in MB. Minimum size is 32 MB for TimesTen.
Windows ODBC Data Source Administrator	Not applicable	

General Connection Attributes

General connection attributes are set by each connection and persist for the duration of the connection.

The general connection attributes are described in detail next.

Copyright © 1996, 2026, Oracle and/or its affiliates.

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 `CommitBufferSizeMax` as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`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	Not applicable	

> **Note**
>
> 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 enables you to attach a symbolic name to any database connection. Connection names are unique within a process.

This attribute is also available as a client connection attribute.

Copyright © 1996, 2026, Oracle and/or its affiliates.

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:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`ConnectionName`	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.
Windows ODBC Data Source Administrator	**Connection** field on the Oracle TimesTen client DSN Setup dialog	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.

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.

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

```
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

Copyright © 1996, 2026, Oracle and/or its affiliates.

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:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen Classic	`DDLReplicationAction`	`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.
Windows ODBC Data Source Administrator	Not applicable	

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 3 (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 `ttCacheUidPwdSet` built-in procedure. You do not need to stop and restart the cache agent or replication agent to execute the `ttCacheUidPwdSet`

Copyright © 1996, 2026, Oracle and/or its affiliates.

built-in procedure. For more information, see Changing Cache User Names and Passwords in *Oracle TimesTen In-Memory Database Cache Guide* or ttCacheUidPwdSet.

Some things to be aware of when setting this attribute to are:

- If set to 0, open cursors are automatically closed with the implicit commit that occurs in a transaction that contains a DDL statement. You should not use cursors in this way in DDL transactions.

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 *Oracle TimesTen In-Memory Database Replication Guide*.

To learn more about replicating DDL, see Making DDL Changes in an Active Standby Pair in *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:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen Classic	`DDLReplicationLevel`	`1` - Replicates `ALTER TABLE ADD or DROP COLUMN` to the standby database. Does not replicate `CREATE` and `DROP` operations for tables, indexes, or synonyms to the standby database. 2 - Replicates creating and dropping of tables, indexes and synonyms. 3 (default) - Replicates creating and dropping of views and sequences and replicates the results of the `ttCacheUidPwdSet` built-in procedure.
Windows ODBC Data Source Administrator	Not applicable	

Restrictions

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

Copyright © 1996, 2026, Oracle and/or its affiliates.

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 ttCacheUidPwdSet 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:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems odbc.ini file in TimesTen	Diagnostics	0 - TimesTen does not generate diagnostic messages. 1 (default) - TimesTen generates base-level diagnostics messages.
Windows ODBC Data Source Administrator	Not applicable	

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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Required Privilege

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

Setting

Set `DurableCommits` as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen Classic	`DurableCommits`	`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 applicable	

Also see LogFlushMethod.

IncludeInCore

The `IncludeInCore` 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:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`IncludeInCore`	`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.
Windows ODBC Data Source Administrator	N/A	

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 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.

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_221`:

1. Call `ttIsql` and connect to the `DSN` with `PassThrough` level `3`:

   ```
   % ttIsql;
   Command> connect "dsn=repdb1_221;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;
   ```


Copyright © 1996, 2026, Oracle and/or its affiliates.

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:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`Isolation`	`0` - Connects to database in serializable isolation mode. 1 (default) - Connects to database in read committed mode.
Windows ODBC Data Source Administrator	Not applicable	

LockLevel

Specifies whether the connection should use row-level locking (value = 0) or database-level locking (value = 1).

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 ttLockLevel built-in procedure. Connections can also wait for unavailable locks by calling the 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 ttXactAdmin utility.

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

Required Privilege

`ADMIN` privilege is required if the value of this attribute is 1.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Setting

Set `LockLevel` as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`LockLevel`	`0` (default) - Transactions access the database using row-level locking. `1` - Transactions access the database by acquiring an exclusive lock on the entire database.
Windows ODBC Data Source Administrator	Not applicable	

LockWait

The `LockWait` attribute 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:

```
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:

```
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:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`LockWait`	*s* - Indicates the number of seconds to wait for locking conflict resolution before timing out. The default is `10` seconds.
Windows ODBC Data Source Administrator	Not applicable	

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 or comments.

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:

```
TT_INDEX (t1,i1, 0; t2, i2,0)
```

use

```
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:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`OptimizerHint`	A string specifying optimizer hints and their values. The maximum length of the string is 512.
Windows ODBC Data Source Administrator	Not applicable	

Copyright © 1996, 2026, Oracle and/or its affiliates.

PermWarnThreshold

The `PermWarnThreshold` attribute 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:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`PermWarnThreshold`	*p* - Percentage at which TimesTen should issue out-of-memory warnings. Default is `90`.
Windows ODBC Data Source Administrator	Not applicable	

PrivateCommands

Determines if commands are shared between connections.

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:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`PrivateCommands`	`0` (default) - Commands are shared with other connections. `1` - Commands are not shared with any other connection.
Windows ODBC Data Source Administrator	Not applicable	

Copyright © 1996, 2026, Oracle and/or its affiliates.

Note

- 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.
- TimesTen does not store the value of the `PWD` attribute anywhere in the TimesTen system.
- If you are not using `PwdWallet` to specify a wallet and if you want to provide an encrypted password, use `PWDCrypt` instead of `PWD` to specify an encrypted password that corresponds with the specified UID.

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:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`PWDCrypt`	Enter the value generated by the ttUser utility.
Windows ODBC Data Source Administrator	**PWDCrypt** field on the Oracle TimesTen client DSN Setup dialog	Enter the value generated by the ttUser utility.

PwdWallet

TimesTen enables you to store user names and associated passwords in an Oracle Wallet. This is the most secure and preferred method of providing credentials for connecting to a TimesTen database. The `PwdWallet` connection attribute is the path to the location of the wallet, from which TimesTen retrieves the password for the specified user name.
You specify the wallet from which to retrieve credentials for your connection using the UID and `PwdWallet` connection attributes. You provide the user name to identify which credentials to retrieve from the wallet for the user that is specified.

```
Connect “dsn=mydb;uid=terry;PwdWallet=/home/terry/wallets/mywallet”;
```

Notes

- To use `PwdWallet` for client/server connections, the wallet must exist on the client. You get an error if you attempt to retrieve credentials from a wallet located in the server.
- On the client side, the client process reads the stored credentials for the user ID and uses them to connect to the TimesTen server.
- If you are not using `PwdWallet` to specify a wallet, then use the PWD connection attribute to specify the password that corresponds with the specified UID.
- For more information on creating the wallet and managing Oracle cache administration user IDs and passwords, see Providing a User Name and Password in an Oracle Wallet in *Oracle TimesTen In-Memory Database Security Guide*.

Required Privilege

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

Setting

Set `PwdWallet` as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`PwdWallet`	Absolute path and name of the wallet that stores the cache administration user name and password.
Windows ODBC Data Source Administrator	**PwdWallet** field on the Oracle TimesTen client DSN Setup dialog.	

Also see ttUser

Copyright © 1996, 2026, Oracle and/or its affiliates.

QueryThreshold

Use this attribute to write a warning to the daemon 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:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`QueryThreshold`	A non-negative integer. Default is `0` and indicates that TimesTen does not return a warning.
Windows ODBC Data Source Administrator	Not applicable	

ReplicationTrack

When managing track-based parallel replication, this attribute assigns 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:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`ReplicationTrack`	*n* - An integer between 1 and 64 that specifies the replication track to be used by transactions issued by the connection.
Windows ODBC Data Source Administrator	Not applicable	

SQLQueryTimeout

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

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

Both `SQLQueryTimeout` and SQLQueryTimeoutMSec attributes are internally mapped to one timeout value in milliseconds. If different values are specified for these attributes, only one value is retained.

The `SQLQueryTimeout`/`SQLQueryTimeoutMsec` value should be less than the `TransactionTimeout` value; the `TransactionTimeout` value should be less than the `TTC_Timeout` value.

For more details, see Choose SQL and PL/SQL Timeout Values in the *Oracle TimesTen In-Memory Database Operations Guide*.

> **Note**
>
> - When SQL query timeouts are used (`SQLQueryTimeout` or `SQLQueryTimeoutMsec`), TimesTen behavior is on a best-effort basis. It is not possible to guarantee that the timeout will actually occur within the specified time.
> - SQL query timeouts are honored during dynamic load unless the dynamic load requires a new connection to the Oracle database, in which case the connection is allowed to complete. If the connection completes successfully but the dynamic load times out, the connection will be retained.

Required Privilege

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

Setting

Set `SQLQueryTimeout` as follows:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`SQLQueryTimeout`	*n* - Time limit in seconds for which the database should execute SQL queries. The value of `n` 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.
Windows ODBC Data Source Administrator	Not applicable	

SQLQueryTimeoutMSec

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

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

Both SQLQueryTimeout and `SQLQueryTimeoutMsec` attributes are internally mapped to one timeout value in milliseconds. If different values are specified for these attributes, only one value is retained.

The `SQLQueryTimeout`/`SQLQueryTimeoutMsec` value should be less than the `TransactionTimeout` value; the `TransactionTimeout` value should be less than the `TTC_Timeout` value.

For more details, see Choose SQL and PL/SQL Timeout Values in the *Oracle TimesTen In-Memory Database Operations Guide*.

> **Note**
>
> - When SQL query timeouts are used (`SQLQueryTimeout` or `SQLQueryTimeoutMsec`), TimesTen behavior is on a best-effort basis. It is not possible to guarantee that the timeout will actually occur within the specified time.
> - SQL query timeouts are honored during dynamic load unless the dynamic load requires a new connection to the Oracle database, in which case the connection is allowed to complete. If the connection completes successfully but the dynamic load times out, the connection will be retained.

Required Privilege

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

Setting

Set `SQLQueryTimeoutMsec` as follows:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`SQLQueryTimeoutMSec`	*n* - Time limit in milliseconds for which the database should execute SQL queries. The value of `n` 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.
Windows ODBC Data Source Administrator	Not applicable	

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.

Required Privilege

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

Setting

Set `TempWarnThreshold` as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`TempWarnThreshold`	*p* - Percentage at which warning should be issued. Default is `90`.
Windows ODBC Data Source Administrator	Not applicable	

TransactionTimeout

Use this attribute to specify the time limit in seconds for a user transaction to complete.

If a transaction times out, TimesTen returns an error indicating that either:

- TimesTen rolled back the transaction on behalf of the user, or
- The user must roll back the transaction.

The `SQLQueryTimeout`/`SQLQueryTimeoutMsec` value should be less than the TransactionTimeout value; the `TransactionTimeout` value should be less than the `TTC_Timeout` value.

Copyright © 1996, 2026, Oracle and/or its affiliates.

> **Note**
>
> - When a transaction timeout is provided, TimesTen behavior is on a best-effort basis. It is not possible to guarantee that the timeout will actually occur within the specified time.
> - Transaction timeouts are honored during the active phases of the transaction. However, transactions do not timeout during non-active phases, such as commit, reclaim or in-doubt transaction recovery.

Required Privilege

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

Setting

Set `TransactionTimeout` as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`TransactionTimeout`	*n* - Time limit in seconds for a user transaction to complete. The value of `n` can be any integer equal to or greater than `0`. The default value is `0`, indicating that the transaction does not time out.
Windows ODBC Data Source Administrator	Not applicable	

UID and PWD

`UID` specifies a user name defined on the TimesTen server, while `PWD` specifies the password corresponding to that UID.

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 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.

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

Copyright © 1996, 2026, Oracle and/or its affiliates.

> **Note**
>
> You can set user names and passwords in connection strings or DSN definitions with the UID and PWD connection attributes, and encrypt passwords with the PWDCrypt connection attribute. However, we recommend you store your passwords in wallets. See ttUser for more information.

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:

```
% ttIsql -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:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems odbc.ini file in TimesTen	UID	Character string specifying the user ID.
C or Java programs or UNIX and Linux systems odbc.ini file in TimesTen	PWD	Character string specifying the password that corresponds to the user ID.
Windows ODBC Data Source Administrator	**User ID** and **Password** fields on the Oracle TimesTen Client DSN Setup dialog	Character string specifying the user ID.

WaitForConnect

The WaitForConnect connection attribute specifies that the connection attempt should wait if an immediate connection is not possible.

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 usually waits

Copyright © 1996, 2026, Oracle and/or its affiliates.

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 *Oracle TimesTen In-Memory Database Error Help*.

Required Privilege

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

Setting

Set `WaitForConnect` as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`WaitForConnect`	`0` - Does not wait if connection to database fails. `1` (default) - Waits until connection to database is possible.
Windows ODBC Data Source Administrator	Not applicable	

Globalization Connection Attributes

These attributes control the globalization behaviors of the database.

The attributes are listed in The National Language Support (NLS) General Connection Attributes section.

See the List of Connection Attributes for a comprehensive list of the connection attributes in TimesTen, their types, descriptions, and default values.

NLS General Connection Attributes

The National Language Support (NLS) connection attributes set the default length semantics configuration, determine whether an error caused by data loss is reported, and what collating sequence for linguistic comparisons should be used.

These attributes are set by each connection and persist for the duration of the connection, but 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.

The NLS general connection attributes are described in detail next.

ConnectionCharacterSet

Specifies the character encoding for the connection and it is also available as a Client connection attribute.

The character encoding for the connection 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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

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.

Required Privilege

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

Setting

Set `ConnectionCharacterSet` as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`ConnectionCharacterSet`	The default value for `ConnectionCharacterSet` is `US7ASCII`.
Windows ODBC Data Source Administrator	**Connection CharacterSet** list field on the Oracle TimesTen Client DSN Setup dialog	The default value for `ConnectionCharacterSet` is `US7ASCII`.

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:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`NLS_LENGTH_ SEMANTICS`	Specify either `BYTE` (default) or `CHAR`.
Windows ODBC Data Source Administrator	Not applicable	

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, and 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:

Where to Set the Attribute	How the Attribute is Represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`NLS_NCHAR_CONV_EXCP`	`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.
Windows ODBC Data Source Administrator	Not applicable	

NLS_SORT

The `NLS_SORT` attribute indicates which collating sequence to use for linguistic comparisons.

It accepts the monolingual and multilingual values from the Supported Linguistic Sorts tables. 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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

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:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems odbc.ini file in TimesTen	NLS_SORT	Specify the linguistic sort sequence or BINARY (default).
Windows ODBC Data Source Administrator	Not applicable	

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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Monolingual Linguistic Sorts

Basic name	Extended name
ARABIC	N/A
ARABIC_MATCH	N/A
ARABIC_ABJ_SORT	N/A
ARABIC_ABJ_MATCH	N/A
ASCII7	N/A
AZERBAIJANI	XAZERBAIJANI
BENGALI	N/A
BIG5	N/A
BINARY	N/A
BULGARIAN	N/A
CANADIAN FRENCH	N/A
CATALAN	XCATALAN
CROATIAN	XCROATIAN
CZECH	XCZECH
CZECH_PUNCTUATION	XCZECH_PUNCTUATION
DANISH	XDANISH
DUTCH	XDUTCH
EBCDIC	N/A
EEC_EURO	N/A
EEC_EUROPA3	N/A
ESTONIAN	N/A
FINNISH	N/A
FRENCH	XFRENCH
GERMAN	XGERMAN
GERMAN_DIN	XGERMAN_DIN
GBK	N/A
GREEK	N/A
HEBREW	N/A
HKSCS	N/A
HUNGARIAN	XHUNGARIAN
ICELANDIC	N/A
INDONESIAN	N/A
ITALIAN	N/A
LATIN	N/A
LATVIAN	N/A
LITHUANIAN	N/A

Copyright © 1996, 2026, Oracle and/or its affiliates.

Basic name	Extended name
MALAY	N/A
NORWEGIAN	N/A
POLISH	N/A
PUNCTUATION	XPUNCTUATION
ROMANIAN	N/A
RUSSIAN	N/A
SLOVAK	XSLOVAK
SLOVENIAN	XSLOVENIAN
SPANISH	XSPANISH
SWEDISH	N/A
SWISS	XSWISS
THAI_DICTIONARY	N/A
TURKISH	XTURKISH
UKRAINIAN	N/A
UNICODE_BINARY	N/A
VIETNAMESE	N/A
WEST_EUROPEAN	XWEST_EUROPEAN

Multilingual Linguistic Sorts

Sort Name	Description
CANADIAN_M	Canadian French sort supports reverse secondary, special expanding characters.
DANISH_M	Danish sort supports sorting uppercase characters before lowercase characters.
FRENCH_M	French sort supports reverse sort for secondary.
GENERIC_M	Generic sorting order which is based on ISO14651 and Unicode canonical equivalence rules but excluding compatible equivalence rules.
JAPANESE_M	Japanese sort supports SJIS character set order and EUC characters which are not included in SJIS.
KOREAN_M	Korean sort Hangul characters are based on Unicode binary order. Hanja characters based on pronunciation order. All Hangul characters are before Hanja characters.
SPANISH_M	Traditional Spanish sort supports special contracting characters.
THAI_M	Thai sort supports swap characters for some vowels and consonants.
SCHINESE_RADICAL_M	Simplified Chinese sort is based on radical as primary order and number of strokes order as secondary order.
SCHINESE_STROKE_M	Simplified Chinese sort uses number of strokes as primary order and radical as secondary order.
SCHINESE_PINYIN_M	Simplified Chinese Pinyin sorting order.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Sort Name	Description
TCHINESE_RADICAL_M	Traditional Chinese sort based on radical as primary order and number of strokes order as secondary order.
TCHINESE_STROKE_M	Traditional Chinese sort uses number of strokes as primary order and radical as secondary order. It supports supplementary characters.

PL/SQL Connection Attributes

The PL/SQL connection attributes are associated with the shared memory segment, the number of cursors, the compilation, and other settings related to PL/SQL.

The PL/SQL connection attributes are grouped into the following categories:

- PL/SQL First Connection Attributes
- PL/SQL General Connection Attributes

See the List of Connection Attributes for a comprehensive list of the connection attributes in TimesTen, their types, descriptions, and default values.

PL/SQL First Connection Attributes

Use these attributes to determine the virtual address and size of the shared memory segment required by PL/SQL and specify the maximum number of cursors that can be open in a session at one time.

The PL/SQL first connection attributes are described in detail next.

PLSQL_MEMORY_ADDRESS

Use of PL/SQL requires a shared memory segment. This attribute determines the virtual address at which this shared memory segment is loaded into each process that uses the TimesTen direct drivers.

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.

The memory address at which this shared memory segment is loaded 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:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Operating system	Address
Linux x86-64	0000005000000000
AIX	06fffff000000000
Windows	000000005b8c0000
HP-UX	0

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:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems odbc.ini file in TimesTen	PLSQL_MEMORY_ADDRESS	A hexidecimal value that indicates the memory address for PL/SQL process.
Windows ODBC Data Source Administrator	Not applicable	

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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

- 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:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems odbc.ini file in TimesTen	PLSQL_MEMORY_SIZE	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.
Windows ODBC Data Source Administrator	Not applicable	

PLSQL_OPEN_CURSORS

This attribute specifies the maximum number of PL/SQL cursors that can be open in a session at one time.

Use this to prevent a session from opening an excessive number of cursors. Default is 50 PL/SQL cursors.

Updating the value of this attribute takes effect on the next connection, not the current connection. The value also can be set at a database level via the ttDBConfig built-in procedure, providing a default for future connections.
If you decrease the value and the number of open cursors currently exceeds or equals the new setting, no new cursors can be opened until the total number of open cursors is less than the new setting (i.e., some of the currently open cursors have to close).

A value of 0 indicates no PL/SQL cursors can be open. (But if there are cached PL/SQL cursors that contain any PL/SQL code, they could still be executed.)

> **Note**
>
> This attribute has the same functionality as OPEN_CURSORS in Oracle Database.

Required Privilege

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

Copyright © 1996, 2026, Oracle and/or its affiliates.

Setting

Set PLSQL_OPEN_CURSORS as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems odbc.ini file in TimesTen	PLSQL_OPEN_CURSORS	A positive integer from 0 to 65535 representing the number of cursors that can be open in one session at one time. The default value is 50.
Windows ODBC Data Source Administrator	N/A	

PL/SQL General Connection Attributes

Use these attributes to control and specify options for PL/SQL compilation, processing, and the number of session cursors to cache.

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

The PL/SQL general connection attributes are described in detail next.

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 *vi* 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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Setting

Set PLSQL_CCFLAGS as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems odbc.ini file in TimesTen	PLSQL_CCFLAGS	'A string literal with this format: *'v1:c1,v2:c2,...,vn:cn'* Default: null
Windows ODBC Data Source Administrator	Not applicable	

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:

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

Required Privilege

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

Setting

Set PLSQL_CONN_MEM_LIMIT as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems odbc.ini file in TimesTen	PLSQL_CONN_MEM_LIMIT	An integer value in MB. Default value is 100. A setting of 0 means no limit.
Windows ODBC Data Source Administrator	Not applicable	

Note that the amount of space consumed by PL/SQL variables is roughly what you might expect comparable variables to consume in other programming languages. As an example, consider a large array of strings:

```
type chararr is table of varchar2(32767)
             index by binary_integer;
big_array chararr;
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

If 100,000 strings of 100 bytes each are placed into such an array, approximately 12 megabytes of memory is consumed.

Memory consumed by variables in PL/SQL blocks is used while the block executes, then is released. Memory consumed by variables in PL/SQL package specifications or bodies (not within a procedure or function) is used for the lifetime of the package. Memory consumed by variables in a PL/SQL procedure or function, including one defined within a package, is used for the lifetime of the procedure or function. However, in all cases, memory freed by PL/SQL is not returned to the operating system. Instead, it is kept by PL/SQL and reused by future PL/SQL invocations. The memory is freed when the application disconnects from TimesTen.

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:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`PLSQL_OPTIMIZE_LEVEL`	The default value is 2.
Windows ODBC Data Source Administrator	Not applicable	

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;
```

The value of this connection attribute also can be set at a database level by the ttDBConfig built-in procedure.

Copyright © 1996, 2026, Oracle and/or its affiliates.

> **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:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`PLSQL_SESSION_CACHED_CURSORS`	A positive integer from 1 to 65535 representing the number of cursors to cache. The default value is `50`.
Windows ODBC Data Source Administrator	N/A	

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;
```

> **Note**
>
> - 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:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`PLSQL_TIMEOUT`	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`.
Windows ODBC Data Source Administrator	Not applicable	

Also see `TTC_TCP_KEEPALIVE_TIME_MS`.

Cache Connection Attributes

Use cache connection attributes only when caching Oracle Database data.

The cache connection attributes are grouped into three categories:

- Cache First Connection Attributes
- Cache Database Attributes
- Cache General Connection Attributes

See the List of Connection Attributes for a comprehensive list of the connection attributes in TimesTen, their types, descriptions, and default values.

Cache First Connection Attributes

Cache first connection attributes are only used when caching Oracle Database data.

The cache first connection attributes are described in detail next.

CacheAdminWallet

The `CacheAdminWallet` attribute specifies that credentials for the Oracle cache administration user that are registered with the `ttCacheUidPwdSet` built-in procedure are stored in a system-managed Oracle Wallet.

> **Note**
>
> In TimesTen, you can register credentials by using `ttAdmin -cacheUidPwdSet` or calling the `ttCacheUidPwdSet` built-in procedure.

- Value set to 0: TimesTen encrypts the credentials you define and stores them in memory.
- Value set to 1: The default value. TimesTen creates an Oracle Wallet to store the credentials.

The following scenarios apply when setting `CacheAdminWallet`:

- Changing the value from 0 to 1. TimesTen creates an Oracle Wallet on disk to store the user ID and password and clears the password from memory on the first connection on the transition from 0 to 1.

Copyright © 1996, 2026, Oracle and/or its affiliates.

- Changing the value from 1 to 0. TimesTen deletes the Oracle Wallet from the disk. Since the password is not in memory, you must set the password again.

This example sets the credentials by calling the `ttCacheUidPwdSet` built-in procedure.

```
CALL ttCacheUidPwdSet('cacheadmin','orapwd');
```

The following shows an `odbc.ini` file with the `cache1` client DSN setting the `CacheAdminWallet` to 1.

```
[cache1]
DataStore=/users/OracleCache/ttcache
PermSize=64
OracleNetServiceName=oracledb
DatabaseCharacterSet=AL32UTF8
CacheAdminWallet=1
```

Required Privilege

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

Setting

Set `CacheAdminWallet` as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`CacheAdminWallet`	`0` - Encrypts the Oracle cache administration user ID and password and stores them in memory. 1 (default) - Creates an Oracle Wallet to store the Oracle cache administration user ID and password. It clears the password from memory if it was previously set with `CacheAdminWallet=0`.
Windows ODBC Data Source Administrator	Not applicable	

Also see

- `ttCacheUidPwdSet`

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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

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.

> **Note**
>
> - This attribute can also be set through the `ttDBConfig` built-in procedure, which overrides the connection attribute setting. See ttDBConfig.
> - 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:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`CacheAWTMethod`	`0` - Use SQL array execution method. 1 (default) - Use PL/SQL collections and anonymous blocks (PL/SQL execution method).
Windows ODBC Data Source Administrator	Not applicable	

Cache Database Attributes

Cache database attributes are only used when caching Oracle Database data.

The cache database attributes are described in detail next.

CacheAWTParallelism

`CacheAWTParallelism` indicates the number of threads that apply changes to the Oracle database.

This attribute has a relationship to ReplicationParallelism and ReplicationApplyOrdering.

Copyright © 1996, 2026, Oracle and/or its affiliates.

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 Improving AWT Throughput with Parallel Propagation to the Oracle Database in *Oracle TimesTen In-Memory Database Cache Guide*.

Required Privilege

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

Setting

Set `CacheAWTParallelism` as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`CacheAWTParallelism`	*n* - An integer between 1 and 31 that indicates the number of threads that apply changes to the Oracle database. The default is 1.
Windows ODBC Data Source Administrator	Not applicable	

UseCacheConnPool

Enable the cache connection pool with the `UseCacheConnPool` connection attribute.

The cache connection pool can only be initiated from client/server applications and is used only for dynamic loads initiated for dynamic read-only cache groups.

To learn more about the cache connection pool, see Managing a Cache Connection Pool to the Oracle Database for Dynamic Load Requests in *Oracle TimesTen In-Memory Database Cache Guide*.

Required Privilege

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

Setting

Set `UseCacheConnPool` as follows:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`UseCacheConnPool`	`2` - Enabled: For each statement that requires a dynamic load from the Oracle database, the connections to Oracle are released after the load completes and the connection is returned to the cache connection pool. If a dynamic load is performed from a direct connection, the dynamic load proceeds as if the feature is not enabled. `0` (default) - Disabled: For each statement that requires a dynamic load from the Oracle database, the connections to Oracle are not closed by TimesTen after the load completes.
Windows ODBC Data Source Administrator	Not applicable	

Cache General Connection Attributes

Cache general connection attributes are only used when caching Oracle Database data.

The cache general connection attributes are described in detail next.

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:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`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	Not applicable	

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 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 Dynamic Cache Groups in *Oracle TimesTen In-Memory Database Cache 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 *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:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`DynamicLoadErrorMode`	`0` (default) - Statements execute against the cached data with no error. `1` - Statements use dynamic load or fail with an error.
Windows ODBC Data Source Administrator	Not applicable	

Copyright © 1996, 2026, Oracle and/or its affiliates.

OracleNetServiceName

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:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`OracleNetServiceName`	Character string specifying the Oracle Service Name that is to be used as the Oracle ID.
Windows ODBC Data Source Administrator	Not applicable	

OraclePWD

When using TimesTen for caching, the `OraclePWD` attribute is the password the Oracle cache administration user provides to connect to the Oracle database to perform cache operations. These cache operations include, for example, cache group SQL, passthrough, and ttLoadFromOracle.
For more information about cache group SQL operations, see Setting a Passthrough Level in *Oracle TimesTen In-Memory Database Cache Guide*. For information about loading data from Oracle Database and using `ttLoadFromOracle`, see Use TimesTen Built-In Procedures to Recommend a Table and Load SQL Query Results in *Oracle TimesTen In-Memory Database Operations Guide*.

Notes:

- The name for the Oracle cache administration user must be the same as the TimesTen cache administration user.
- To specify the TimesTen cache administration user name, use `UID`.
- To specify the TimesTen cache administration password, use `PWD`.
- The value of the `OraclePWD` attribute can be different from the value specified for the `PWD` attribute.

Required Privilege

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

Setting

You can use the `OraclePWD` attribute to register an Oracle password in the connection string. However, we recommend you to store your passwords in a wallet. See Providing Cache Administration User Credentials When Connecting in *Oracle TimesTen In-Memory Database Cache Guide*.

Copyright © 1996, 2026, Oracle and/or its affiliates.

If you decide to use the OraclePWD attribute to create an Oracle password, suppose you have defined the following odbc.ini file on Linux:

```
[myDSN]
Datastore=/data/myDSN
PermSize=512
TempSize=256
LogBufMB=256
LogFileSize=256
DatabaseCharacterSet=AL32UTF8
ConnectionCharacterSet=AL32UTF8
OracleNetServiceName=ttorcl
```

The example specifies OraclePWD as myoraclepwd for user cacheadmin by connecting to myDSN. The TimesTen cache administration password PWD is pwd. The Oracle cache administration user is not the instance administrator.

```
% ttisql
Copyright (c) 1996, 2022, Oracle. All rights reserved.
Type ? or "help" for help, type "exit" to quit ttIsql.

Command> connect"DSN=myDSN;UID=cacheadmin;PWD=mypwd;OraclePWD=myoraclepwd";
Connection successful:
DSN=myDSN;UID=cacheadmin;DataStore=/=/data/myDSN;DatabaseCharacterSet=AL32UTF8;
ConnectionCharacterSet=AL32UTF8;LogFileSize=256;LogBufMB=256;PermSize=512;
TempSize=256;OracleNetServiceName=ttorcl;(Default setting AutoCommit=1)
```

To connect using Windows, you must provide the OraclePWD in the connect string. The following example provides the cache administration user, the TimesTen cache administration user password and the Oracle cache administration user password.

```
ttIsql -connStr "DSN=database1CS;UID=cacheadmin;PWD=mypwd;OraclePWD=orapwd;"
```

Also see

- PwdWallet
- ttUser
- UID and PWD

PassThrough

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 see Setting a Passthrough Level in the *Oracle TimesTen In-Memory Database Cache 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.

Required Privilege

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

Copyright © 1996, 2026, Oracle and/or its affiliates.

Setting

Set PassThrough as follows.

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems odbc.ini file in TimesTen	PassThrough	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.
Windows ODBC Data Source Administrator	Not applicable	

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

Copyright © 1996, 2026, Oracle and/or its affiliates.

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.
- 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.

For more information, see *Oracle TimesTen In-Memory Database Cache 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:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`RACCallback`	`0` - Do not install TAF and FAN callbacks. 1 (default) - Install the TAF and FAN callbacks.
Windows ODBC Data Source Administrator	Not applicable	

StandbyNetServiceName

This attribute identifies the Service Name for the standby Oracle instance from an Oracle Active Data Guard environment.

To cache Oracle database tables and enable communication with the standby Oracle database, you must specify an Oracle Service Name.

Required Privilege

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

Setting

Set `StandbyNetServiceName` as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`StandbyNetServiceName`	Character string specifying the Oracle Service Name that is to be used as the standby Oracle ID.
Windows ODBC Data Source Administrator	Not applicable	

Client and Server Connection Attributes

Use these attributes when connecting a TimesTen client application to a TimesTen server.

The client and server connection attributes are grouped into two categories:

- Client General Connection Attributes
- Server First Connection Attributes

See the List of Connection Attributes for a comprehensive list of the connection attributes in TimesTen, their types, descriptions, and default values.

Client General Connection Attributes

Used only when you are connecting to a TimesTen server from a TimesTen client application.

In addition to the attributes detailed 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:

Copyright © 1996, 2026, Oracle and/or its affiliates.

- ConnectionCharacterSet
- ConnectionName
- UID and PWD
- PWDCrypt

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*.

The client general connection attributes are described in detail next.

TTC_REDIRECT

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:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`TTC_REDIRECT`	1 (default) - redirect to any available data instance 0 - error if redirection to specified data instance fails
Windows ODBC Data Source Administrator	Not applicable	

Copyright © 1996, 2026, Oracle and/or its affiliates.

CipherSuites

The `CipherSuites` client connection attribute contains one or more cipher suites used to configure Transport Layer Security (TLS) client/server encryption.

TimesTen Release 26.1 supports TLS protocol version 1.3 and 1.2. While TLS 1.3 is more secure, faster, and efficient than TLS 1.2, TimesTen supports TLS 1.2 cipher suites for clients using older versions of TimesTen. A cipher suite is a set of authentication, encryption, and data integrity algorithms used for exchanging messages between network entities. In this case, the network entities are the TimesTen Client and the TimesTen Server.

You can specify one or more cipher suites for the TimesTen client-specific `CipherSuites` connection attribute and for the TimesTen server-specific `CipherSuites` connection attribute. If you specify more than one cipher suite for the connection attribute, ensure to separate each by a comma. TimesTen recommends that you list the cipher suites from strongest to weakest. Ensure that the client and the server have at least one common cipher suite. Cipher suite negotiation works as follows:

- If both client and server list at least one TLS 1.3 cipher suite, TLS 1.3 is negotiated and TLS 1.2 cipher suites are ignored. When TLS 1.3 is negotiated, the client and server select the first TLS 1.3 cipher suite that appears in both `CipherSuites` lists. The order of cipher suites in the client's `CipherSuites` list determines the selection.
- If no mutually configured TLS 1.3 cipher suite exists, TimesTen falls back to TLS 1.2 cipher suites.. When TLS 1.2 is negotiated, the client proposes its ordered list of TLS 1.2 cipher suites. The server selects the first cipher suite from the client's list that is also configured on the server.

If the server and client cannot agree on a common cipher suite, the secure connection fails.

See Secure Network Communication in TimesTen in *Oracle TimesTen In-Memory Database Security Guide* for more details:

Required Privilege

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

Setting

Set `CipherSuites` as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen for encrypted communication between clients and the server.	`CipherSuites`	Specify one or more TLS version 1.3 or 1.2 cipher suites in a comma-separated list from strongest to weakest. See Cipher Suites in TimesTen in *Oracle TimesTen In-Memory Database Security Guide* for the full list of supported cipher suites for TimesTen.
Windows ODBC Data Source Administrator	**CipherSuites** field on the Oracle TimesTen Client DSN Setup dialog.	Specify one or more TLS version 1.3 or 1.2 cipher suites in a comma-separated list from strongest to weakest. See Cipher Suites in TimesTen in *Oracle TimesTen In-Memory Database Security Guide* to learn more about the supported cipher suites and how TimesTen negotiates cipher suite selection.

Copyright © 1996, 2026, Oracle and/or its affiliates.

ChildServer

Identify the child server process with the `ChildServer` connection attribute.

In a client/server environment, TimesTen can have multiple TimesTen child server processes to handle incoming requests from clients. You provide the `ChildServer` connection attribute to identify a specific child server process for certain cache connection pool built-in procedures.

Each child server process is identified by a number assigned with the `ChildServer=`*n* connection attribute, where *n* is a number ranging from 1 to the number of running child server processes. Once connected to the child server process, you can execute either the `ttCacheConnPoolGet('current')` or `ttCacheConnPoolApply` built-in procedures that are meant for a specific child server process.

For example, the following connects to the child server process identified as 1 and applies the saved cache connection pool configuration to this child server process. It does the same process for child server process 2 (given that `ServersPerDSN`=2).

```
Command> connect "DSN=cache1;ChildServer=1;";
Command> call ttCacheConnPoolApply;
Command> disconnect;

Command> connect "DSN=cache1;ChildServer=2;";
Command> call ttCacheConnPoolApply;
Command> disconnect;
```

To learn more about the `ClientServer` connection attribute, see Managing a Cache Connection Pool to the Oracle Database for Dynamic Load Requests in the *Oracle TimesTen In-Memory Database Cache Guide*.

Required Privilege

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

Setting

Set `ChildServer` as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`ChildServer`	`ChildServer=`*n*, where *n* is a number ranging from 1 to the number of running child server processes. The default value is `0`. When you specify the `ChildServer` connection attribute your client process connects using the identified child server process. If the attribute is not specified, then the client process connects using a randomly selected child server process.
Windows ODBC Data Source Administrator	Not applicable	

Copyright © 1996, 2026, Oracle and/or its affiliates.

Encryption

The `Encryption` attribute specifies whether encryption is accepted, rejected, requested, or required for a client/server connection.

You must set this attribute for both the client and the server. Cipher settings must be the same on both the client and server, in most cases.

See Configuration for TLS for Client/Server in *Oracle TimesTen In-Memory Database Security Guide* for more details.

Required Privilege

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

Setting

Set `Encryption` as follows.

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen Classic	`Encryption`	• `accepted`: Enable an encrypted session if required or requested by the client; use an unencrypted session otherwise. (default) • `rejected`: Demand an unencrypted session. (If the server does not support encryption, TimesTen behaves as if this is the setting on the server.) The connection is rejected if the client requires encryption. • `requested`: Request an encrypted session if the client allows it (if the client has any setting other than `rejected`); use an unencrypted session otherwise. • `required`: Demand an encrypted session. Reject the connection if the client rejects encryption.
Windows ODBC Data Source Administrator	**Encryption** field on the Oracle TimesTen Client DSN Setup dialog.	• `accepted`: Enable an encrypted session if required or requested by the client; use an unencrypted session otherwise. (default) • `rejected`: Demand an unencrypted session. (If the server does not support encryption, TimesTen behaves as if this is the setting on the server.) The connection is rejected if the client requires encryption. • `requested`: Request an encrypted session if the client allows it (if the client has any setting other than `rejected`); use an unencrypted session otherwise. • `required`: Demand an encrypted session. Reject the connection if the client rejects encryption.

SSLClientAuthentication

The `SSLClientAuthentication` attribute specifies whether SSL client authentication is required (setting of 1) or not (setting of 0, the default). With client authentication, the server validates an identity presented by the client, and requires an identity (public/private key) in the client wallet.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Regardless of the client authentication setting, server authentication is performed, where the client validates the server.

You must set this attribute for both the client and the server. Regardless of the client authentication setting, server authentication is performed, where the client validates the server.

See Configuration for TLS for Client/Server in *Oracle TimesTen In-Memory Database Security Guide* for more details.

Required Privilege

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

Setting

Set `SSLClientAuthentication` as follows.

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen Classic	`SSLClientAuthentication`	`1` - Client authentication is required `0` - No authentication is required. (the default)
Windows ODBC Data Source Administrator	**SSL Client Authentication** field on the Oracle TimesTen Client DSN Setup dialog.	`1` - Client authentication is required `0` - No authentication is required. (the default)

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.

The default TCP/IP port number is assumed for `TCP_Port` unless you specify a value in the `TTC_Server` connection attribute, in the ODBC connection string, or in the logical server definition.

If the TimesTen server is listening on a non-default port number, you must provide the port number in one of the following ways:

- If using TimesTen Classic, you can specify the port number within the logical server definition, which contains the network address and port number pair. See Defining a Logical Server Name in *Oracle TimesTen In-Memory Database Operations Guide* for more information on defining a logical server.
- You can specify the port number within the `TTC_Server` connection attribute using:

  ```
  TTC_SERVER=server_host_name/server_port;
  ```

- You can specify the port number in the ODBC connection string.

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

  Or:

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


Copyright © 1996, 2026, Oracle and/or its affiliates.

Required Privilege

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

Setting

Set `TCP_Port` as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs	`TCP_Port`	Specify the port number where the server is listening.
Windows ODBC Data Source Administrator and UNIX and Linux systems `ttconnect.ini` file. See Creating and Configuring a Logical Server Name on Linux and UNIX in *Oracle TimesTen In-Memory Database Operations Guide* for more details.	**TCP_Port** on the Oracle TimesTen Logical Server Name Setup dialog.	Specify the port number where the server is listening.

TCP_Port2, TCP_PortN

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

The default TCP/IP port number is assumed for `TCP_Port2` and `TCP_Port`*`N`* unless you specify a value in the appropriate `TTC_Server`*`N`* connection attribute, in the ODBC connection string, or in the logical server definition. See TTC_Server or TTC_Server1 for more details.

Unspecified values for `TCP_Port`*`N`* inherit the value of `TCP_PORT` (or `TCP_PORT1`). For example, if `TTC_Server2` is specified but `TTC_Server_DSN2` and `TCP_Port2` are not, then `TTC_Server_DSN2` is set to the `TTC_Server_DSN` value and `TCP_Port2` is set to the `TCP_Port` value.

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, TCP_PortN` as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs	`TCP_Port2, TCP_Port`*`N`*	Specify the failover port number where the server should listen.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Where to set the attribute	How the attribute is represented	Setting
Windows ODBC Data Source Administrator and UNIX and Linux systems `ttconnect.ini` file. See Creating and Configuring a Logical Server Name on Linux and UNIX in *Oracle TimesTen In-Memory Database Operations Guide* for more details.	Specify **TCP_Port2, TCP_PortN** in a connection string when also specifying a TTC_Server2, TTC_ServerN.	Specify the failover port number where the server should listen.

TTC_ConnectTimeout

The `TTC_ConnectTimeout` attribute specifies the maximum number of seconds the client waits for a `SQLDriverConnect` or `SQLDisconnect` request. It overrides the value of `TTC_Timeout` for those requests. Set the `TTC_ConnectTimeout` when you want connection requests to timeout with a different timeframe than the timeout provided for query requests. For example, you can set a longer timeout for connections if you know that it takes a longer time to connect, but set a shorter timeout for all other queries.

(You can use `TTC_SqlTimeoutMS` to override `TTC_Timeout` for all other requests, at a millisecond level.)

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.

For more details, see Choose SQL and PL/SQL Timeout Values in the *Oracle TimesTen In-Memory Database Operations Guide*.

Also see TTC_Timeout and TTC_SqlTimeoutMS.

Required Privilege

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

Setting

Set `TTC_ConnectTimeout` as follows.

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`TTC_ConnectTimeout`	Seconds to wait for a client connect or disconnect request. Default is 20 seconds.
Windows ODBC Data Source Administrator	Not applicable	

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

Copyright © 1996, 2026, Oracle and/or its affiliates.

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:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`TTC_FailoverPortRange`	Specify a lower value and an upper value for the port numbers in the format *`lowervalue`*–*`uppervalue`*.
Windows ODBC Data Source Administrator	**Failover Port Range** field on the Oracle TimesTen Client DSN Setup dialog.	Specify a lower value and an upper value for the port numbers in the format *`lowervalue`*–*`uppervalue`*.

TTC_NetMsgMaxBytes

The `TTC_NetMsgMaxBytes` attribute specifies the maximum size in bytes of a client result set buffer.

The server stores a maximum number of bytes in the result set *buffer size* returned from a `SELECT` statement. To improve the application's performance, you can increase or decrease the maximum size in bytes of a client result set buffer with the `TTC_NetMsgMaxBytes` connection attribute.

The buffer size can be set in terms of bytes or rows. The lower limit takes precedence. If you set one of them, it is recommended that you set the other also. It is suggested to decide first which maximum attribute you want to use and set it; then, set the other attribute to a value high enough to ensure that this value is not reached first.

The `SELECT` statement will *always* return a complete result set even if filling the buffer takes multiple iterations. The `SELECT` result is not limited by the `TTC_NetMsgMaxBytes` attribute.

You can set the `TTC_NetMsgMaxBytes` attribute in a connection string or DSN to serve as the value for any `SELECT` statement on the connection.

> **Note**
>
> Although the minimum possible setting value for the `TTC_NetMsgMaxBytes` attribute is one byte, using this value would result in a buffer that is too small to contain the smallest row. For this reason, regardless of the value specified, the buffer is always sized to be large enough to contain at least one row.

Copyright © 1996, 2026, Oracle and/or its affiliates.

See Sizing the Client Result Set Buffer in the *Oracle TimesTen In-Memory Database Operations Guide* for more information.

Required Privilege

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

Setting

Set `TTC_NetMsgMaxBytes` as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`TTC_NetMsgMaxBytes`	A positive integer value. The minimum size is 1. Default is 2097152 bytes (2MB).
Windows ODBC Data Source Administrator	**Net Msg Max Bytes** field on the Oracle TimesTen Client DSN Setup dialog.	A positive integer value. The minimum size is 1. Default is 2097152 bytes (2MB).

TTC_NetMsgMaxRows

The `TTC_NetMsgMaxRows` attribute specifies the maximum number of rows stored in the client result set buffer.

The server stores a maximum number of rows in the result set *buffer size* returned from a `SELECT` statement. To improve the application's performance, you can increase or decrease the maximum number of rows of a client result set buffer with the `TTC_NetMsgMaxRows` connection attribute.

Notes

- The `SELECT` statement will *always* return a complete result set even if filling the buffer takes multiple iterations. The `SELECT` result is not limited by the `TTC_NetMsgMaxRows` attribute.
- The buffer size can be set in terms of rows or bytes. The lower limit takes precedence. If you set one of them, it is recommended that you set the other also. It is suggested to decide first which maximum attribute you want to use and set it; then, set the other attribute to a value high enough to ensure that this value is not reached first.

You can set the `TTC_NetMsgMaxRows` attribute in a connection string or DSN to serve as the value for any `SELECT` statement on the connection.

See Sizing the Client Result Set Buffer in *Oracle TimesTen In-Memory Database Operations Guide* for more information.

Required Privilege

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

Setting

Set `TTC_NetMsgMaxRows` as follows:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`TTC_NetMsgMaxRows`	A positive integer value. The minimum is 1 row. Default is 8192 rows.
Windows ODBC Data Source Administrator	**Net Msg Max Rows** field on the Oracle TimesTen Client DSN Setup dialog.	A positive integer value. The minimum is 1 row. Default is 8192 rows.

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).

Required Privilege

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

Setting

Set `TTC_NoReconnectOnFailover` as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`TTC_NoReconnectOnFailover`	`0` = Client reconnects to server after failover (default). `1` = Client does all the failover processing, but does not reconnect after failover.
Windows ODBC Data Source Administrator	**No reconnect on failover** field on the Oracle TimesTen Client DSN Setup dialog.	`0` = Client reconnects to server after failover (default). `1` = Client does all the failover processing, but does not reconnect after failover.

TTC_Random_Selection

Specifies that the TimesTen client, if necessary, 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.

- `1` (default): Initially, the list of failover servers provided by `TTC_ServerN` connection attributes is randomized. After which, the client selects sequentially from the randomized list for the initial connection and then for any client failover request.
- `0`: Client selects the first server specified by the `TTC_ServerN` connection attributes.

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

Copyright © 1996, 2026, Oracle and/or its affiliates.

Required Privilege

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

Setting

Set `TTC_Random_Selection` as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`TTC_Random_Selection`	`0` = Client selects the first server specified by the `TTC_ServerN` attributes. 1 (default)= Initially, the list of failover servers provided by `TTC_ServerN` connection attributes is randomized. After which, the client selects sequentially from the randomized list for the initial connection and then for any client failover request.
Windows ODBC Data Source Administrator	Not applicable	

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. The network address provided can be a domain name server (DNS), host name or IP address.

> **Note**
>
> If you are configuring for client failover, you may define more than one TimesTen server. See TTC_Server2, TTC_ServerN for more information.

If the TimesTen server is listening on a non-default port number, you must provide the port number in one of the following ways:

- You can specify the port number within the `TTC_Server` connection attribute using:

  ```
  TTC_SERVER=server_host_name/server_port;
  ```

- You can specify the port number in the ODBC connection string.

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

  Or:

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

- If using TimesTen, you can specify the port number within the logical server definition in the `ttconnect.ini` file.

Copyright © 1996, 2026, Oracle and/or its affiliates.

See TCP_Port for more details.

You can specify a logical server name for the TTC_Server attribute with TimesTen that contains the network address and port number pair in the ttconnect.ini file. Once the logical server name is defined in the ttconnect.ini file, you can use that name as the value for the TTC_Server attribute in a Client DSN definition. 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. See Creating and Configuring a Logical Server Name on Linux and UNIX in the *Oracle TimesTen In-Memory Database Operations Guide* for more details.

Required Privilege

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

Setting

Set TTC_Server or TTC_Server1 as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems odbc.ini file in TimesTen	TTC_Server, TTC_Server1	Character string specifying the logical server.
Windows ODBC Data Source Administrator	**Server Name or Network Address** field on the Oracle TimesTen Client DSN Setup dialog.	Character string specifying the logical server.

TTC_Server2, TTC_ServerN

For TimesTen, 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.

- When using automatic client failover with an active standby pair replication scheme in TimesTen, you can only define TTC_Server2. After which, the client alternately attempts to connect to TTC_Server and TTC_Server2 until a connection succeeds or the TTC_TIMEOUT attribute expires.
- For other types of automatic client failover, you can specify a list of failover servers with TTC_Server*N* connection attributes where *N* >= 2. TimesTen can iterate through this list of designated failover servers (as necessary) that you configured as TTC_Server2, TTC_Server3, TTC_Server4, and so on. The maximum number of servers that the client can specify is 999.

> **Note**
>
> See TTC_Random_Selection on how TimesTen iterates through the list of designated failover servers.

Unspecified values for TTC_Server*N*, TTC_Server_DSN*N*, and TCP_Port*N* inherit the value of TTC_Server (or equivalently, TTC_Server1), TTC_Server_DSN (or TTC_Server_DSN1), and TCP_PORT (or TCP_PORT1), respectively. For example, if TTC_Server2 is specified but

Copyright © 1996, 2026, Oracle and/or its affiliates.

TTC_Server_DSN2 and TCP_Port2 are not, then TTC_Server_DSN2 is set to the TTC_Server_DSN value and TCP_Port2 is set to the TCP_Port value.

You should configure your failover servers sequentially. If you do skip a number when configuring your failover servers, then TimesTen automatically creates the missing definition and assigns it to the server identified by TTC_Server. In this case, your client could fail over to the same server multiple times.

When using an active standby pair replication scheme for client failover, the TTC_Server or TTC_Server1 and TTC_Server2 connection attributes could potentially have the same setting if it is a virtual IP address. Virtual IP addresses can dynamically move to different hosts; thus, both connection attributes may have the same definition, but could be referencing distinct databases.

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 attributes.

Setting

Set TTC_Server2 or TTC_ServerN as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems odbc.ini file in TimesTen	TTC_Server2, TTC_ServerN	Character string specifying the logical server to be used if an automatic failover occurs.
Windows ODBC Data Source Administrator	**Failover Server Name or Network Address** field on the Oracle TimesTen Client DSN Setup dialog configures the TTC_Server2 connection attribute.	Character string specifying the logical server to be used if an automatic failover occurs.

TTC_Server_DSN

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

More details on this topic can be found in Defining Client DSNs on a TimesTen Client System in the *Oracle TimesTen In-Memory Database Operations Guide*.

Required Privilege

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

Setting

Set TTC_Server_DSN as follows:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`TTC_Server_DSN`	Character string specifying the DSN that resides on the server.
Windows ODBC Data Source Administrator	**Server DSN** field on the Oracle TimesTen Client DSN Setup dialog.	Character string specifying the DSN that resides on the server.

TTC_Server_DSN2, TTC_Server_DSNn

For TimesTen, this attribute specifies the Server DSN on the computer running the TimesTen server. 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` or `TTC_Server_DSN`*n*.

Unspecified values for `TTC_Server_DSN`*n* inherit the value of `TTC_Server_DSN` (or `TTC_Server_DSN1`). For example, if `TTC_Server2` is specified but `TTC_Server_DSN2` and `TCP_Port2` are not, then `TTC_Server_DSN2` is set to the `TTC_Server_DSN` value and `TCP_Port2` is set to the `TCP_Port` value.

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` or `TTC_Server_DSN`*n* as follows:

Where to set the attribute	How the attribute is represented	Setting
C or `Java programs or UNIX and Linux systems` ODBC.INI file	`TTC_Server_DSN2`, `TTC_Server_DSN`*n*	Character string specifying the DSN that resides on the server to be used if an automatic failover occurs.
Windows ODBC Data Source Administrator	**Failover Server DSN** field (only for `TTC_Server_DSN2`) on the Oracle TimesTen Client DSN Setup dialog.	Character string specifying the DSN that resides on the server to be used if an automatic failover occurs.

TTC_SqlTimeoutMS

For client/server, this specifies the integer number of milliseconds the client will wait for a response to a request. It overrides the `TTC_Timeout` connection attribute for any SQL execution request other than `SQLDriverConnect` or `SQLDisconnect`. A value of 0 means there is no timeout. A negative value defers to the `TTC_Timeout` setting (which is in integer seconds).

(The timeout for `SQLDriverConnect` and `SQLDisconnect` is specified by setting the TimesTen `TTC_ConnectTimeout` connection attribute.)

Copyright © 1996, 2026, Oracle and/or its affiliates.

Also see TTC_Timeout and TTC_ConnectTimeout.

Required Privilege

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

Setting

Set TTC_SqlTimeoutMS as follows.

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems ODBC.INI file	TTC_SqlTimeoutMS	Integer milliseconds to wait for response from client.
Windows ODBC Data Source Administrator	N/A	

TTC_TCP_KEEPALIVE_INTVL_MS

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

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.

Required Privilege

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

Setting

Set TTC_TCP_KEEPALIVE_INTVL_MS as follows.

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems odbc.ini file in TimesTen	TTC_TCP_KEEPALIVE_INTVL_MS	A positive integer value. Default is 1000.
Windows ODBC Data Source Administrator	**TCP KeepAlive Interval** field on the Oracle TimesTen Client DSN Setup dialog.	A positive integer value. Default is 1000.

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.

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.

Required Privilege

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

Copyright © 1996, 2026, Oracle and/or its affiliates.

Setting

Set TTC_TCP_KEEPALIVE_PROBES as follows.

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems odbc.ini file in TimesTen	TTC_TCP_KEEPALIVE_PROBES	A positive integer value. Default is 2.
Windows ODBC Data Source Administrator	**TCP KeepAlive Probes** field on the Oracle TimesTen Client DSN Setup dialog.	A positive integer value. Default is 2.

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.

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.

Required Privilege

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

Setting

Set TTC_TCP_KEEPALIVE_TIME_MS as follows.

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems odbc.ini file in TimesTen	A positive integer value. Default is 1000.	
Windows ODBC Data Source Administrator	**TCP KeepAlive Time** field on the Oracle TimesTen Client DSN Setup dialog.	A positive integer value. Default is 1000.

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 OCIStmtExecute() 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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

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` and for all other SQL execution requests by `TTC_SqlTimeoutMS` (at millisecond level). See TTC_ConnectTimeout and TTC_SqlTimeoutMS.

The `SQLQueryTimeout`/`SQLQueryTimeoutMsec` value should be less than the `TransactionTimeout` value; the `TransactionTimeout` value should be less than the `TTC_Timeout` value.

Required Privilege

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

Setting

Set `TTC_Timeout` as follows.

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`TTC_Timeout`	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.
Windows ODBC Data Source Administrator	**Network Timeout** field on the Oracle TimesTen Client DSN Setup dialog.	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.

Wallet

The `Wallet` attribute specifies the fully qualified directory path name, where you placed the certificates that you generated (preferably the same directory path as on the client). There is no default location. If you specify a relative path, it is relative to the *`timesten_home`*`/info` directory.

You must set this attribute for both the client and the server. You must set the same path on both the client and server.

See Configuration for TLS for Client/Server in *Oracle TimesTen In-Memory Database Security Guide* for more details.

Required Privilege

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

Copyright © 1996, 2026, Oracle and/or its affiliates.

Setting

Set `Wallet` as follows.

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`Wallet`	A fully qualified directory path name (no default).
Windows ODBC Data Source Administrator	**Wallet** field on the Oracle TimesTen Client DSN Setup dialog.	A fully qualified directory path name (no default).

Server First Connection Attributes

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.

Server first connection attributes are specified in the server DSN only and are read at first connection. See Defining Server DSNs for TimesTen Server on a Linux or UNIX System in *Oracle TimesTen In-Memory Database Operations Guide*.

> **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.

The server first connection attributes are described in detail next.

CipherSuites

The `CipherSuites` attribute lists the cipher suite or suites that can be used, depending also on the client setting.

TimesTen Release 26.1 supports TLS protocol version 1.3 and 1.2. While TLS 1.3 is more secure, faster, and efficient than TLS 1.2, TimesTen supports TLS 1.2 cipher suites for clients using older versions of TimesTen. A cipher suite is a set of authentication, encryption, and data integrity algorithms used for exchanging messages between network entities. In this case, the network entities are the TimesTen Client and the TimesTen Server.

You can specify one or more cipher suites for the TimesTen client-specific `CipherSuites` connection attribute and for the TimesTen server-specific `CipherSuites` connection attribute. If you specify more than one cipher suite for the connection attribute, ensure to separate each by a comma. TimesTen recommends that you list the cipher suites from strongest to weakest.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Ensure that the client and the server have at least one common cipher suite. Cipher suite negotiation works as follows:

- If both client and server list at least one TLS 1.3 cipher suite, TLS 1.3 is negotiated and TLS 1.2 cipher suites are ignored. When TLS 1.3 is negotiated, the client and server select the first TLS 1.3 cipher suite that appears in both `CipherSuites` lists. The order of cipher suites in the client's `CipherSuites` list determines the selection.
- If no mutually configured TLS 1.3 cipher suite exists, TimesTen falls back to TLS 1.2 cipher suites. When TLS 1.2 is negotiated, the client proposes its ordered list of TLS 1.2 cipher suites. The server selects the first cipher suite from the client's list that is also configured on the server.

If the server and client cannot agree on a common cipher suite, the secure connection fails.

See Secure Network Communication in TimesTen in *Oracle TimesTen In-Memory Database Security Guide* for more details:

Required Privilege

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

Setting

Set `CipherSuites` as follows:

Where to set the attribute	How the attribute is represented	Setting
Client/Server: C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen for encrypted communication between clients and the server.	`CipherSuites`	Specify one or more TLS version 1.3 or 1.2 cipher suites in a comma-separated list from strongest to weakest. See Cipher Suites in TimesTen in *Oracle TimesTen In-Memory Database Security Guide* to learn more about the supported cipher suites and how TimesTen negotiates cipher suite selection.
Replication: C or Java programs or UNIX and Linux systems `timesten.conf` file in TimesTen for encrypted communications between replication agents or between TimesTen utilities and replication agents.	`replication_cipher_suite`	Specify one or more TLS version 1.3 or 1.2 cipher suites in a comma-separated list from strongest to weakest. See Cipher Suites in TimesTen in *Oracle TimesTen In-Memory Database Security Guide* for the full list of supported cipher suites for TimesTen.

Encryption

The `Encryption` attribute specifies whether encryption is accepted, rejected, requested, or required for a client/server connection.

You must set this attribute for both the client and the server. Cipher settings must be the same on both the client and server, in most cases.

Copyright © 1996, 2026, Oracle and/or its affiliates.

See Configuration for TLS for Client/Server in *Oracle TimesTen In-Memory Database Security Guide* for more details.

Required Privilege

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

Setting

Set `Encryption` as follows.

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen Classic	`Encryption`	• `accepted`: Enable an encrypted session if required or requested by the client; use an unencrypted session otherwise. (default) • `rejected`: Demand an unencrypted session. (If the server does not support encryption, TimesTen behaves as if this is the setting on the server.) The connection is rejected if the client requires encryption. • `requested`: Request an encrypted session if the client allows it (if the client has any setting other than `rejected`); use an unencrypted session otherwise. • `required`: Demand an encrypted session. Reject the connection if the client rejects encryption.
Windows ODBC Data Source Administrator	**Encryption** field on the Oracle TimesTen Client DSN Setup dialog.	• `accepted`: Enable an encrypted session if required or requested by the client; use an unencrypted session otherwise. (default) • `rejected`: Demand an unencrypted session. (If the server does not support encryption, TimesTen behaves as if this is the setting on the server.) The connection is rejected if the client requires encryption. • `requested`: Request an encrypted session if the client allows it (if the client has any setting other than `rejected`); use an unencrypted session otherwise. • `required`: Demand an encrypted session. Reject the connection if the client rejects encryption.

SSLClientAuthentication

The `SSLClientAuthentication` attribute specifies whether SSL client authentication is required (setting of 1) or not (setting of 0, the default).

With client authentication, the server validates an identity presented by the client, and requires an identity (public/private key) in the client wallet.

Regardless of the client authentication setting, server authentication is performed, where the client validates the server.

You must set this attribute for both the client and the server. Regardless of the client authentication setting, server authentication is performed, where the client validates the server.

See Configuration for TLS for Client/Server in *Oracle TimesTen In-Memory Database Security Guide* for more details.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Required Privilege

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

Setting

Set `SSLClientAuthentication` as follows.

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen Classic	`SSLClientAuthentication`	`1` - Client authentication is required. `0` - No authentication is required (the default).
Windows ODBC Data Source Administrator	**SSL Client Authentication** field on the Oracle TimesTen Client DSN Setup dialog.	`1` - Client authentication is required. `0` - No authentication is required (the default).

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 Connection Attributes for Data Manager DSNs or Server DSNs 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 System Limits.

Changes to TimesTen server settings do not occur until the TimeTen server is restarted. To restart 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:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`MaxConnsPerServer`	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.
Windows ODBC Data Source Administrator	Not applicable	

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 (`MaxConnsPerServer > 1`). If `ServersPerDSN` is set to 1 then the first MaxConnsPerServer client connections to the server DSN will be assigned to one child server process, the next `MaxConnsPerServer` connections to a second child server process and so on. See Connection Attributes for Data Manager DSNs or Server DSNs 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 `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:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`ServersPerDSN`	A value between 1 and 2047. The default is 1.
Windows ODBC Data Source Administrator	Not applicable	

Copyright © 1996, 2026, Oracle and/or its affiliates.

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 Defining Server DSNs for TimesTen Server on a Linux or UNIX System 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:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`ServerStackSize`	Valid values depend on the platform. The default is 768 KB. If the `sysconf` call is available, the minimum is: `sysconf(_SC_THREAD_STACK_MIN)/ 1024` else the minimum is 0.
Windows ODBC Data Source Administrator	Not applicable	

SSLRenegotiationPeriod

The `SSLRenegotiationPeriod` attribute specifies a period of time, in minutes, after which session renegotiation is performed.

The default setting is 0, meaning do not renegotiate based on a time period.

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

If both `SSLRenegotiationSize` and `SSLRenegotiationPeriod` are set with non-zero values, whichever setting occurs first causes the renegotiation.

See Configuration for TLS for Client/Server in *Oracle TimesTen In-Memory Database Security Guide* for more details.

Copyright © 1996, 2026, Oracle and/or its affiliates.

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 `SSLRenegotiationPeriod` as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen Classic	`SSLRenegotiationPeriod`	An integer representing the number of minutes to wait to for session renegotiation. `0` - indicates that the time period is not used to renegotiate session start (the default)
Windows ODBC Data Source Administrator	Not applicable	

SSLRenegotiationSize

The `SSLRenegotiationSize` attribute specifies a number of megabytes of data transfer in either direction between the client and server, after which session renegotiation is performed.

The default setting is 0, meaning do not renegotiate based on megabytes transferred.

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

If both `SSLRenegotiationSize` and `SSLRenegotiationPeriod` are set with non-zero values, whichever setting occurs first causes the renegotiation.

See Configuration for TLS for Client/Server in *Oracle TimesTen In-Memory Database Security Guide* for more details.

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 `SSLRenegotiationSize` as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen Classic	`SSLRenegotiationSize`	An integer representing the number of megabytes of data transfer after which the session restart is renegotiated. `0` - indicates that the size is not used to renegotiate session start (the default)
Windows ODBC Data Source Administrator	Not applicable	

Copyright © 1996, 2026, Oracle and/or its affiliates.

Wallet

The `Wallet` attribute specifies the fully qualified directory path name, where you placed the certificates that you generated (preferably the same directory path as on the client).

There is no default location. If you specify a relative path, it is relative to the *`timesten_home`*`/info` directory.

You must set this attribute for both the client and the server. You must set the same path on both the client and server.

See Configuration for TLS for Client/Server in *Oracle TimesTen In-Memory Database Security Guide* for more details.

Required Privilege

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

Setting

Set `Wallet` as follows.

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX and Linux systems `odbc.ini` file in TimesTen	`Wallet`	A fully qualified directory path name (no default).
Windows ODBC Data Source Administrator	**Wallet** field on the Oracle TimesTen Client DSN Setup dialog.	A fully qualified directory path name (no default).

Copyright © 1996, 2026, Oracle and/or its affiliates.

3
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:

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

This is the equivalent JDBC call:

```
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 Using 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`.

```
Command> 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 `ttIsql` command line:

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


Copyright © 1996, 2026, Oracle and/or its affiliates.

> **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

There are 133 built-in procedures available in TimesTen. All are listed in the table below.

Built-in Name	Description
ttAgingLRUConfig	Sets the Least Recently Used (LRU) aging attributes on all regular tables that have been defined with an LRU aging policy.
ttAgingTableLRUConfig	Sets the Least Recently Used (LRU) aging attributes on a specified table that has been defined with an LRU aging policy.
ttAgingScheduleNow	Starts the aging process
ttApplicationContext	Sets application-defined context for the next update record to pass application specific data to XLA readers.
ttBackupStatus	Returns information about the current or last backup of the database.
ttBlockInfo	Provides information about perm blocks and the amount of block-level fragmentation in a database.
ttBookmark	Returns information about the TimesTen transaction log.
ttCacheADGStandbyStateGet	Returns the state for the standby Oracle database in an Active Data Guard configuration.
ttCacheADGStandbyStateSet	Sets the state of the standby Oracle database in an Active Data Guard configuration.
ttCacheADGStandbyTimeoutGet	Retrieves the timeout value of the Oracle database in an Active Data Guard configuration.
ttCacheADGStandbyTimeoutSet	Sets the timeout value of the standby Oracle database in an Active Data Guard configuration.
ttCacheAllowFlushAwtSet	Enables you to run a `FLUSH CACHE GROUP` statement on an AWT cache group.
ttCacheAutorefIntervalStatsGet	Returns statistical information about the last 10 autorefresh cycles for a specified autorefresh interval.
ttCacheAutorefresh	Starts an immediate autorefresh on a set of cache groups.
ttCacheAutorefreshLogDefrag	Compacts the trigger log space for a cache autorefresh table.
ttCacheAutorefreshStatsGet	Returns information about the last 10 autorefresh transactions on the specified cache group.
ttCacheAutorefreshSelectLimit	Configures the incremental autorefresh on a specific number of rows.
ttCacheAutorefreshXactLimit	Starts an immediate autorefresh on single table cache groups within a specified autorefresh interval and commits after the specified number of operations.
ttCacheAWTMonitorConfig	Sets AWT cache group monitoring.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Built-in Name	Description
ttCacheAWTThresholdGet	Returns the current transaction log file threshold for databases that include AWT cache groups.
ttCacheAWTThresholdSet	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.
ttCacheCheck	Checks for missing constraints for cached tables on the Oracle database
ttCacheConfig	Configures timeout value and recovery policies for cache groups.
ttCacheConnPoolApply	Applies the cache connection pool settings.
ttCacheConnPoolGet	Retrieves the cache connection pool settings.
ttCacheConnPoolSet	Configures the cache connection pool for dynamic cache groups.
ttCacheDbCgStatus	Returns the automatic refresh status of the database and the specified cache group.
ttCacheDDLTrackingConfig	Configures tracking of DDL statements issued on cached Oracle database tables.
ttCacheLoadSCNGet	Retrieves the load SCN (System Change Number) for a log-based cache group.
ttCacheLogStats	Provides statistics on changes applied by GoldenGate to the specified log-based cache group.
ttCachePolicyGet	Returns the current policy used to determine when the TimesTen cache agent for the connected database should run.
ttCachePolicySet	Sets the policy used to determine when the TimesTen cache agent for the connected database should run.
ttCachePropagateFlagSet	Configures propagation of committed updates to a cache group within the current transaction to the Oracle database.
ttCacheSqlGet	Generates the Oracle SQL statements to install or uninstall Oracle database objects for certain types of cache groups.
ttCacheStart	Starts the TimesTen cache agent.
ttCacheStop	Stops the TimesTen cache agent.
ttCacheUidGet	Returns the cache administration user ID.
ttCacheUidPwdSet	Sets the cache administration user ID and password.
ttCkpt	Performs a non-blocking checkpoint operation.
ttCkptBlocking	Performs a blocking checkpoint operation.
ttCkptConfig	Reconfigures the background checkpointer dynamically or returns the currently active settings of the configuration parameters.
ttCkptConfigGet	Returns the current values of the configuration parameters.
ttCkptHistory	Returns information about the last eight checkpoints.
ttCommitBufferStats	Returns the number of commit buffer overflows and the high watermark for memory used by transaction reclaim records during transaction commit process.
ttCommitBufferStatsReset	Resets transaction commit buffer statistics to 0.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Built-in Name	Description
ttCompact	Compacts both the permanent and temporary data partitions of the database.
ttComputeTabSizes	Refreshes table size statistics stored in TimesTen system tables.
ttConfiguration	Returns the values for most, but not all, connection attributes for the current database connection.
ttContext	Returns the context value of the current connection.
ttDataStoreStatus	Returns the list of processes connected to a database.
ttDBCompactConfig	Sets or returns the value of a TimesTen database system parameter.
ttDBConfig	Sets or returns the value of a TimesTen database system parameter.
ttDBWriteConcurrencyModeGet	Returns information about the write concurrency mode of the database and the status of write concurrency mode operations and transitions.
ttDBWriteConcurrencyModeSet	Controls read optimization during periods of concurrent write operations.
ttDurableCommit	Sets transaction durability.
ttHeapInfo	Returns information about the size and usage of heap memory.
ttHostNameGet	Returns the name of the current local host.
ttHostNameSet	Specifies the name of the default local host
ttIndexAdviceCaptureDrop	Drops existing capture data for either the current connection
ttIndexAdviceCaptureEnd	Ends either an active connection level capture from the current connection or an active database level capture
ttIndexAdviceCaptureInfoGet	Returns information for each active capture.
ttIndexAdviceCaptureOutput	Returns index recommendations from the last recorded capture at the specified level.
ttIndexAdviceCaptureStart	Enables index advice capture.
ttLatchStatsGet	Displays latch statistics.
ttLoadFromOracle	Runs a query on the Oracle database and loads the result into a TimesTen table.
ttLockLevel	Changes the lock level between row-level and database-level locking on the next transaction and for all subsequent transactions for the connection.
ttLockWait	Changes the lock timeout interval of the current connection.
ttLogHolds	Returns information about transaction log holds
ttMonitorHighWaterReset	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.
ttOptClearStats	Clears the statistics for the specified table.
ttOptCmdCacheInvalidate	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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Built-in Name	Description
ttOptEstimateStats	Updates the statistics for the specified table.
ttOptGetColStats	Returns statistics information in text format.
ttOptGetFlag	Returns the optimizer flag settings for the current transaction.
ttOptGetMaxCmdFreeListCnt	Returns the size of the free list of SQL compiled command cache.
ttOptGetOrder	Returns a single-row result set containing the join order for the current transaction.
ttOptSetColIntvlStats	Modifies the statistics for the specified columns with interval information.
ttOptSetColStats	Modifies the statistics for the specified columns.
ttOptSetFlag	Sets flags to alter the generation of execution plans by the TimesTen query optimizer.
ttOptSetMaxCmdFreeListCnt	Sets the maximum count of the free list of SQL compiled commands for regular tables.
ttOptSetMaxPriCmdFreeListCnt	Sets the maximum count of the free list of SQL compiled commands that perform materialized view maintenance.
ttOptSetOrder	Specifies the order in which tables should be joined by the optimizer.
ttOptSetTblStats	Modifies the statistics for the specified table.
ttOptShowJoinOrder	Returns the join order of the last prepared or performed SQL statement in the current transaction.
ttOptStatsExport	Returns the set of statements required to restore the table statistics to the current state.
ttOptUpdateStats	Updates the statistics for the specified table.
ttOptUseIndex	Alters the generation of execution plans by the TimesTen query optimizer.
ttPageLevelTableInfo	Shows the page allocation for each table to determine whether TimesTen is actually reusing empty slots and freeing empty pages or if new pages are allocated to store new rows.
ttPLSQLMemoryStats	Returns result statistics about PL/SQL library cache performance and activity.
ttRamPolicyAutoReloadGet	Returns the RAM autoreload policy used to determine if a database is reloaded into RAM after an invalidation.
ttRamPolicyAutoReloadSet	Determines the RAM autoreload policy if a database is invalidated.
ttRamPolicyGet	Returns the RAM policy used to determine when a database is loaded into memory.
ttRamPolicySet	Defines the policy used to determine when a database is loaded into memory.
ttRedundantIndexCheck	Scans tables to find redundant indexes.
ttRekeyStatus	Returns information about the currently ongoing or last completed rekey operation.
ttRepDeactivate	Changes the state of the active database in an active standby pair from `ACTIVE` to `IDLE`.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Built-in Name	Description
ttReplicationStatus	Returns the status of one or more replication peer databases.
ttRepPolicyGet	Returns the replication restart policy
ttRepPolicySet	Specifies the replication restart policy
ttRepQueryThresholdGet	Returns the number of seconds that was most recently specified as the query threshold for the replication agent.
ttRepQueryThresholdSet	Specifies the number of seconds that a query can be run by the replication agent before TimesTen writes a warning to the daemon log.
ttRepStart	Starts the TimesTen replication agent for the connected database.
ttRepStateGet	Returns the current replication state of a database in an active standby pair.
ttRepStateSave	Saves the state of a remote peer database in an active standby pair to the currently connected database.
ttRepStateSet	Sets the replication state of a database in an active standby pair replication scheme.
ttRepStop	Stops the TimesTen replication agent for the connected database.
ttRepSubscriberStateSet	Changes a replicating subscriber's state with respect to the executing master store.
ttRepSubscriberWait	Causes the caller to wait until all transactions that committed before the call have been transmitted to the subscriber.
ttRepSyncGet	Returns static attributes associated with the caller's use of the replication- based return service.
ttRepSyncSet	Sets static attributes associated with the caller's use of the replication-based return service.
ttRepSyncSubscriberStatus	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.
ttRepTransmitGet	Returns the status of transmission of updates to subscribers for the current transaction.
ttRepTransmitSet	Updates on the connection it is performed in from being replicated to any subscriber.
ttRepXactStatus	Checks the status of a `RETURN RECEIPT` or `RETURN TWOSAFE` replication transaction.
ttRepXactTokenGet	Returns a token for `RETURN RECEIPT` or `RETURN TWOSAFE` replication transactions.
ttSetUserColumnID	Sets the value for the user-specified column ID.
ttSetUserTableID	Sets the value of the user table ID.
ttSize	Estimates the size of a table or view and the size of indexes.
ttSQLCmdCacheInfo	Returns information about all prepared SQL statements in the TimesTen SQL command cache.
ttSQLCmdCacheInfoGet	Returns information about the commands in the TimesTen SQL command cache.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Built-in Name	Description
ttSQLCmdQueryPlan	Returns all detailed runtime query plans for SQL statements in the TimesTen SQL command cache.
ttSQLExecutionTimeHistogram	Returns a histogram of SQL execution times.
ttStatsConfig	Controls statistics collection and parameters for the `ttStats` utility.
ttStatsConfigGet	Returns parameters of the ttStats utility that you can set with the `ttStatsConfig` built-in procedure.
ttTableSchemaFromOraQueryGet	Evaluates a `SELECT` query on a table in an Oracle database and generates a `CREATE TABLE SQL` statement that you can choose to run.
ttVersion	Returns TimesTen release information.
ttWarnOnLowMemory	Specifies that operations run on the current connection should return a warning if they allocate memory and find that memory is low.
ttXactIdGet	Returns transaction ID information for interpreting lock messages.
ttXlaBookmarkCreate	Creates the specified bookmark.
ttXlaBookmarkDelete	Deletes the specified bookmark.
ttXlaSubscribe	Configures persistent XLA tracking of a table.
ttXlaUnsubscribe	Stops persistent XLA tracking of a table.

ttAgingLRUConfig

The `ttAgingLRUConfig` procedure sets the Least Recently Used (LRU) aging attributes in terms of the percentage of MB that rows occupy. You can use it on all regular tables 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.

Related Views

This procedure has no related views.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Syntax

```
ttAgingLRUConfig([LowUsageThreshHold], [HighUsageThreshHold], [AgingCycle])
```

Parameters

`ttAgingLRUConfig` has these optional parameters:

Parameter	Type	Description
lowUsageThreshold	BINARY_FLOAT	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. The default low usage threshold is .8 (80 percent).
highUsageThreshold	BINARY_FLOAT	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. The default high usage threshold is .9 (90 percent).
agingCycle	TT_INTEGER	Sets, displays or resets the number of minutes between aging cycles, specified in minutes. Default is 1 minute. If you use this procedure to change the aging cycle, the cycle is reset based on the time that this procedure is called. For example, if you call this procedure at 12:00 p.m. and specify a cycle of 15 minutes, aging occurs at 12:15, 12:30, 12:45, and so on. If the cycle is set to a value of 0, aging occurs once every second.

Result Set

`ttAgingLRUConfig` returns these results:

Column	Type	Description
lowUsageThreshold	BINARY_FLOAT NOT NULL	The current setting for the low end of percentage of database PermSize, specified in decimals.
highUsageThreshold	BINARY_FLOAT NOT NULL	The current setting for the high end of percentage of database PermSize, specified in decimals.
agingCycle	TT_INTEGER NOT NULL	The current setting for the number of minutes between aging cycles, specified in minutes.

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:

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


Copyright © 1996, 2026, Oracle and/or its affiliates.

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

```
CALL ttAgingLRUConfig();
```

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

```
<.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:

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

> **Note**
>
> 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

ttAgingTableLRUConfig
ttAgingScheduleNow
Usage-Based Aging in the *Oracle TimesTen In-Memory Database Operations Guide*

ttAgingTableLRUConfig

The `ttAgingTableLRUConfig` procedure sets the Least Recently Used (LRU) aging attributes based on the number of rows per table, giving you control over how many rows delete. You can use it on regular and cache tables 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 thresholds are reached based on rows for specified tables and on cache instances when applied to specified cache groups. You can limit the number of rows deleted from a specified table by setting row thresholds for the table.

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. LRU aging automatically deletes rows from child tables. The aging policy is defined on tables when they are created or altered, using the `CREATE TABLE` or `ALTER TABLE` SQL statements.

If both `LowRowsThreshold` and `HighRowsThreshold` are set to zero, table based LRU aging is disabled for this table. LRU aging on this table is switched to be based on the permanent memory in use threshold.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Required Privileges

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

Related Views

This procedure has no related views.

Syntax

```
ttAgingTableLRUConfig(tblOwner, tblName, [LowRowsThreshHold],
[HighRowsThreshHold], [AgingCycle])
```

Parameters

`ttAgingTableLRUConfig` has these optional parameters:

Parameter	Type	Description
`tblOwner`	`TT_CHAR (31)`	TimesTen table or cache group owner.
`tblName`	`TT_CHAR (31)`	Name of an application table. Provide the cache root table for a cache group.
`lowRowsThreshold`	`TT_BIGINT`	Sets, displays or resets the number of the rows at which LRU aging is deactivated. LRU aging stops when the number of rows reaches `lowRowsThreshold` rows in the table.
`highRowsThreshold`	`TT_BIGINT`	Sets, displays or resets the number of the rows at which LRU aging is activated. LRU aging starts when the number of rows reaches `highRowsThreshold` rows in the table.
`agingCycle`	`TT_INTEGER`	Sets, displays or resets the number of minutes between aging cycles, specified in minutes. Default is 1 minute. If you use this procedure to change the aging cycle, the cycle is reset based on the time that this procedure is called. For example, if you call this procedure at 12:00 p.m. and specify a cycle of 15 minutes, aging occurs at 12:15, 12:30, 12:45, and so on. If the cycle is set to a value of `0`, aging occurs once every second.

Result Set

`ttAgingTableLRUConfig` returns these results:

Column	Type	Description
`tblOwner`	`TT_CHAR (31) NOT NULL`	TimesTen table or cache group owner.
`tblName`	`TT_CHAR (31) NOT NULL`	Name of an application table or cache root table.
`lowRowsThreshold`	`TT_BIGINT NOT NULL`	The number of the rows at which LRU aging is deactivated.
`highRowsThreshold`	`TT_BIGINT NOT NULL`	The number of the rows at which LRU aging is activated.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Column	Type	Description
agingCycle	TT_INTEGER NOT NULL	The current setting for the number of minutes between aging cycles, specified in minutes.

Examples

The following example sets the aging threshold for rows in the `user1.table1` table to a low threshold of 10K rows and a high threshold of 100K rows. The aging cycle is set to run at the default of once every minute.

```
Command> Call ttAgingTableLRUConfig('user1', 'table1', 10000, 100000);
< USER1, TABLE1, 10000, 100000, 1 >
1 row found.
```

The following example sets the aging threshold for rows in the `user1.table1` table to a low threshold of 5K rows and a high threshold of 12K rows. The aging cycle is set to run once every two minutes.

```
Command> Call ttAgingTableLRUConfig('user1', 'table1', 5000, 12000, 2);
< USER1, TABLE1, 10000, 100000, 0 >
1 row found.
```

The following example turns off aging by setting the low and high thresholds to zero.

```
Command> Call ttAgingTableLRUConfig('user1', 'table1', 0, 0);
< USER1, TABLE1, 0, 0, 1 >
1 row found.
```

You can retrieve the threshold settings by running the built-in procedure with just the schema owner and table name.

```
Command> Call ttAgingTableLRUConfig('user1', 'table1');
< USER1, TABLE1, 10000, 100000, 1 >
1 row found.
```

> **Note**
>
> 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

ttAgingLRUConfig
ttAgingScheduleNow
Usage-Based Aging in the *Oracle TimesTen In-Memory Database Operations Guide*

ttAgingScheduleNow

The `ttAgingScheduleNow` 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

Copyright © 1996, 2026, Oracle and/or its affiliates.

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 performed manually or in an external scheduler (such as a `cron` job), this procedure starts the aging process at the time the procedure is performed (if there is no aging process in progress) or as soon as the current aging process has been completed. In the case that you want aging to occur *only* when the external scheduler runs 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.

Related Views

This procedure has no related views.

Syntax

```
ttAgingScheduleNow ('tblname')
```

Parameters

`ttAgingScheduleNow` has the parameter:

Parameter	Type	Description
`tblname`	`TT_CHAR (61)`	The name of the table on which to start the aging process. If *`tblName`* is omitted, the aging process is started on all tables defined with any aging policy. Using a synonym to specify a table name is not supported.

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 ();
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

In this example, the user sampleuser1 (not the instance administrator) creates the table agingex with time-based aging policy and the aging state set 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 SAMPLEUSER1.AGINGEX:
    Columns:
     *COL1   TT_INTEGER NOT NULL
      TS     TIMESTAMP (6) NOT NULL
    PRIMARY KEY (COL1) RANGE INDEX
    Aging: Timestamp based uses column TS lifetime 1 minute cycle 30 minutes off

  1 table found.
  (primary key columns are indicated with *)
```

The user inserts two rows with data. Then the user calls the ttAgingScheduleNow procedure and the rows are deleted from the table. After ttAgingScheduleNow is called, the aging state remains OFF.

```
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, 2022-11-08 23:57:00.000000>
< 2, 2022-11-08 23:57:18.000000>
2 rows found.

Command> CALL ttAgingScheduleNow ('agingex');
Command> SELECT * FROM agingex;
0 rows found.
```

See Also

ttAgingLRUConfig

ttApplicationContext

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.

Related Views

This procedure has no related views.

Syntax

```
ttApplicationContext (cmd)
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

Parameters

`ttApplicationContext` has the parameter:

Parameter	Type	Description
cmd	`VARBINARY(16384) NOT NULL`	Context information to be passed to the XLA readers.

Result Set

`ttApplicationContext` returns no results.

Examples

```
CALL ttApplicationContext (0x123);
```

See Also

XLA Reference in *Oracle TimesTen In-Memory Database C Developer's Guide*

ttBackupStatus

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.

Related Views

This procedure has this related view:

SYS.V$BACKUP_STATUS

Syntax

```
ttBackupStatus ()
```

Parameters

`ttBackupStatus` has no parameters.

Result Set

`ttBackupStatus` returns the results:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Column	Type	Description
status	TT_INTEGER NOT NULL	An INTEGER code representing the current progress of a backup or the completion status of the last backup. Values are: 0 - No backup has been taken on the database since the last first-connect. 1 - A backup is currently in progress. 2 - The last backup completed successfully. 3 - The last backup failed. In this case the error column contains the error code for the failure.
destination	TT_INTEGER	The type of backup taken. The value is NULL when no backup has been taken on the database. Value is one of: 0 - Backup is/was being written to a file. 1 - Backup is/was being written to a stream. 2 - Backup is/was taken on behalf of replication duplicate.
backupType	TT_INTEGER	Backup type, either full or incremental. The value is NULL when no backup has been taken on the database. Value is one of: 0 - Incremental backup. 1 - Full backup.
startTime	TT_TIMESTAMP	Time when the backup was started. The value is NULL when no backup has been taken on the database.
endTime	TT_TIMESTAMP	Time when the backup completed. If NULL and *startTime* is non-NULL, a backup is currently in progress.
backupLFN	TT_INTEGER	The transaction log file number of the backup point. The value is NULL when no backup has been taken on the database.
backupLFO	TT_BIGINT	The transaction log file offset of the backup point. The value is NULL when no backup has been taken on the database.
error	TT_INTEGER	If a backup fails, this column indicates the reason for the failure. The value is one of the TimesTen error numbers. The value is NULL when no backup has been taken on the database.
processId	TT_INTEGER	The ID of the process or daemon performing the backup (if known).

Examples

```
CALL ttBackupStatus ();
< 2, 2, 1, 2021-09-13 13:10:32.587557,
2005-08-12 13:10:33.193269, 1, 1531840, 0, 6968 >
1 row found.
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

> **Note**
>
> The procedure does not return information about previous backups other than the current or last one. Also, the information returned is not persistent across database startup or shutdown.

ttBlockInfo

This procedure provides information about perm blocks and the amount of block-level fragmentation in a database.

Required Privilege

This procedure requires no privilege.

Related Views

This procedure has this related view.

SYS.V$BLOCK_INFO

Syntax

```
ttBlockInfo()
```

Parameters

`ttBlockInfo` has no parameters.

Result Set

`ttBlockInfo` returns the result set:

Column	Type	Description
TotalBlocks	TT_BIGINT NOT NULL	Total number of blocks in the database.
FreeBlocks	TT_BIGINT NOT NULL	Total number of free blocks in the database.
FreeBytes	TT_BIGINT NOT NULL	Total size of the free blocks.
LargestFree	TT_BIGINT NOT NULL	Size of the largest free block.

Examples

```
CALL ttBlockInfo();
< 1537, 16, 236036720, 235991352 >
1 row found.
```

ttBookmark

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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

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.

Related Views

This procedure has this related view.

SYS.V$BOOKMARK

Syntax

```
ttBookmark()
```

Parameters

`ttBookmark` has no parameters.

Result Set

`ttBookmark` returns the result set:

Column	Type	Description
writeLFN	TT_INTEGER	Last written transaction log file.
writeLFO	TT_BIGINT	Last written offset in transaction log file.
forceLFN	TT_INTEGER	Last transaction log file forced to disk.
forceLFO	TT_BIGINT	Offset of last transaction log file forced to disk.
holdLFN	TT_INTEGER	Replication bookmark transaction log file.
holdLFO	TT_BIGINT	Replication bookmark log offset.

Examples

```
CALL ttBookmark();
<379, 60193048,  379, 60192768, -1, -1>
1 row found.
```

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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Related Views

This procedure has no related views.

Syntax

```
ttCacheADGStandbyStateGet()
```

Parameters

`ttCacheADGStandbyStateGet` has no parameters.

Result Set

`ttCacheADGStandbyStateGet` returns the state of the standby Oracle database.

Column	Type	Description
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

ttCacheADGStandbyStateSet
ttCacheADGStandbyTimeoutGet
ttCacheADGStandbyTimeoutSet

ttCacheADGStandbyStateSet

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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

  - 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.

Related Views

This procedure has no related views.

Syntax

```
ttCacheADGStandbyStateSet('[OK | FAILED]')
```

Parameters

ttCacheADGStandbyStateSet has the parameters:

Parameter	Type	Description
ADGStandbyState	TT_VARCHAR(20) NOT NULL	OK: Tells the cache agent that the standby Oracle database in an Active Data Guard configuration is active. This is the default. FAILED: 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.

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');
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

```
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.

Related Views

This procedure has no related views.

Syntax

```
ttCacheADGStandbyTimeoutGet()
```

Parameters

`ttCacheADGStandbyTimeoutGet` has no parameters.

Result Set

`ttCacheADGStandbyTimeoutGet` returns the *timeout*:

Column	Type	Description
ADGStandbyTimeout	TT_INTEGER	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.

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

Copyright © 1996, 2026, Oracle and/or its affiliates.

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 running the `ttCacheADGStandbyStateSet` built-in procedure and set the state to `OK`.

Required Privilege

This procedure requires no privileges.

Related Views

This procedure has no related views.

Syntax

`ttCacheADGStandbyTimeoutSet(`*`timeout`*`)`

Parameters

`ttCacheADGStandbyTimeoutSet` has the parameters:

Parameter	Type	Description
`ADGStandbyTimeout`	`TT_INTEGER NOT NULL`	A timeout specified in seconds on how long to wait for the standby Oracle database to respond before using only the primary Oracle database to autorefresh the cache groups. Default is 0, which indicates that no timeout is used and the state of the standby Oracle database does not change from `OK` to `FAILED` as a result of this timeout.

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.
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

See Also

ttCacheADGStandbyStateSet
ttCacheADGStandbyTimeoutGet
ttCacheADGStandbyStateGet

ttCacheAllowFlushAwtSet

The `ttCacheAllowFlushAwtSet` built-in procedure enables you to run a `FLUSH CACHE GROUP` statement against an AWT cache group.

This procedure 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 In-Memory Database Replication Guide*.

Set auto commit to off before running the `ttCacheAllowFlushAwtSet` built-in procedure when setting the *`enableFlush`* parameter to 1; otherwise, this parameter automatically resets to 0 directly after running the built-in procedure. Then, perform a commit after you run the `FLUSH CACHE GROUP` statement and run the `ttCacheAllowFlushAwtSet` built-in procedure to reset the *`enableFlush`* parameter back to 0.

> **Note**
>
> This procedure does not support log-based cache groups.

Required Privilege

This procedure requires no privileges.

Related Views

This procedure has no related views.

Syntax

```
ttCacheAllowFlushAwtSet (enableFlush)
```

Parameters

`ttCacheAllowFlushAwtSet` has the parameters:

Parameter	Type	Description
`allow`	`TT_INTEGER NOT NULL`	`0` - The user is prevented from running a `FLUSH CACHE GROUP` statement against an AWT cache group, which is the intended restriction. `1` - The user is allowed to run a `FLUSH CACHE GROUP` statement against an AWT cache group, which should only be done for recovery, as described in When There Is Unsynchronized Data in the Cache Groups section in the *Oracle TimesTen In-Memory Database Replication Guide*.

Result Set

`ttCacheAllowFlushAwtSet` returns no results.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Examples

The following example shows how to run the `ttCacheAllowFlushAwtSet` built-in procedure to first allow and then disallow a `FLUSH CACHE GROUP` statement to be run 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 In-Memory Database Replication Guide*.

ttCacheAutorefIntervalStatsGet

The `ttCacheAutorefIntervalStatsGet` built-in procedures returns statistical information about the last 10 autorefresh cycles for a particular autorefresh interval.

> **Note**
>
> This procedure does not support log-based cache groups. No interval is to be specified.

Required Privilege

This procedure requires no privileges.

Related Views

This procedure has no related views.

Syntax

`ttCacheAutorefIntervalStatsGet (`*`autoRefInterval`*`, `*`isStatic`*`)`

Parameters

`ttCacheAutorefIntervalStatsGet` has the parameters:

Parameter	Type	Description
`autoRefInterval`	`TT_BIGINT NOT NULL`	The *`autorefreshInterval`* 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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Parameter	Type	Description
isStatic	TT_INTEGER	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.

Result Set

ttCacheAutorefIntervalStatsGet returns statistical information about the last 10 autorefresh cycles for a particular autorefresh interval:

Column	Type	Description
autorefInterval	TT_BIGINT	Autorefresh interval in milliseconds.
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 in process.
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 performed since the autorefresh thread was started.
totalNumCommits	TT_BIGINT	Number of commits since the autorefresh thread was started.
totalNumRollbacks	TT_BIGINT	Number of s since the autorefresh thread started
totalNumSnapshotOld	TT_BIGINT	Number of "Snapshot too old" errors received since the autorefresh thread started

Examples

The following example shows how to run 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, 0, 1, 0, 0, <NULL>,
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

```
132121, 132121, 13, 21, 0, 0, 0, 0 >
< 7000, 1, 40, 2013-04-25 15:16:53.000000, 0, 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, 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, 0, 1, 0, 0, <NULL>,
132121, 132121, 10, 21, 0, 0, 0, 0 >
< 7000, 1, 37, 2013-04-25 15:16:32.000000, 0, 6305, 6305, 1, 0, 131072,
<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>, 125816, 125816, 8, 21, 0, 0, 0, 0 >
< 7000, 1, 35, 2013-04-25 15:16:17.000000, 0, 18176, 18176, 1, 0, 131072,
<NULL>, 110200, 110200, 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 >
< 7000, 1, 33, 2013-04-25 15:16:03.000000, 0, 15360, 15360, 1, 0, 131072,
<NULL>, 77688, 77688, 5, 21, 0, 0, 0, 0 >
< 7000, 1, 32, 2013-04-25 15:15:56.000000, 0, 11520, 11520, 1, 0, 131072,
<NULL>, 62328, 62328, 4, 21, 0, 0, 0, 0 >

10 rows found.
```

This procedure is available only for cache operations.

See Also

- ttCacheAutorefreshSelectLimit
- ttCacheAutorefreshXactLimit
- Running Large Transactions with Incremental Autorefresh Read-Only Cache Groups and Configuring a Select Limit for Incremental Autorefresh for Read-Only Cache Groups in *Oracle TimesTen In-Memory Database Cache Guide*.

ttCacheAutorefresh

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 usually 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.

> **Note**
>
> This procedure does not support log-based cache groups.

Required Privilege

This procedure requires the `CACHE_MANAGER` or `ADMIN` privilege.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Related Views

This procedure has no related views.

Syntax

```
ttCacheAutorefresh ('cgOwner', 'cgName', synchronous)
```

Parameters

`ttCacheAutorefresh` has the parameters:

Parameter	Type	Description
cgOwner	`VARCHAR2 (30)`	Name of the cache group owner.
cgName	`VARCHAR2 (30) NOT NULL`	Name of the cache group.
synchronous	`TT_INTEGER`	Species whether data is updated on synchronously or asynchronously. `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 cache operations.

ttCacheAutorefreshLogDefrag

The `ttCacheAutorefreshLogDefrag` built-in procedure compact the trigger log space for a cache autorefresh table.

> **Note**
>
> This procedure does not support log-based cache groups.

For usage details, see Defragmenting Change Log Tables in the Tablespace in *Oracle TimesTen In-Memory Database Cache Guide*

Copyright © 1996, 2026, Oracle and/or its affiliates.

Required Privilege

This procedure requires the CACHE_MANAGER privilege.

Related Views

This procedure has no related views.

Syntax

```
ttCacheAutorefreshLogDefrag ('action')
```

Parameters

ttCacheAutorefreshLogDefrag has the parameters:

Parameter	Type	Description
action	VARCHAR (50) NOT NULL	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');
```

This procedure is available only for cache operations.

See Also

ttCacheConfig
ttCacheAutorefreshStatsGet

ttCacheAutorefreshStatsGet

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

Copyright © 1996, 2026, Oracle and/or its affiliates.

- The cache group is dropped and recreated

> **Note**
>
> This procedure does not support log-based cache groups.

Required Privilege

This procedure requires no privilege.

Related Views

This procedure has no related views.

Syntax

```
ttCacheAutorefreshStatsGet ('cgOwner', 'cgname')
```

Parameters

`ttCacheAutorefreshStatsGet` has the parameters:

Parameter	Type	Description
CACHE_GROUP_OWNER	`VARCHAR2 (30)`	Name of the cache group owner.
CACHE_GROUP_NAME	`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.

`ttCacheAutorefreshStatsGet` returns the results:

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.	Y
cacheAgentUpTime	`TT_BIGINT`	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.	Y
autorefNumber	`TT_BIGINT`	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.	Y

Copyright © 1996, 2026, Oracle and/or its affiliates.

Column	Type	Description	Returned for full autorefresh
autorefDuration	TT_BIGINT	The number of milliseconds spent in this autorefresh transaction.	Y
autorefNumRows	TT_BIGINT	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.	N
numOracleBytes	TT_BIGINT	The number of bytes transferred from the Oracle database in this autorefresh transaction.	N
autorefNumRootTblRows	TT_BIGINT	The number of root table rows autorefreshed in this autorefresh transaction.	Y
autorefQueryExecDuration	TT_BIGINT	The duration in milliseconds that it takes for the autorefresh query to execute on the Oracle database.	N
autorefQueryFetchDuration	TT_BIGINT	The duration in milliseconds that it takes for the autorefresh query to fetch rows from the Oracle database.	N
autorefTtApplyDuration	TT_BIGINT	The duration in milliseconds that it takes for TimesTen to apply the autorefresh.	N
totalNumRows	TT_BIGINT	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.	N
totalNumOracleBytes	TT_BIGINT	The total number of bytes transferred from the Oracle database since the cache agent started.	N
totalNumRootTblRows	TT_BIGINT	The total number of root table rows autorefreshed since the cache agent started.	Y
totalDuration	TT_BIGINT	The total autorefresh duration in milliseconds since the cache agent started.	Y

status	VARCHAR2 (128)	A string description of the status of the current autorefresh. Supported values for this field are:
		`Complete`
		`inProgress`
		`Failed`
			Y

numlogrows	TT_BIGINT	Number of rows fetched from the Oracle database in this autorefresh.	Y

Copyright © 1996, 2026, Oracle and/or its affiliates.

Column	Type	Description	Returned for full autorefresh
totalnumlogrows	TT_BIGINT	The cumulative number of rows fetched from the Oracle database in this autorefresh.	Y
autorefLogFragmentationPct	TT_BIGINT	A low-water mark for table usage by percentage. If less than the specified percent of the table is used, the table is compacted.	Y
autorefLogFragmentationTs	TT_TIMESTAMP	The timestamp when the last utilization/ fragmentation ratio was calculated	Y
autorefLogDefragGcnt	TT_BIGINT	The number of times the table has been compacted.	Y

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.

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, 659456, 94208,560, 320, 13410, 1591192,
10341748, 1591192, 129430, Complete, 2, 68 >
< 1164260, 2011-07-23 15:42:32.000000, 770180, 40,
12080, 99328, 695296, 99328,450, 290, 11340, 1496984,
9682292, 1496984, 115130, Complete, 2, 66 >
< 1164260, 2011-07-23 15:42:12.000000, 750130, 39,
10380, 86016, 598368, 86016,430, 230, 9720, 1397656,
8986996, 1397656, 103050, 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,
7687860, 1199000, 79140, Complete, 2, 60 >
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

```
< 1164260, 2011-07-23 15:41:22.000000, 700120, 36,
2120, 10240, 50330, 10240, 50, 200, 1870, 1142680,
7361050, 1142680, 69770, Complete, 2, 58 >
< 1164260, 2011-07-23 15:41:12.000000, 690110, 35,
0, 0, 0, 0, 0, 0, 0, 1132440, 7310720, 1132440,
67650, Complete, 2, 56 >
10 rows found.
```

This procedure is available only for cache operations.

ttCacheAutorefreshSelectLimit

Enables the user to configure a select limit of rows to join the Oracle database base table.

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.

> **Note**
>
> This procedure does not support log-based cache groups.

Required Privilege

This procedure requires the `ADMIN` or `CACHE_MANAGER` privileges.

Related Views

This procedure has no related views.

Syntax

```
ttCacheAutorefreshSelectLimit ( autorefreshInterval, value )
```

Parameters

`ttCacheAutorefreshSelectLimit` has the parameters:

Parameter	Type	Description
param	`VARCHAR2 (50) NOT NULL`	The *`autorefreshInterval`* designates the cache group (the one with this autorefresh interval value) on which to apply the *`value`*. 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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Parameter	Type	Description
value	VARCHAR2 (200)	The *value* 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 *value* can be one of the following: • 'ON': Select at most 1000 rows at a time from the autorefresh change log table to apply for every autorefresh cycle. • *number*: 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. • 'OFF': Disables the select limit. The incremental autorefresh selects all rows from the change log table during the autorefresh cycle. • NULL: If the *value* provided is NULL or not specified, the current setting is returned.

Result Set

ttCacheAutorefreshSelectLimit returns the select limit value that has been set for a particular autorefresh interval:

Column	Type	Description
param	VARCHAR2(50)	The *autorefreshInterval* that designates the cache group (the one with this autorefresh interval value).
value	VARCHAR2(200)	The current *value* that shows the number of rows that is selected from the autorefresh change log file to apply to the cached 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 cache operations.

Copyright © 1996, 2026, Oracle and/or its affiliates.

- 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 for Incremental Autorefresh for Read-Only Cache Groups in *Oracle TimesTen In-Memory Database Cache Guide*.

ttCacheAutorefreshXactLimit

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.

> **Note**
>
> This procedure does not support log-based cache groups.

Required Privilege

This procedure requires the `CACHE_MANAGER` or `ADMIN` privilege.

Related Views

This procedure has no related views.

Syntax

```
ttCacheAutorefreshXactLimit ('IntervalValue', 'Value')
```

Parameters

`ttCacheAutorefreshXactLimit` has the parameters:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Parameter	Type	Description
param	VARCHAR2 (50) NOT NULL	Indicates the interval at which the autorefresh cache groups are defined to occur in units of milliseconds. *IntervalValue* is an integer value in milliseconds that was specified when the autorefresh cache group was created on how often autorefresh is scheduled.
value	VARCHAR2 (200)	The *Value* can be one of the following: • 'ON' - Enables autorefresh to commit after every 256 operations. • 'OFF' - Disables the transaction limit for autorefresh cache groups and sets autorefresh back to using a single transaction. • 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. • NULL - When the value is NULL, 0 or not specified, the current setting is returned.

Result Set

ttCacheAutorefreshXactLimit returns the results:

Column	Type	Description
param	VARCHAR2 (50)	The interval at which the autorefresh cache groups are defined to occur in units of milliseconds.
value	VARCHAR2 (200)	The *Value* can be one of the following: • 'ON' - Enables autorefresh to commit after every 256 operations. • 'OFF' - Disables the transaction limit for autorefresh cache groups and sets autorefresh back to using a single transaction. • *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. • NULL - When the value is NULL or not specified, the current setting is returned.

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.

```
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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

```
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:

```
call ttCacheAutorefreshXactLimit('10000', '2000');
```

Notes

- This procedure is available only for cache operations. 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
Running Large Transactions with Incremental Autorefresh Read-Only Cache Groups in *Oracle TimesTen In-Memory Database Cache Guide*.

ttCacheAWTMonitorConfig

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 properly, 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.

> **Note**
>
> This procedure does not support log-based cache groups.

Required Privilege

This procedure requires the `CACHE_MANAGER` privilege.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Related Views

This procedure has no related views.

Syntax

```
ttCacheAWTMonitorConfig (['state'], [samplingRate])
```

Parameters

`ttCacheAWTMonitorConfig` has the optional parameters:

Parameter	Type	Description
state	TT_CHAR(10)	Enables and disables AWT monitoring. Its value can be ON or OFF. Default is OFF
samplingRate	TT_INTEGER	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.

Result Set

`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.

Column	Type	Description
state	TTVARCHAR (10) NOT NULL	Current state of AWT monitoring. The value can be ON or OFF.
AWTSamplingFactor	TT_INTEGER NOT NULL	Positive integer that specifies the frequency with which the AWT workflow is sampled.

Examples

Retrieve the current state and sampling factor when monitoring is disabled.

```
Command> CALL ttCacheAWTMonitorConfig;
< OFF, 0 >
1 row found.
```

Enable monitoring and set the sampling frequency to 16.

```
Command> CALL ttCacheAWTMonitorConfig ('ON', 16);
< ON, 16 >
1 row found.
```

Disable monitoring.

Copyright © 1996, 2026, Oracle and/or its affiliates.

```
Command> CALL ttCacheAWTMonitorConfig; ('OFF')
< OFF, 0 >
1 row found.
```

See Also

ttRepAdmin

ttCacheAWTThresholdGet

This procedure returns the current transaction log file threshold for databases that include AWT cache groups.

> **Note**
>
> This procedure does not support log-based cache groups.

Required Privilege

This procedure requires no privilege.

Related Views

This procedure has no related views.

Syntax

```
ttCacheAWTThresholdGet()
```

Parameters

`ttCacheAWTThresholdGet` has no parameters.

Result Set

`ttCacheAWTThresholdGet` returns the result:

Column	Type	Description
threshold	`TT_INTEGER NOT NULL`	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.

Examples

```
CALL ttCacheAWTThresholdGet();
```

This procedure is available only for cache operations.

See Also

ttCacheAWTThresholdSet

Copyright © 1996, 2026, Oracle and/or its affiliates.

ttCacheAWTThresholdSet

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

> **Note**
>
> This procedure does not support log-based cache groups.

Required Privilege

This procedure requires the `CACHE_MANAGER` privilege.

Related Views

This procedure has no related views.

Syntax

```
ttCacheAWTThresholdSet(threshold)
```

Parameters

`ttCacheAWTThresholdSet` has the parameter:

Parameter	Type	Description
threshold	`TT_INTEGER`	Specifies the number of transaction log files for all AWT cache groups associated with the database. If the threshold is `NULL`, the log failure threshold is set to zero.

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 only for cache operations.
- The user is responsible to recover when the threshold is exceeded.

See Also

ttCacheAWTThresholdGet

Copyright © 1996, 2026, Oracle and/or its affiliates.

ttCacheCheck

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.

See Improving AWT Throughput with Parallel Propagation to the Oracle Database in *Oracle TimesTen In-Memory Database Cache Guide*.

> **Note**
>
> This procedure does not support log-based cache groups.

Required Privilege

This procedure requires the `CACHE_MANAGER` privilege.

Related Views

This procedure has no related views.

Syntax

```
ttCacheCheck('operation', cgOwner, cgName)
```

Parameters

`ttCacheCheck` has these parameters:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Parameter	Type	Description
operation	TT_VARCHAR(30)	Specifies the constraint to be checked. Supported values are: • ForeignKey - Checks foreign key constraints • Unique - Checks unique constraints • Awt - Checks both foreign key and unique constraints • NULL - Checks both foreign key and unique constraints
cgOwner	TT_VARCHAR(30)	Specifies the owner of the cached Oracle database table. If NULL, checks all asynchronous writethrough cache groups owned by the connection user. If both *cgOwner* and *cgName* are NULL, checks all asynchronous cache groups.
cgName	TT_VARCHAR(30)	Specifies the name of the cached Oracle database table. If NULL, but the *cgOwner* is specified checks all asynchronous writethrough cache groups owned by *cgOwner*. If both *cgOwner* and *cgName* are NULL, checks all asynchronous cache groups.

Result Set

ttCacheCheck returns the result set:

Column	Type	Value
cgOwner	TT_VARCHAR(30) NOT NULL	The owner of the cache group.
cgName	TT_VARCHAR(30) NOT NULL	The name of the cache group.
tblOwner	TT_VARCHAR(30)	The owner of the table.
tblName	TT_VARCHAR(30)	The name of the table.
objectType	TT_VARCHAR(15)	The type of Oracle object: unique index, constraint or foreign key.
objectOwner	TT_VARCHAR(30)	The owner of the Oracle object.
objectName	TT_VARCHAR(30)	The object name.
msgType	TT_SMALLINT NOT NULL	The type of message: 0 = Informational 1 = Warning -1 = Error
msg	TT_VARCHAR(100000) NOT NULL	Message describing the issue.
objectDesc	VARCHAR2(200000)	A description of the object. If the object is AWT checking, the description is the SQL statement that describes the object.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Examples

The following example determines if there are any missing constraints for the cache group `update_orders` that is owned by `cacheadmin`. 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, 'cacheadmin, 'update_orders');

< CACHEADMIN, UPDATE_ORDERS, CACHEADMIN, ORDERTAB, Foreign Key, CACHEADMIN,
CUST_FK, 1, Transactions updating this table will be serialized to Oracle
because: The missing foreign key connects two AWT cache groups.,
table CACHEADMIN.ORDERTAB constraint CACHEADMIN.CUST_FK foreign key(CUSTID)
references CACHEADMIN.ACTIVE_CUSTOMER(CUSTID) >
1 row found.
```

This procedure is available only for cache operations.

See Also

- ttCacheDbCgStatus
- ttCachePolicyGet
- ttCachePolicySet
- ttCacheStart
- ttCacheStop
- ttCacheUidGet
- ttCacheUidPwdSet
- ttAdmin

ttCacheConfig

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.

Related Views

This procedure has no related views.

Syntax

`ttCacheConfig(`*`Param`*`, `*`tblOwner`*`, `*`tblName`*`, `*`Value`*`)`

Parameters

`ttCacheConfig` has these parameters:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Parameter	Type	Description
Param	VARCHAR2(50) NOT NULL	Specifies the parameter to be set by *Value*: • 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. • AgentTimeout - Number of seconds before a database is declared terminated if the cache agent cannot connect to the Oracle database server. • AutoRefreshLogFragmentationWarningPCT - The percent of table usage that must occur before warning the user to compact the table. By default, value is 40. • AutorefreshLogMonitorInterval - Specifies the time interval (in seconds) for how often to perform the percentage calculation of the cache administration user's tablespace usage. Default value is 300 seconds. • AutoRefreshLogDeFragmentAction - Compaction mode for the specified tables. • AutoRefreshLogTblSpaceUsagePCT - Specifies the cache administration user's tablespace usage warning threshold as a percentage. • CacheCommitDurable - Specifies how the autorefresh changes to the metadata is durably committed. • DeadDbRecovery - Specifies the type of autorefresh recovery when the cache agent restarts. • DisableFullAutorefresh - Disables or enables full autorefresh behavior. Default is 0, meaning that full autorefresh is enabled. The value of 1 indicates that full autorefresh is disabled. • TblSpaceFullRecovery - Specifies the action that TimesTen takes when the cached Oracle database table is updated and the cache administration user's tablespace is full. **Note** It does not support log-based cache groups.
tblOwner	VARCHAR2(30)	Specifies the owner of the cached Oracle database table. This parameter is required if *Param* is set TblspaceFullRecovery. Do not specify *tblOwner* for other values of *Param*. A synonym cannot be used to specify a table name.
tblName	VARCHAR2(30)	Specifies the name of the cached Oracle database table. This parameter is required if *Param* is set TblspaceFullRecovery. Do not specify *tblOwner* for other values of *Param*. Using a synonym to specify a table name is not supported.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Parameter	Type	Description
Value	VARCHAR2(200)	Specifies the value to be set for *Param*. • When *Param* is AgentFailoverTimeout, 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. • 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. The default is 0, which means that the database is never declared terminated. • When *Param* is AutoRefreshLogTblSpaceUsagePCT, the value can be 0 to 100. The default is 0, which means no warning is returned regardless of the tablespace usage. • When *Param* is AutoRefreshLogFragmentationWarningPCT, the value of the fourth parameter must be an integer between 1 and 100, representing a percentage of the table. The default is 40. • When *Param* is AutorefreshLogMonitorInterval, it specifies the interval in seconds when to calculate the percentage of usage of the cache administration user's tablespace. The default is every 300 seconds. • When *Param* is AutoRefreshLogDeFragmentAction, the value can be Manual, CompactOnly or CompactandReclaim If Manual is specified no action is taken. The user can run ttCacheAutorefreshLogDefrag built-in procedure to defragment the logs. If CompactOnly is specified trigger log space is compacted. If CompactandReclaim is specified both the trigger log space and the transaction log buffer (reclaim space) are compacted. The default is Manual. • When *Param* is CommitCacheDurable, the value can be 0 or 1. When value is set to 1 (the default), then an autorefresh operation applies all changes to the cache group tables, durably commits the metadata (and all pending transaction data in the transaction log buffer), and then initiates garbage collection for the autorefresh tracking tables stored on the Oracle database. When value is set to 0, then an autorefresh operation applies all changes to the cache group tables and non-durably commits the metadata. Then, starts a new thread to durably commit the metadata (and all pending transaction data in the transaction log buffer). Garbage collection is initiated for the autorefresh tracking tables stored on the Oracle database after the durable commit of the metadata completes. This results in a slight performance cost as garbage collection is delayed until after the durable commit completes. • When *Param* is DeadDbRecovery, the value can be Normal, Manual or None. Normal specifies a full automatic refresh. Manual specifies that REFRESH CACHE GROUP statement must be issued. None specifies that cache groups must be dropped and then re-created after the cache agent starts. The default is Normal. See Impact of Failed Autorefresh Operations on TimesTen Databases in the *Oracle TimesTen In-Memory Database Cache Guide* for details. • When *Param* is TblSpaceFullRecovery, the value can be Reload or None. Reload specifies that rows are deleted from the change log table and a full automatic refresh is performed. None specifies that an Oracle database error is returned when

Copyright © 1996, 2026, Oracle and/or its affiliates.

Parameter	Type	Description
		the cached Oracle database table is updated. The default is None. Or Specifies the value to be set by *AwtErrorXmlOutput*: • ASCII - A text file that contains the AWT error report. (Default) • XML - An XML file that contains the AWT error report and the associated DTD file.

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.

Column	Type	Value
Param	VARCHAR2(50)	Parameter name: AgentTimeout AgentFailoverTimeout AutoRefreshLogTblSpaceUsagePCT AutoRefreshLogFragmentationWarningPCT AutorefreshLogMonitorInterval AutoRefreshLogDeFragmentAction DeadDbRecovery DisableFullAutorefresh TblSpaceFullRecovery
tblOwner	VARCHAR2(30)	Owner of the cached Oracle database table.
tblName	VARCHAR2(30)	Name of the cached Oracle database table. Using a synonym to specify a table name is not supported.
Value	VARCHAR2(200)	Specifies the value set for *Param*. • 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. • When *Param* is AutoRefreshLogTblSpaceUsagePCT, the value can be 0 to 100. • When *Param* is AutoRefreshLogFragmentationWarningPCT, the value can be 0 to 100. • When *Param* is AutorefreshLogMonitorInterval, the value can be an integer. • When *Param* is AutoRefreshLogDeFragmentAction, the value can be Manual, CompactOnly or CompactandReclaim. • When *Param* is DeadDbRecovery, the value can be Normal or Manual. • When *Param* is TblSpaceFullRecovery, the value can be Reload or None.

Examples

To set the cache agent timeout to 600 seconds (10 minutes), enter:

Copyright © 1996, 2026, Oracle and/or its affiliates.

```
CALL ttCacheConfig('AgentTimeout',,,'600');
```

To determine the current cache agent timeout setting, enter:

```
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:

```
CALL ttCacheconfig('DeadDbRecovery',,,'Manual');
```

Configure TimesTen 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`. See Impact of Failed Autorefresh Operations on TimesTen Databases in the *Oracle TimesTen In-Memory Database Cache Guide* for details.

```
CALL ttCacheConfig('TblSpaceFullRecovery','terry','customer','None');
```

To determine the current setting for `TblSpaceFullRecovery` on the `terry.customer` cached Oracle database table, enter:

```
Command> 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:

```
Command> CALL ttCacheConfig('AutoRefreshLogTblSpaceUsagePCT',,,'85');
```

TimesTen Classic calculates the percentage of fragmentation for the change log tables as a ratio of used space to the total size of the space. If this ratio falls below a defined threshold, TimesTen alerts you of the necessity for defragmentation of the change log tables by logging a message. By default, this threshold is set to 40%. For example, to set the fragmentation threshold to 50%, perform:

```
Command> CALL ttCacheConfig('AutoRefreshLogFragmentationWarningPCT',,,'50');
< AutoRefreshLogFragmentationWarningPCT, <NULL>, <NULL>, 50 >
1 row found.
```

To set the time interval to 3600 seconds for when to calculate the fragmentation percentage of the change log tables, perform:

```
Command> CALL ttCacheConfig('AutorefreshLogMonitorInterval',,,'3600');
< AutorefreshLogMonitorInterval, <NULL>, <NULL>, 3600 >
1 row found.
```

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.
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

Notes

- This procedure is available only for cache operations.
- 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
Managing a Caching Environment, Reporting Oracle Database Permanent Errors for AWT Cache Groups, and Setting Up Cache in an Oracle RAC Environment (regarding Agent Failover) in *Oracle TimesTen In-Memory Database Cache Guide*.

ttCacheConnPoolApply

This procedure enables you to dynamically resize the cache connection pool parameters on each child server process, after which the cache connection pool parameters are associated with the child server process.

Use the `ChildServer` connection attribute to identify each child server process, where `ChildServer=`*n* and *n* is a number ranging from 1 to the number of running child server processes. Once connected to the child server process, you can run the `ttCacheConnPoolApply` built-in procedure that is meant for a specific child server process.

For more details, see Managing a Cache Connection Pool to the Oracle Database for Dynamic Load Requests in *Oracle TimesTen In-Memory Database Cache Guide*.

Required Privilege

This procedure requires TimesTen cache administration manager or `Admin` privileges to run.

Related Views

This procedure has no related views.

Syntax

```
ttCacheConnPoolApply()
```

Parameters

There are no parameters for this procedure.

Result Set

`ttCacheConnPoolApply` returns no results..

Copyright © 1996, 2026, Oracle and/or its affiliates.

Examples

The following connects to the child server process identified as 1 and applies the saved cache connection pool configuration to this child server process. It does the same process for child server process 2 (given that `ServersPerDSN=2`).

```
Command> connect "DSN=cache1;ChildServer=1;";
Command> call ttCacheConnPoolApply ();
Command> disconnect;

Command> connect "DSN=cache1;ChildServer=2;";
Command> call ttCacheConnPoolApply ();
Command> disconnect;
```

Notes

- This procedure is available only for cache operations.
- You can only run the `ttCacheConnPoolApply` built-in procedure from a multithreaded client/server connection.
- If the cache connection pool fails, you can recreate the pool by executing the `ttCacheConnPoolApply` built-in procedure from any child server process.

See Also

ttCacheConnPoolGet
ttCacheConnPoolSet

ttCacheConnPoolGet

This procedure retrieves the current values of the cache connection pool parameters.

The `ChildServer` connection attribute identifies each child server process, where `ChildServer=`*n* and *n* is a number ranging from 1 to the number of running child server processes. Once connected to the child server process, you can run the `ttCacheConnPoolGet('current')` built-in procedure that is meant for a specific child server process.

For more details, see Managing a Cache Connection Pool to the Oracle Database for Dynamic Load Requests in *Oracle TimesTen In-Memory Database Cache Guide*.

Required Privilege

This procedure requires TimesTen cache administration manager or `Admin` privileges to run.

Related Views

This procedure has no related views.

Syntax

```
ttCacheConnPoolGet(state)
```

Parameters

The `ttCacheConnPoolGet` has the parameter `state`. This parameter can be either:

- `saved`: Returns the cache connection parameters that are saved on the Oracle database. When querying the saved configuration, you can run the `ttCacheConnPoolGet` built-in

Copyright © 1996, 2026, Oracle and/or its affiliates.

procedure from a direct connection, a single-threaded client/server connection or a multithreaded client/server connection.

- `current`: Returns the cache connection parameters that have defined the cache connection pool for this current child server process. When querying the current configuration of a specific child server process, you can run the `ttCacheConnPoolGet` built-in procedure only from a multithreaded client/server connection.

Result Set

`ttCacheConnPoolGet` returns the following values:

Column	Type	Description
`MinSize_CURRENT`	`TT_INTEGER`	The set value for the minimum number of open connections for the cache connection pool.
`MaxSize_CURRENT_`	`TT_INTEGER`	The set value for the maximum number of open connections for the cache connection pool.
`IncrSize_CURRENT`	`TT_INTEGER`	The set value for the increment by which the number of open connections increases when needed.
`Timeout_CURRENT`	`TT_INTEGER`	A timeout (in seconds) for the connections in the cache connection pool. Connections that idle for more than this timeout are terminated to maintain an optimum number of open connections and returned to the cache connection pool.
`ConnNoWait_CURRENT`	`TT_INTEGER`	A directive given whether to wait for an available connection from the cache connection pool when no connection is immediately available. Valid values are: • Disabled (`0`): Any dynamic load operations wait for an available connection in the cache connection pool before proceeding. • Enabled (1): Any dynamic load operations fail with an error if there is no available connection in the cache connection pool.
`OpenCount_CURRENT`	`TT_INTEGER`	The current number of open connections in the cache connection pool. If you query the saved parameters, a -1 is displayed for this value.
`BusyCount_CURRENT`	`TT_INTEGER`	The current number of busy connections in the cache connection pool. If you query the saved parameters, a value of -1 is displayed for this value.
`LastError_CURRENT`	`TT_INTEGER`	Displays the number of the last Oracle Database error returned (if applicable) when attempting to retrieve a connection from the cache connection pool. For example, a value of 1034 would indicate that "`ORA-0134: ORACLE not available`" error was the last Oracle Database error returned. When requesting `current` and no Oracle Database error is returned, a 0 is returned. If you query the saved parameters, a -1 is displayed for this value.

Examples

Query the values for the cache connection pool that are saved on the Oracle database.

```
Command> call ttCacheConnPoolGet('saved');
< 1, 10, 1, 10, 0, -1, -1, -1>
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

This procedure is available only for cache operations.

See Also

ttCacheConnPoolApply
ttCacheConnPoolSet

ttCacheConnPoolSet

This procedure sizes the cache connection pool to avoid contention for connections.

The `ttCacheConnPoolSet` procedure also saves the values of these parameters on the Oracle database, which are then used as the default values when restarting the TimesTen server.

If you are dynamically changing the sizing, you can apply the changes to each TimesTen server by executing the `ttCacheConnPoolApply` built-in procedure.

For more details, see Managing a Cache Connection Pool to the Oracle Database for Dynamic Load Requests in *Oracle TimesTen In-Memory Database Cache Guide*.

Required Privilege

This procedure requires TimesTen cache administration manager or `Admin` privileges to run.

Related Views

This procedure has no related views.

Syntax

```
ttCacheConnPoolSet(MinSize, MaxSize, IncrSize, Timeout, ConnNoWait)
```

Parameters

All parameters must be specified for the `ttCacheConnPoolSet` built-in procedure. You can run the `ttCacheConnPoolSet` built-in procedure from a direct connection, a single-threaded client/server connection or a multithreaded client/server connection.

Parameter	Type	Description
`MinSize_IN`	`TT_INTEGER`	Specifies the minimum number of open connections for the cache connection pool. The default is 10 connections. The minimum is 1; the maximum is 2000 connections.
`MaxSize_IN`	`TT_INTEGER`	Specifies the maximum number of open connections for the cache connection pool. The default is 32 connections. The minimum is 1; the maximum is 2000 connections.
`IncrSize_IN`	`TT_INTEGER`	Specifies the increment by which the number of open connections increases when needed. The default increment is 1. The minimum is 1; the maximum is 2000 connections.
`Timeout_IN`	`TT_INTEGER`	Specifies a timeout (in seconds) for the connections in the cache connection pool. Connections that idle for more than this timeout are terminated to maintain an optimum number of open connections and returned to the cache connection pool. The default is 100 seconds. If set to 0, then the connections never time out. The minimum is 0; the maximum is 300.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Parameter	Type	Description
ConnNoWait_IN	TT_INTEGER	Specifies whether to wait for an available connection from the cache connection pool when no connection is immediately available. Valid values are: • Disabled (0 - the default): Any dynamic load operations wait for an available connection in the cache connection pool before proceeding. • Enabled (1): Any dynamic load operations fail with an error if there is no available connection in the cache connection pool.

Examples

The following initiates the minimum and maximum number of pooled connections to be between 10 and 32 connections and the increment is 1. The maximum idle time by the client is set to 10 seconds. And all dynamic load operations will wait for an available connection from the cache connection pool.

```
Command> call ttCacheConnPoolSet(10, 32, 1, 10, 0);
```

This procedure is available only for cache operations.

See Also

ttCacheConnPoolGet
ttCacheConnPoolApply

ttCacheDbCgStatus

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.

> **Note**
>
> This procedure does not support log-based cache groups.

Required Privilege

This procedure requires no privilege.

Related Views

This procedure has no related views.

Syntax

```
ttCacheDbCgStatus([cgOwner], [cgName])
```

Parameters

ttCacheDbCgStatus has these optional parameters:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Parameter	Type	Description
cgOwner	VARCHAR2(30)	Specifies the user name of the cache group owner.
cgName	VARCHAR2(30)	Specifies the cache group name.

Result Set

ttCacheDbCgStatus returns the result:

Column	Type	Value
dbStatus	VARCHAR2(20)	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.
cgStatus	VARCHAR2(20)	Specifies the autorefresh status of the specified cache group. The status is one of: ok - The specified cache group is synchronized with the Oracle database. 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 and the specified cache group may be out of sync with the Oracle database server. recovering - The specified cache group is being resynchronized with the Oracle database server.

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.
```

This procedure is available only for cache operations.

See Also

ttCacheConfig

Copyright © 1996, 2026, Oracle and/or its affiliates.

ttCachePolicyGet
ttCachePolicySet
ttCacheStart
ttCacheStop
ttCacheUidGet
ttCacheUidPwdSet
ttAdmin

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.

See Tracking DDL Statements Issued on Cached Oracle Database Tables in *Oracle TimesTen In-Memory Database Cache Guide*.

Required Privilege

This procedure requires the `CACHE_MANAGER` privilege.

Related Views

This procedure has no related views.

Syntax

```
ttCacheDDLTrackingConfig('trackingStatus')
```

Parameters

`ttCacheDDLTrackingConfig` has the parameter:

Parameter	Type	Description
`trackingStatus`	`TT_VARCHAR(10)`	Specifies whether DDL statements issued on cached Oracle database tables are tracked. Valid values are: `enable` - Enables tracking. `disable` (default) - Disables tracking.

Result Set

`ttCacheDDLTrackingConfig` returns no results.

Examples

```
Command> CALL ttCacheDDLTrackingConfig('enable');
```

This procedure is available only for cache operations.

Copyright © 1996, 2026, Oracle and/or its affiliates.

ttCacheLoadSCNGet

This procedure returns the load SCN a specific SCN value that corresponds to the last time a cache group was loaded from the Oracle database into the TimesTen cache. This is applicable only for log-based cache groups.

Required Privilege

This procedure requires the `TTPRIV_CACHEMGR` privilege.

Related Views

This procedure has this related view.

SYS.V$CACHE_GROUP_LOAD_STATUS

Syntax

```
ttCacheLoadSCNGet('cache_group_owner', 'cache_group_name')
```

Parameters

`ttCacheLoadSCNGet` has these parameters:

Parameter	Type	Description
cache_group_owner	`TT_VARCHAR2(30)`	Cache group owner.
cache_group_name	`TT_VARCHAR2(30) NOT NULL`	The name of the cache group.

Result Set

`ttCacheLoadSCNGet` returns the result:

Column	Type	Value
load_scn	`VARCHAR2(21)`	The SCN (System Change Number) at which the cache group was last loaded from the Oracle database.

Examples

```
Command> CALL ttCacheLoadSCNGet('HR', 'EMP_CACHE');
Load_SCN: 3943852
```

Returns the SCN when the `EMP_CACHE` cache group owned by `HR` was last loaded.

You can also use the `cachegroups` command to find the SCN value. See Task 3: Create TimesTen Users and Tables in the *Oracle TimesTen In-Memory Database Cache Guide*.

Copyright © 1996, 2026, Oracle and/or its affiliates.

ttCacheLogStats

This procedure returns statistics on the changes made by GoldenGate to the specified log-based cache group. If no cache group is specified, all cache groups are displayed.

Required Privilege

This procedure requires the `TTPRIV_NONE` privilege.

Related Views

This procedure has this related view.

SYS.V$CACHE_GROUP_LOG_STATS

Syntax

```
ttCacheLogStats('cache_group_owner', 'cache_group_name')
```

Parameters

`ttCacheLogStats` has these parameters:

Parameter	Type	Description
cache_group_owner	`TT_VARCHAR2(30)`	Cache group owner.
cache_group_name	`TT_VARCHAR2(30) NOT NULL`	The name of the cache group.

Result Set

`ttCacheLogStats` returns the results:

Column	Type	Value
cache_group_owner	`VARCHAR2(30)`	Cache group owner.
cache_group_name	`VARCHAR2(30)`	Name of the cache group.
numInserts	`TT_BIGINT`	Number of inserts that GoldenGate successfully applied into the cache group's parent table.
numUpdates	`TT_BIGINT`	Number of updates that GoldenGate successfully applied into the cache group's parent table.
numDeletes	`TT_BIGINT`	Number of deletes that GoldenGate successfully applied into the cache group's parent table.
numNoops	`TT_BIGINT`	Number of DML operations that were older than the row being affected in the parent table and were therefore ignored (noop operations).
numLoadWaits	`TT_BIGINT`	Number of DML operations that were newer than ongoing dynamic loads and waited for such ongoing dynamics loads to complete.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Examples

```
Command> CALL ttCacheLogStats('HR', 'EMP_CACHE');
<HR, EMP_CACHE, 1523, 874, 212, 18, 3>
1 row found.
```

For all cache groups:

```
Command> CALL ttCacheLogStats();
  CACHE_GROUP_OWNER: HR
  CACHE_GROUP_NAME: EMP_CACHE
  NUMINSERTS: 1523
  NUMUPDATES: 874
  NUMDELETES: 212
  NUMNOOPS: 18
  NUMLOADWAITS: 3

  CACHE_GROUP_OWNER: CACHEADMIN
  CACHE_GROUP_NAME: CGT1
  NUMINSERTS: 0
  NUMUPDATES: 2
  NUMDELETES: 2
  NUMNOOPS: 2
  NUMLOADWAITS: 0

2 rows found.
```

ttCachePolicyGet

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.

Related Views

This procedure has no related views.

Syntax

```
ttCachePolicyGet()
```

Parameters

`ttCachePolicyGet` has no parameters.

Result Set

`ttCachePolicyGet` returns the result:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Column	Type	Value
cachePolicy	TT_VARCHAR(10)	Specifies the policy used to determine when the 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 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.

Examples

To get the current policy for the TimesTen agent, use:

```
CALL ttCachePolicyGet ();
```

This procedure is available only for cache operations.

See Also

- ttCacheConfig
- ttCacheDbCgStatus
- ttCachePolicySet
- ttCacheStart
- ttCacheStop
- ttCacheUidGet
- ttCacheUidPwdSet
- ttAdmin

ttCachePolicySet

This 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.

Related Views

This procedure has no related views.

Syntax

```
ttCachePolicySet('cachePolicy')
```

Parameters

ttCachePolicySet has these parameters:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Parameter	Type	Description
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 cache operations.
- 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
ttCacheDbCgStatus
ttCachePolicyGet
ttCacheStart
ttCacheStop
ttCacheUidGet
ttCacheUidPwdSet
ttAdmin

ttCachePropagateFlagSet

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-

Copyright © 1996, 2026, Oracle and/or its affiliates.

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.

> **Note**
>
> This procedure does not support log-based cache groups.

Required Privilege

This procedure requires the `CACHE_MANAGER` privilege.

Related Views

This procedure has no related views.

Syntax

`ttCachePropagateFlagSet(`*`CommitsOn`*`)`

Parameters

`ttCachePropagateFlagSet` has the parameter:

Parameter	Type	Description
`CommitsOn`	`TT_INTEGER NOT NULL`	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.

Result Set

`ttCachePropagateFlagSet` returns no results.

Notes

- This procedure is available only for cache operations.
- If the value of `ttCachePropagateFlagSet` is reenabled several times during a single transaction, the transaction is only partially propagated to the Oracle database.
- `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 detail table for `oratt.writetab`. Then, propagation is reenabled by calling the `ttCachePropagateFlagSet` built-in procedure and setting the flag to one.

Copyright © 1996, 2026, Oracle and/or its affiliates.

```
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

This procedure generates the Oracle SQL statements to install or uninstall Oracle database objects.

This procedure generates statements 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.

> **Note**
>
> This procedure does not support log-based cache groups.

Required Privilege

This procedure requires the `CACHE_MANAGER` privilege.

Related Views

This procedure has no related views.

Syntax

```
ttCacheSqlGet('feature_name', 'cache_group_name', install_flag)
```

Parameters

`ttCacheSqlGet` has these parameters:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Parameter	Type	Description
feature_name	TT_VARCHAR (100)	Can be specified as INCREMENTAL_AUTOREFRESH or ASYNCHRONOUS_WRITETHROUGH.
cache_group_name	TT_VARCHAR (100)	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.
install_flag	TT_INTEGER NOT NULL	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.

Result Set

ttCacheSqlGet returns the result set:

Column	Type	Description
retval	TT_VARCHAR (4096) NOT NULL	The Oracle SQL statement to uninstall or install autorefresh or asynchronous writethrough Oracle database objects.
continueFlag	TT_SMALLINT NOT NULL	nonzero only if the Oracle SQL statement in the *retval* result column exceeds 4096 bytes and must be continued into the next result row.

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);
```

Notes

- This procedure is available only for cache operations.
- 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 cachesqlget 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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

ttCacheStart

This procedure starts the TimesTen cache agent for the connected database.

Required Privilege

This procedure requires the `CACHE_MANAGER` privilege.

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 cache agent, use:

```
CALL ttCacheStart ();
```

Notes

- This procedure is available only for cache operations.
- 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 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 cache agent with this built-in procedure, the agent is not started because TimesTen Data Manager looks for the database in the `\srv` 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
ttCacheDbCgStatus
ttCachePolicyGet
ttCachePolicySet
ttCacheStop
ttCacheUidPwdSet
ttCacheUidGet
ttAdmin

Copyright © 1996, 2026, Oracle and/or its affiliates.

ttCacheStop

This procedure stops the TimesTen cache agent for the connected database.

Required Privilege

This procedure requires the `CACHE_MANAGER` privilege.

Related Views

This procedure has no related views.

Syntax

`ttCacheStop(`*`timeout`*`)`

Parameters

`ttCacheStop` has the parameter:

Parameter	Type	Description
`timeout`	`TT_INTEGER`	Specifies that the TimesTen daemon should stop the cache agent if it does not stop within *`timeout`* seconds. If set to `0`, the daemon waits forever for the cache agent. The default value is `100`.

Result Set

`ttCacheStop` returns no results.

Examples

To stop the cache agent, use:

```
CALL ttCacheStop();
```

Notes

- This procedure is available only for cache operations.
- 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

Copyright © 1996, 2026, Oracle and/or its affiliates.

ttCacheUidGet

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.

Related Views

This procedure has no related views.

Syntax

```
ttCacheUidGet(UID)
```

Parameters

`ttCacheUidGet` has no parameters.

Result Set

`ttCacheUidGet` returns the results:

Column	Type	Description
UID	`TT_VARCHAR (30)`	The current cache administration user ID, used for autorefresh and asynchronous writethrough cache groups.

Examples

```
CALL ttCacheUidGet();
```

> **Note**
>
> This procedure is available only for cache operations.

See Also

ttCacheUidPwdSet
ttAdmin

ttCacheUidPwdSet

This built-in procedure lets you set the Oracle cache administration user and password within a system-managed wallet in TimesTen to connect to the Oracle database.

You only need to register the Oracle cache administration user ID and password once for each new database. You can change the Oracle cache administration password at any time.

Copyright © 1996, 2026, Oracle and/or its affiliates.

> **Note**
>
> If you defined `CacheAdminWallet=1` as a first connection attribute (in the DSN or on the initial database connection), then the credentials registered with the `ttCacheUidPwdSet` are saved in an Oracle Wallet (this is the recommended method). If you defined `CacheAdminWallet=0` as a first connection attribute, the credentials are stored in memory. See `CacheAdminWallet`.

Required Privilege

This procedure requires the `CACHE_MANAGER` privilege.

Related Views

This procedure has no related views.

Syntax

```
ttCacheUidPwdSet('UID', 'PWD')
```

Parameters

`ttCacheUidPwdSet` has these parameters:

Parameter	Type	Description
UID	`TT_VARCHAR (30)`	The Oracle cache administration user ID that is used for cache operations. It is usually represented in examples as `cacheadmin`. Calling `ttCacheUidPwdSet`() without a *UID* or with a null value for the *UID* erases the stored UID.
PWD	`TT_VARCHAR (30)`	The password for the Oracle cache administration user. It is usually referred as `OraclePWD` and represented in examples as `orapwd`.

Result Set

`ttCacheUidPwdSet` returns no results.

Examples

The first example shows setting the Oracle cache administration user and password using `ttCacheUidPwdSet` built-in procedure. Then, the example calls `ttCacheUidGet` to show the Oracle cache administration user that is set in TimesTen. The next part of the example shows that if you do not provide any values for the Oracle cache administration in the `ttCacheUidPwdSet` built-in procedure, then the Oracle cache administration user is erased.

```
Command> CALL ttCacheUidPwdSet ('cacheadmin','orapwd');
Command> CALL ttCacheUidGet;
< CACHEADMIN >
1 row found.
Command> CALL ttCacheUidPwdSet();
Command> CALL ttCacheUidGet;
< <NULL> >
1 row found.
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

Notes

- This procedure cannot be called from an unencrypted client/server connection.
- This procedure is available only for cache operations
- 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 *Oracle TimesTen In-Memory Database Cache 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
ttAdmin

ttCkpt

The `ttCkpt` built-in procedure enables you to manually perform fuzzy or nonblocking checkpoints.

Fuzzy checkpoints, or non-blocking checkpoints, allow transactions to run against the database while the checkpoint is in progress. A nonblocking checkpoint does not require any locks on the database. See Checkpoint Operations in *Oracle TimesTen In-Memory Database Operations Guide* to learn more about checkpoints.

By default, TimesTen automatically performs background checkpoints at regular intervals. Usually, you can set the automatic checkpoint mechanism with the CkptFrequency and CkptLogVolume connection attributes, but you can also manually initiate a checkpoint operation by calling the `ttCkpt` procedure. Also, TimesTen recommends that you perform regular backups using the ttBackup utility to minimize the risk of potential data loss.

A backup always contains a checkpoint file; therefore, backing up and checkpointing operations can conflict. Regardless of whether the checkpoint is a background checkpoint or an application-requested checkpoint, the following scenarios are possible:

- 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 checkpoint is currently running and you attempt a checkpoint, the following error returns:

  ```
  TT625: Checkpoint cannot proceed because the following conflicting operation
  is in progress: Checkpoint started by connection id id (pid pid; name 'db-
  name')
  ```

- If a backup is currently running and you attempt a checkpoint, the following error returns:

  ```
  TT625: Checkpoint cannot proceed because the following conflicting operation
  is in progress: Backup started by connection id id (pid pid; name 'db-name')
  ```

When a database crashes and the checkpoints on disk are nonblocking checkpoints, TimesTen uses both the checkpoint files and the log to recover.

Required Privilege

This procedure requires the `ADMIN` privilege.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Related Views

This procedure has no related views.

Syntax

```
ttCkpt()
```

Parameters

`ttCkpt` has these parameters.

Parameter	Type	Description
timeout	TT_INTEGER	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.
retries	TT_INTEGER	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.

Result Set

`ttCkpt` returns no results.

Examples

```
CALL ttCkpt();
```

See Also

- ttCkptBlocking
- ttCkptConfig
- ttCkptHistory

ttCkptBlocking

This procedure performs a blocking checkpoint operation. A checkpoint operation saves the in-memory image of a database to files, known as checkpoint 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.
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.

Related Views

This procedure has no related views.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Syntax

```
ttCkptBlocking([timeout], [retries])
```

Parameters

`ttCkptBlocking` has these optional parameters:

Parameter	Type	Description
timeout	TT_INTEGER	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.
retries	TT_INTEGER	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.

Result Set

`ttCkptBlocking` returns no results.

Examples

```
CALL ttCkptBlocking();
CALL ttCkptBlocking(1,10);
```

> **Note**
>
> 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

This procedure reconfigures the background checkpointer dynamically or returns the currently active settings of the configuration parameters. Changes made using `ttCkptConfig` become

Copyright © 1996, 2026, Oracle and/or its affiliates.

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.

Related Views

This procedure has this related view.

SYS.V$CKPT_CONFIG

Syntax

```
ttCkptConfig(ckptFrequency, ckptLogVolume, ckptRate)
```

Parameters

`ttCkptConfig` has these parameters:

Parameter	Type	Description
ckptFrequency	`TT_INTEGER`	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.
ckptLogVolume	`TT_INTEGER`	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.
ckptRate	`TT_INTEGER`	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.

Result Set

`ttCkptConfig` returns the following results.

Column	Type	Description
ckptFrequency	`TT_INTEGER NOT NULL`	Currently active setting for checkpoint frequency in seconds.
ckptLogVolume	`TT_INTEGER NOT NULL`	Currently active setting for log volume between checkpoints in MB.
ckptRate	`TT_INTEGER NOT NULL`	Current rate at which TimesTen writes checkpoints to disk.

Examples

To view the current settings of the background checkpointer configuration parameters, use:

Copyright © 1996, 2026, Oracle and/or its affiliates.

```
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

ttCkptConfigGet

This procedure returns the currently active settings of the configuration parameters. It only helps to view the configuration parameters. To configure the parameters, see ttCkptConfig.

Required Privilege

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

Copyright © 1996, 2026, Oracle and/or its affiliates.

Related Views

This procedure has this related view.

SYS.V$CKPT_CONFIG

Syntax

```
ttCkptConfigGet(ckptFrequency, ckptLogVolume, ckptRate)
```

Parameters

`ttCkptConfigGet` has no parameters.

Result Set

`ttCkptConfigGet` returns the following results.

Column	Type	Description
ckptFrequency	TT_INTEGER NOT NULL	Currently active setting for checkpoint frequency in seconds.
ckptLogVolume	TT_INTEGER NOT NULL	Currently active setting for log volume between checkpoints in MB.
ckptRate	TT_INTEGER NOT NULL	Current rate at which TimesTen writes checkpoints to disk.

Examples

To view the current settings of the background checkpointer configuration parameters, use:

```
CALL ttCkptConfigGet;
< 600, 32, 0 >
1 row found.
```

See Also

- CkptFrequency
- CkptLogVolume
- ttCkpt
- ttCkptHistory

ttCkptHistory

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.

Related Views

This procedure has this related view.

SYS.V$CKPT_HISTORY

Copyright © 1996, 2026, Oracle and/or its affiliates.

Syntax

```
ttCkptHistory()
```

Parameters

`ttCkptHistory` has no parameters.

Result Set

`ttCkptHistory` returns the result set:

Column	Type	Description
startTime	`TT_TIMESTAMP NOT NULL`	Time when the checkpoint was begun.
endTime	`TT_TIMESTAMP`	Time when the checkpoint completed.
type	`TT_CHAR (16) NOT NULL`	The type of checkpoint taken. Value is one of: `Static` - Automatically taken at database creation and at last disconnect. `Blocking` - Transaction-consistent checkpoint. `Fuzzy` - nonblocking checkpoint. The background checkpointer performs this type if possible. `None` - For temporary databases, which have no checkpoint files.
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 checkpointer. `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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Column	Type	Description
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.
Percent_Complete	TT_INTEGER	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.
ckptVNo	TT_INTEGER NOT NULL	The checkpoint sequence number that is incremented for each checkpoint.
logsPurged	TT_BIGINT	The number of log files purged by this checkpoint.
bookmarkName	TT_VARCHAR (30) INLINE	The name of the log hold up to which this checkpoint purged log records. For example: Backup, Checkpoint, or Oldest Transaction Undo.
additional_details	TT_VARCHAR (1000)	Additional information provided for this checkpoint, such as error codes and timestamps for multiple failed checkpoints.

Examples

This example shows a checkpoint in progress:

Copyright © 1996, 2026, Oracle and/or its affiliates.

```
< 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, 2019-02-05 16:55:48.188764,
Fuzzy           , Completed       , Checkpointer    ,
Background      , <NULL>,
1, 0, 8964304, 294, 33554432, 291, 5677288, 27, 1019512,
1065408, <NULL>, 5, 0, Checkpoint, <NULL> >

< 2019-02-05 16:54:47.106110, 2019-02-05 16:54:47.723379,
Static          , Completed       , Subdaemon       ,
FinalCkpt       , <NULL>,
0, 0, 8960328, 294, 33554432, 291, 5677288, 256, 33157172,
5321548, <NULL>, 4, 0, Checkpoint, <NULL> >

< 2019-02-05 16:54:41.633792, 2019-02-05 16:54:42.568469,
Blocking        , Completed       , User            ,
BuiltIn         , <NULL>,
1, 0, 8958160, 294, 33554432, 291, 5677288, 31, 1162112,
6604976, <NULL>, 3, 0, Checkpoint, <NULL> >

< 2019-02-05 16:54:37.438827, 2019-02-05 16:54:37.977301,
Static          , Completed       , User            ,
DbCreate        , <NULL>,
0, 0, 1611984, 93, 33554432, 92, 1853848, 93, 33554432,
1854052, <NULL>, 2, 0, Checkpoint, <NULL> >

< 2019-02-05 16:54:36.861728, 2019-02-05 16:54:37.438376,
Static          , Completed       , User            ,
DbCreate        , <NULL>,
1, 0, 1609936, 93, 33554432, 92, 1853848, 93, 33554432,
1854052, <NULL>, 1, 0, Checkpoint, <NULL> >
```

This example shows that an error occurred during the most recent checkpoint attempt, which was a user-initiated checkpoint:

```
< 2019-02-05 16:57:14.476860, 2019-02-05 16:57:14.477957,
Fuzzy           , Failed , User            ,
BuiltIn         , 847,
1, <NULL>, <NULL>, 0, 0, 0, 0, 0, 0, 0, <NULL>, 7, 0, <NULL>,
Errors   1: TT0847: 16:57:14 (2019-02-05) >

< 2019-02-05 16:56:34.169520, 2019-02-05 16:56:59.715451,
Fuzzy           , Completed       , User            ,
BuiltIn         , <NULL>,
0, 0, 8966472, 294, 33554432, 291, 5677288, 5, 522000,
532928, <NULL>, 6, 0, Checkpoint, <NULL> >

< 2019-02-05 16:55:47.703199, 2019-02-05 16:55:48.188764,
Fuzzy           , Completed       , Checkpointer    ,
Background      , <NULL>,
1, 0, 8964304, 294, 33554432, 291, 5677288, 27, 1019512,
1065408, <NULL>, 5, 0, Checkpoint, <NULL> >

< 2019-02-05 16:54:47.106110, 2019-02-05 16:54:47.723379,
Static          , Completed       , Subdaemon       ,
FinalCkpt       , <NULL>,
0, 0, 8960328, 294, 33554432, 291, 5677288, 256, 33157172,
5321548, <NULL>, 4, 0, Checkpoint, <NULL> >
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

```
< 2019-02-05 16:54:41.633792, 2019-02-05 16:54:42.568469,
Blocking        , Completed       , User            ,
BuiltIn         , <NULL>,
1, 0, 8958160, 294, 33554432, 291, 5677288, 31, 1162112,
6604976, <NULL>, 3, 0, Checkpoint, <NULL> >

< 2019-02-05 16:54:37.438827, 2019-02-05 16:54:37.977301,
Static          , Completed       , User            ,
DbCreate        , <NULL>,
0, 0, 1611984, 93, 33554432, 92, 1853848, 93, 33554432,
1854052, <NULL>, 2, 0, Checkpoint, <NULL> >

< 2019-02-05 16:54:36.861728, 2019-02-05 16:54:37.438376,
Static          , Completed       , User            ,
DbCreate        , <NULL>,
1, 0, 1609936, 93, 33554432, 92, 1853848, 93, 33554432,
1854052, <NULL>, 1, 0, Checkpoint, <NULL> >
```

This example selects specific columns from the checkpoint history:

```
select type, reason, bookmarkname, logsPurged from ttCkptHistory;
< 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

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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

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.

Related Views

This procedure has this related view.

SYS.V$COMMIT_BUFFER_STATS

Syntax

```
ttCommitBufferStats()
```

Parameters

`ttCommitBufferStats` takes no parameters.

Result Set

`ttCommitBufferStats` returns these results:

Column	Type	Description
overflows	TT_INTEGER NOT NULL	Total number of commit buffer overflows.
maxReached	TT_BIGINT NOT NULL	The currently used maximum for the transaction commit buffer in bytes.

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
```

> **Note**
>
> 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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

See Also

ttCommitBufferStatsReset

ttCommitBufferStatsReset

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.

Related Views

This procedure has this related view.

SYS.V$CONFIGURATION

Syntax

```
ttCommitBufferStatsReset()
```

Parameters

`ttCommitBufferStatsReset` takes no parameters.

Result Set

`ttCommitBufferStatsReset` returns no result set.

Example

```
CALL ttCommitBufferStatsReset ();
```

See Also

ttCommitBufferStats

ttCompact

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.

Related Views

This procedure has no related views.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Syntax

```
ttCompact()
```

Parameters

`ttCompact` has no parameters.

Result Set

`ttCompact` returns no results.

Examples

```
CALL ttCompact ();
```

Notes

Compacting data does not modify result addresses.

ttComputeTabSizes

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 implementation 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.

Related Views

This procedure has no related views.

Syntax

```
ttComputeTabSizes (['tblName'], [includeOutOfLine])
```

Parameters

`ttComputeTabSizes` has the parameters:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Parameter	Type	Description
tblName	TT_CHAR(61)	Name of an application table. Can include the table owner. If a value of NULL or an empty string is provided, updates the statistics for all the current tables. The type of tables that can be estimated are: • User tables, including cache group tables • Materialized views • System tables
includeOutOfLine	TT_INTEGER	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.

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 running. 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 performed, and a transaction running concurrently raises the size of the table to 300 MB, ttComputeTabSizes estimates a value between 250 and 300 MB.
- The computed size of tables includes the size of dictionaries.

See Also

ttSize

ttConfiguration

The ttConfiguration built-in procedure returns the values for most, but not all, connection attributes and some options for the current database connection.

Required Privilege

This procedure requires no privilege.

Related Views

This procedure has this related view.

Copyright © 1996, 2026, Oracle and/or its affiliates.

SYS.V$CONFIGURATION

Syntax

```
ttConfiguration(['paramName'])
```

Parameters

`ttConfiguration` has the optional parameter:

Parameter	Type	Description
paramName	`TT_VARCHAR (30)`	The name of a connection attribute or option for which you want this procedure to return the value.

Result Set

`ttConfiguration` returns the result set:

Column	Type	Description
paramName	`TT_VARCHAR (30) NOT NULL`	The names of the connection attributes specified in the connection string, returned in alphabetical order.
paramValue	`TT_VARCHAR (1024)`	The values of the connection attributes specified in the connection string.

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 some attributes, such as `ForceConnect`, may not be returned by this procedure, as well.

See Also

Connection Attributes

Copyright © 1996, 2026, Oracle and/or its affiliates.

ttContext

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.

Related Views

This procedure has this related view.

SYS.V$CONTEXT

Syntax

```
ttContext()
```

Parameters

`ttContext` has no parameters.

Result Set

`ttContext` returns the result set:

Column	Type	Description
context	`BINARY(8)`	Current connection context value.

Example

```
CALL ttContext ();
```

> **Note**
>
> 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

ttDataStoreStatus

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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Required Privilege

This procedure requires no privilege.

Related Views

This procedure has this related view.

SYS.V$DATASTORE_STATUS

Syntax

```
ttDataStoreStatus('dataStore')
```

Parameters

`ttDataStoreStatus` has the parameter:

Parameter	Type	Description
dataStore	`TT_VARCHAR (256)`	Full path name of desired database or `NULL` for all databases.

Result Set

`ttDataStoreStatus` returns the result set:

Column	Type	Description
dataStore	`TT_VARCHAR (256) NOT NULL`	Full path name of database.
PID	`TT_INTEGER NOT NULL`	Process ID.
Context	`BINARY(8) NOT NULL`	Context value of connection.
conType	`TT_CHAR (16) NOT NULL`	Type of process connected. The result can be one of the following: `application`: An ordinary application is connected. `replication`: A replication agent is connected. `subdaemon`: A subdaemon is connected. `cache agent`: A cache agent is connected.
ShmID	`TT_VARCHAR (260) NOT NULL`	A printable version of the shared memory ID that the database occupies.
connection_Name	`TT_CHAR (30) NOT NULL`	The symbolic name of the database connection.
connID	`TT_INTEGER NOT NULL`	The numeric ID of the database connection.

Examples

```
CALL ttDataStoreStatus('/data/Purchasing');
```

See Also

ttContext
ttStatus

Copyright © 1996, 2026, Oracle and/or its affiliates.

ttDBCompactConfig

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.

Related Views

This procedure has this related view.

SYS.V$DB_COMPACT_CONFIG

Syntax

```
ttDBCompactConfig([[value][,[value]][,[value]]])
```

Parameters

`ttDBCompactConfig` has the parameters:

Parameter	Type	Description
`quantum`	`TT_INTEGER`	Specifies the number of data blocks to be compacted. Values from `0` to `100000` are allowed. A value of `0` means that automatic database compaction is disabled.
`compactsPerSecond`	`TT_INTEGER`	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.
`threshold`	`TT_INTEGER`	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 runs whenever there is anything to compact in the database.

Result Set

`ttDBCompactConfig` returns the result set:

Column	Type	Description
`quantum`	`TT_INTEGER NOT NULL`	Currently active setting for `quantum`.
`compactsPerSecond`	`TT_INTEGER NOT NULL`	Currently active setting for `compactsPerSecond`.
`threshold`	`TT_INTEGER NOT NULL`	Currently active setting for `threshold`.

Copyright © 1996, 2026, Oracle and/or its affiliates.

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 `ttCkpt` procedure in

Copyright © 1996, 2026, Oracle and/or its affiliates.

Checkpoint Operations in the *Oracle TimesTen In-Memory Database Operations Guide*. For details about the checkpoint built-in procedure, see 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 `ttDBCompactConfig` cannot be rolled back.

See Also

ttDBConfig

ttDBConfig

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.

Related Views

This procedure has this related view.

SYS.V$DB_CONFIG

Syntax

```
ttDBConfig(['param'[,'value']])
```

Parameters

`ttDBConfig` has the parameters:

Parameter	Type	Description

param	VARCHAR2(50)	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`
		`DefragThreshold`
		`DefragSpeed`
		`DefragCycle`
		`DynamicLoadReduceContention`
		`ParReplMaxDrift`
		`PLSQL_OPEN_CURSORS`
		`PLSQL_SESSION_CACHED_CURSORS`
		`RepAgentCommitBufSize`

value	VARCHAR2(200)	The value of the system parameter. If you do not specify a value, this procedure returns the current value of the specified parameter.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Parameter / Value Pairs

These name/value pairs can be returned in the result set:

Name	Value	Description
`CacheAgentCommitBufSize`	Size in MB	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. For more details, see Improving Performance When Reclaiming Memory During Autorefresh Operations in *Oracle TimesTen In-Memory Database Cache Guide*.
`CacheAwtMethod`	`0` - SQL array method `1` - PL/SQL execution method	Determines whether PL/SQL or SQL array method is used for AWT propagation to apply changes to the Oracle database server. Setting this parameter with `ttDBConfig` overrides the connection attribute value. *Default:* 1 See the description of the CacheAWTMethod connection attribute for details.
`CacheParAwtBatchSize`	Number of rows in a batch	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. **Note:** Do not change the value of this parameter unless advised by Oracle TimesTen technical support.
`DefragThreshold`	`1` - Low `2` - Medium (default) `3` - High	Defines the level of online defragmentation. For more details, see DefragThreshold .
`DefragSpeed`	`0` - Disabled (default) `1` - Low `2` - Medium `3` - High	Specifies how fast the online defragmentation can be done. The default value indicates that the online defragmentation is turned off. For more details, see DefragSpeed.
`DefragCycle`	`0` - Default `86400` - Maximum	Allows you to specify the number of seconds to wait before the next cycle of the online defragmentation. For more details, see DefragCycle.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Name	Value	Description
DynamicLoadReduceContention	0 - Disabled 1 - Enabled	If enabled, changes the way that autorefresh and dynamic load operations coordinate, which results in reduced contention between autorefresh and dynamic load operations. • Dynamic load operations are never blocked by autorefresh operations (due to additional synchronization). • Autorefresh operations are not completely delayed by dynamic load operations. **Note** This parameter does not support dynamic log-based cache group. *Default:* 0 For more details, see Reducing Contention for Dynamic Read-Only Cache Groups with Incremental Autorefresh in *Oracle TimesTen In-Memory Database Cache Guide*.
ParReplMaxDrift	Number of seconds	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.
PLSQL_OPEN_CURSORS	Maximum number of PL/SQL cursors	Specifies the maximum number of PL/SQL cursors that can be open in a session at one time, a value from 1 to 65535, inclusive. Use this to prevent a session from opening an excessive number of cursors. Setting this parameter with ttDBConfig provides a default value for future connections. *Default*: 50 PL/SQL cursors. For more details see the description of the PLSQL_OPEN_CURSORS connection attribute.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Name	Value	Description
PLSQL_SESSION_CACHED_CURSORS	Number of session cursors to cache	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. Setting this parameter with ttDBConfig provides a default value for future connections. *Default*: 50 PL/SQL cursors. For more details, see the description of the PLSQL_SESSION_CACHED_CURSORS connection attribute.
RepAgentCommitBufSize	Size in MB	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. For more details, see Improving Performance When Reclaiming Memory During Autorefresh Operations in *Oracle TimesTen In-Memory Database Cache Guide*.

Result Set

ttDBConfig returns the result set:

Column	Type	Description
param	VARCHAR2(50)	The name of the specified parameter.
value	VARCHAR2(200)	The current value of the specified parameter. This is the new value, if you specified a value.

Examples

To retrieve the current value of the CacheParAwtBatchSize, use:

```
CALL ttDBConfig('CacheParAwtBatchSize');
<CACHEPARAWTBATCHSIZE, 125>
1 row found.
```

To set the value of the RepAgentCommitBufSize to 50 MB, use:

```
CALL ttDBConfig('RepAgentCommitBufSize', '50');
 <REPAGENTCOMMITBUFSIZE, 50>
 1 row found.
```

To set the current value of the CacheAgentCommitBufSize to 100, use:

```
CALL ttDBConfig('CacheAgentCommitBufSize', '100');
 < CACHEAGENTCOMMITBUFSIZE, 100 >
1 row found.
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

To set the recommended value for the `DefragThreshold` attribute to 2, use:

```
CALL ttDBConfig('DEFRAGTHRESHOLD', '2');
 < DEFRAGTHRESHOLD, 2 >
1 row found.
```

To set the recommended value for the `DefragSpeed` attribute to 1, use:

```
CALL ttDBConfig('DEFRAGSPEED', '1');
< DEFRAGSPEED, 1 >
1 row found.
```

To set the current value to 86400 to defragment the tables once every 24 hours, use:

```
CALL ttDBConfig('DEFRAGCYCLE', '86400');
< DEFRAGCYCLE, 86400 >
1 row found.
```

The following example sets `DynamicLoadReduceContention=1`:

```
CALL ttDbConfig('DynamicLoadReduceContention','1');
```

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 *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

Improving AWT Throughput, Configuring Batch Size for Parallel Propagation for AWT Cache Groups, and Improving Performance When Reclaiming Memory During Autorefresh Operations in *Oracle TimesTen In-Memory Database Cache Guide*.

Defragmenting TimesTen Databases in *Oracle TimesTen In-Memory Database Operations Guide*.

ttDBWriteConcurrencyModeGet

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.

Related Views

This procedure has this related view.

SYS.V$DB_WRTE_CONCURRENCY_MODE

Syntax

```
ttDBWriteConcurrencyModeGet()
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

Parameters

`ttDBWriteConcurrencyModeGet` has no parameters:

Result Set

`ttDBWriteConcurrencyModeGet` returns the result set:

Column	Type	Description
ts	`TIMESTAMP NOT NULL`	Time at which the status information was collected.
mode	`TT_INTEGER NOT NULL`	The write concurrency mode: `0` - Optimize according to hints and standard optimization techniques. 1- Optimize for concurrent write operations.
operation	`VARCHAR2 (50)`	The transition status of the write concurrency mode. Either: `NULL` - Not in transition. `TRANSITIONING TO MODE=`*n* where *n*= `0` or `1`.
status	`VARCHAR2 (100) NOT NULL`	The status of the write concurrency mode transition. Either: `IN TRANSITION` or `COMPLETE`.
msg	`VARCHAR2 (5000)`	`NULL` or a status explanation message.

Examples

The following example shows how to determine if your database is optimized for concurrent write operations:

```
CALL ttDBWriteConcurrencyModeGet();

< 2013-09-23 13:48:21.207599, 1, <NULL>, COMPLETE, <NULL> >
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

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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Required Privilege

This procedure requires `ADMIN` privilege.

Related Views

This procedure has no related views.

Syntax

```
ttDBWriteConcurrencyModeSet(mode, wait)
```

Parameters

`ttDBWriteConcurrencyModeSet` has these parameters:

Parameter	Type	Description
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:

```
CALL ttDBWriteConcurrencyModeSet(0,0);
```

> **Note**
>
> 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.

See Also

ttDBWriteConcurrencyModeGet

Copyright © 1996, 2026, Oracle and/or its affiliates.

ttDurableCommit

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.

Related Views

This procedure has no related views.

Syntax

```
ttDurableCommit()
```

Parameters

`ttDurableCommit` has no parameters.

Result Set

`ttDurableCommit` returns no results.

Example

```
CALL ttDurableCommit ();
```

> **Note**
>
> 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 bear 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 Uninterruptible Power Supply (UPS).

ttHeapInfo

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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Related Views

This procedure has this related view.

SYS.V$HEAP_INFO

Syntax

```
ttHeapInfo('name')
```

Parameters

`ttHeapInfo` has the parameter:

Parameter	Type	Description
name	`TT_CHAR (30)`	Name of the heap for which you would like information. If not specified, `ttHeapInfo` lists the names of all the heaps and their metrics.

Result Set

`ttHeapInfo` returns the result:

Column	Type	Description
name	`TT_CHAR (30) NOT NULL`	Name of the heap for which the heap memory info is being returned.
size	`TT_BIGINT NOT NULL`	The allocated sizes of the heap memory.
inUse	`TT_BIGINT NOT NULL`	The amount of heap memory in use.
highWater	`TT_BIGINT NOT NULL`	The maximum amount of heap memory used.
`freeDeferred`	`TT_BIGINT NOT NULL`	The number of deferred freed heap memory buffers.

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

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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Related Views

This procedure has this related view.

SYS.V$HOST_NAME

Syntax

```
ttHostnameGet()
```

Parameters

`ttHostNameGet` has no parameters.

Result Set

`ttHostNameGet` returns the result:

Column	Type	Description
hostName	TT_VARCHAR (200)	The current default local host setting for the database. If a default has not been supplied then the current host name is returned.

Examples

```
CALL ttHostNameGet ();
< myhost >
1 row found.
```

See Also

ttHostNameSet

ttHostNameSet

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.

Related Views

This procedure has no related views.

Syntax

```
ttHostnameSet('hostName')
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

Parameters

`ttHostNameSet` has the parameter:

Parameter	Type	Description
hostName	`TT_VARCHAR (200)`	The required default name for the local computer. To clear the default value, specify `NULL`.

Result Set

`ttHostNameSet` returns no results.

Examples

```
CALL ttHostNameSet ('alias1');
```

> **Note**
>
> The 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

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.

Related Views

This procedure has no related views.

Syntax

```
ttIndexAdviceCaptureDrop([captureLevel])
```

Parameters

`ttIndexAdviceCaptureDrop` has this optional parameter:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Parameter	Type	Description
captureLevel	TT_INTEGER	Supported 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.

Result Set

ttIndexAdviceCaptureDrop returns no results.

Examples

```
CALL ttIndexAdviceCaptureDrop;
```

Notes

- To drop both connection level and database level captures, invoke the command twice, once 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

ttIndexAdviceCaptureEnd
ttIndexAdviceCaptureInfoGet
ttIndexAdviceCaptureOutput
ttIndexAdviceCaptureStart

ttIndexAdviceCaptureEnd

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.

Related Views

This procedure has no related views.

Syntax

```
ttIndexAdviceCaptureEnd([captureLevel])
```

Parameters

ttIndexAdviceCaptureEnd has this optional parameter:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Parameter	Type	Description
captureLevel	TT_INTEGER	Supported values for the capture level are: 0 - Ends index advice capture at the connection level for the current connection. This is the default. 1 - Ends index advice capture at the database level.

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

ttIndexAdviceCaptureInfoGet

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.

Related Views

This procedure has no related views.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Syntax

```
ttIndexAdviceCaptureInfoGet()
```

Parameters

`ttIndexAdviceCaptureInfoGet` has no parameters.

Result Set

`ttIndexAdviceCaptureInfoGet` returns the result set:

Columns	Type	Description
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.
captureLevel	TT_INTEGER	The level of the most recent capture. This row is not returned if no capture has been initiated.
captureMode	TT_INTEGER	The mode of the most recent capture. This row is not returned if no capture has been initiated.
numPrepared	TT_INTEGER	The number of prepared statements during the capture period. This value is NULL if no capture has been initiated.
numExecuted	TT_INTEGER	The number of executed statements during the capture period. This value is NULL if no capture has been initiated.
captureStartTime	TT_TIMESTAMP	The time stamp taken at the start of the capture period. This row is not returned if no capture has been initiated.
captureEndTime	TT_TIMESTAMP	The time stamp taken at the end of the capture period. This value is NULL if no capture is still in progress.

Examples

This example shows capture information for a completed connection level capture for 363 prepared statements and 369 executed statements:

```
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.
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

> **Note**
>
> 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

ttIndexAdviceCaptureOutput

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.

Related Views

This procedure has this related view.

SYS.V$INDEX_ADVICE_OUTPUT

Syntax

```
ttIndexAdviceCaptureOutput([captureLevel])
```

Parameters

`ttIndexAdviceCaptureOutput` has this optional parameter:

Parameter	Type	Description
captureLevel	TT_INTEGER	Supported values for the capture level are: 0 - Outputs index advice at the connection level for the current connection. This is the default value. 1 - Outputs index advice at the database level.

Result Set

`ttIndexAdviceCaptureOutput` returns the result set:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Column	Type	Description
stmtCount	TT_INTEGER	The number of statements in the captured workload that would have benefited from this index if it were present.
createStmt	TT_VARCHAR (8300) NOT NULL	The executable statement that can create the recommended index.

Examples

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.
```

> **Note**
>
> All names returned are fully schema qualified.

See Also

- ttIndexAdviceCaptureDrop
- ttIndexAdviceCaptureEnd
- ttIndexAdviceCaptureInfoGet
- ttIndexAdviceCaptureStart

ttIndexAdviceCaptureStart

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.

Related Views

This procedure has no related views.

Syntax

```
ttIndexAdviceCaptureStart([captureLevel], [captureMode])
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

Parameters

`ttIndexAdviceCaptureStart` has these optional parameters:

Parameter	Type	Description
`captureLevel`	`TT_INTEGER`	Supported values for the capture level are: `0` - Outputs index advice at the connection level for the current connection. This is the default value. `1` - Outputs index advice at the database level.
`captureMode`	`TT_INTEGER`	Supported values for the capture mode are: `0` - Provides complete capture of index advice including execution of the SQL statements. This is the default. `31` - Capture is based only on the computed statistics and plan analysis. Queries (`SELECT` statements only) are prepared but not executed. This mode can only be used with connection level captures (`captureLevel=0`).

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);
```

> **Note**
>
> 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

ttLatchStatsGet

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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

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.

Related Views

This procedure has no related views.

Syntax

```
ttLatchStatsGet (level, operation)
```

Parameters

`ttLatchStatsGet` has these parameters:

Parameter	Type	Description
level	`TT_CHAR (16)`	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).
operation	`TT_CHAR (16)`	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. `showandtell` - Show all the contention points.

Result Set

Results sets are divided into two types: contention point and statistics.

`ttLatchStatsGet` returns the result set for contention points. These describe the location of contention.

Column	Type	Description
`callerName`	`TT_VARCHAR(40) NOT NULL`	Function name containing the contention point.
`fileName`	`TT_VARCHAR(30) NOT NULL`	The file that contains the *`callerName`* function.
`lineNo`	`TT_INTEGER NOT NULL`	The line number of the *`fileName`* file.
`description`	`TT_VARCHAR(100) NOT NULL`	Description of this contention point.

`ttLatchStatsGet` returns the result set for statistics. These describe detailed statistics about this contention point:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Column	Type	Description
connName	TT_VARCHAR(32) NOT NULL	The name of the connection experiencing contention.
spinCount	TT_BIGINT NOT NULL	The number of times the *connName* connection has spun on this contention point.
access	TT_BIGINT NOT NULL	The number of times the *connName* connection has used this contention point.
sleepCnt	TT_INTEGER NOT NULL	The number of times the *connName* connection has slept on this contention point.
firstTry	TT_INTEGER NOT NULL	The number of times the *connName* connection has used this contention point without experiencing contention.
collisions	TT_INTEGER NOT NULL	The number of times the *connName* connection has used this contention point and experienced contention.
avgSpin	TT_BIGINT NOT NULL	The average number of times the *connName* connection has spun on this contention point.

Examples

The following example shows an example result set for a call to `ttLatchStatsGet`.

```
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

ttLoadFromOracle

This procedure takes a TimesTen table name, an Oracle `SELECT` statement and the number of threads for parallel load. It runs the query on the Oracle database and loads the result set into the specified TimesTen table. While performing the load, an implicit commit is run 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 Without Cache 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 run the query on the Oracle database.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Related Views

This procedure has no related views.

Syntax

```
ttLoadFromOracle(['tblOwner'], 'tblName', 'Query' [,numThreads], 'Options')
```

Parameters

`ttLoadFromOracle` has these parameters:

Parameter	Type	Description
tblOwner	TT_CHAR (30)	TimesTen table owner (optional). If not provided, the connection ID is used.
tblName	TT_CHAR (30) NOT NULL	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.
SelectSQL	TT_VARCHAR (409600) NOT NULL	A SELECT query on an Oracle database to derive the table column definition. The query on an Oracle database cannot have any parameter bindings. Provide any expressions in the SELECT list with a column alias. Otherwise, an implementation dependent column name is assumed and the expression is not evaluated.
numThreads	TT_INTEGER	Number of threads for parallel load (optional). If NULL, defaults to 4. Provides parallel loading for tables. Specifies the number of loading threads to run concurrently. One thread performs the bulk fetch from the Oracle database and the other threads perform the inserts into TimesTen. Each thread uses its own connection or transaction. The minimum value for NumThreads is 2. The maximum value is 10. If you specify a value greater than 10, TimesTen assigns the value 10. You can also use the readers option to specify the total number of threads from the numThreads parameter to use for bulk fetching from the Oracle database.
Options	TT_VARCHAR (1000)	Option string, specified as key=value pairs. For the supported values, see Options String. Defaults to NULL. See the table below for more information.

Options String

The options are specified as key = value pairs and the pairs are separated by semi-colons.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option parameter	Value	Description
`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.
`IgnoreDuplicates`	`Y` or `N`	This option makes the `ttLoadFromOracle` built-in procedure ignore uniqueness constraint violations, which results in duplicates being ignored. You can only use this option if the TimesTen table has a uniqueness constraint on it. The default value is `N`.
`ResumeFromSCN`	*`scn_of_last_load`*	This option resumes the load operation from the specific SCN. When the `resumeFromSCN` option is enabled, it automatically ignores duplicates. The default value is the latest SCN.
`DirectLoad`	`Y` or `N`	This option enables a bulk insert mode which has performance benefits. This option is only supported in TimesTen IMDB. The default value is `N`.
`readers`	`numThreads > value > 0`	This option specifies the total number of threads from the `numThreads` parameter to use for bulk fetching from the Oracle database. For example, if you specify a numThreads parameter of 8 and a readers option of 3, 3 threads bulk fetch data from the Oracle database and 5 threads load data into the TimesTen database.

Result Set

`ttLoadFromOracle` returns the result set:

Column	Type	Description
`numRows`	`TT_BIGINT NOT NULL`	Number of rows loaded.
`numErrors`	`TT_INTEGER NOT NULL`	Number of rows with errors.
`errCode`	`TT_INTEGER`	TimesTen error code, one of: `0` - Load completed successfully without errors `-1` - Load completed successfully with errors `-2` - Load terminated early with errors `-3` - Load terminated early with a fatal error, for example, an out-of-space error, a loss of connection or an invalidation.
`errMsg`	`VARCHAR2 (4000)`	Error message, containing: • Start and end time of load • Statement • `SCN` used to query the data • Number of rows with errors • Number of rows loaded

Copyright © 1996, 2026, Oracle and/or its affiliates.

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');
< 99, 0, 0, 'Started=2014-08-01 13:48:21; Ended=2014-08-01 13:48:23;
 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

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 for full details of the different locking levels.

Required Privilege

This procedure requires the `ADMIN` privilege.

Related Views

This procedure has no related views.

Syntax

`ttLockLevel('lockLevel')`

Parameters

`ttLockLevel` has the parameter:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Parameter	Type	Description
lockLevel	TT_CHAR (20) NOT NULL	Locking level for the connection.

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

ttLockWait

ttLockWait

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.

Related Views

This procedure has no related views.

Syntax

```
ttLockWait(seconds)
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

Parameters

`ttLockWait` has the required parameters:

Parameter	Type	Description
seconds	`NUMBER (8,1) NOT NULL`	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.

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);
```

> **Note**
>
> 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

ttLogHolds

This procedure returns information about transaction log holds. It includes 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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Related Views

This procedure has this related view.

SYS.V$LOG_HOLDS

Syntax

```
ttLogHolds()
```

Parameters

`ttLogHolds` has no parameters.

Result Set

`ttLogHolds` returns the result set:

Column	Type	Description
HoldLFN	TT_INTEGER NOT NULL	Returns the transaction log file number of the hold.
HoldLFO	TT_BIGINT NOT NULL	Returns the transaction log file offset of the hold.
type	TT_CHAR (30) NOT NULL	Returns the type of hold, one of: Checkpoint Replication Backup XLA Long-Running Transaction Long-Running XA Transaction TTGrid Replica Element Duplicate

Copyright © 1996, 2026, Oracle and/or its affiliates.

Column	Type	Description
description	TT_VARCHAR (1024) NOT NULL	Describes the type-specific object for which the hold was created. Each description corresponds with the Type returned. Descriptions are one of: • The name of the checkpoint file • The name of the standby master • The name of the replication subscriber • _ORACLE when tracking AWT cache group propagation • The parallel replication track ID used by the subscriber • The backup path • The name of the persistent XLA subscription and the process ID of the last process to open it, if it is open • The XID (transaction ID) of the XA transaction • The TimesTen transaction ID of the long-running transaction • The index of the replica in the partition table, the replica id, the index of the local element, the version of partition table for the replica log hold, and the index for the loop of the list of replicas. • The string `Log hold of Element Duplicate used by LBCU`.

Examples

```
Command> call ttLogHolds();
< 0, 1148544, Long-Running XA Transaction ,
0x1-476c6f62616c-5861637431 >
< 0, 1149752, Long-Running Transaction, 4.2 >
< 0, 1149992, Checkpoint , sample.ds1 >
< 0, 1150168, Checkpoint , sample.ds0 >
```

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:MASTER2 >
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 >
< 0, 19048448, Checkpoint                  , cachealone1.ds1 >
< 0, 19050904, Replication                 , ADC6160529:_ORACLE >
3 rows found.
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

ttMonitorHighWaterReset

This procedure sets values for HIGH_wATER columns. It 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.

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

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.

Related Views

This procedure has no related views.

Syntax

```
ttOptClearStats('tblName', invalidate)
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

Parameters

`ttOptClearStats` has these 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 rerun, 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

Clears the statistics for the `SALLY.ACCTS` table and reprepares all commands that affect the `ACCTS` table.

```
CALL ttOptClearStats ( 'SALLY.ACCTS', 1 );
```

Clears the statistics for all the current user's tables and reprepares all commands that affect these tables.

```
CALL ttOptClearStats();
```

Clears the statistics for all the current user's tables without repreparing commands that reference these tables.

```
CALL ttOptClearStats('', 0);
```

See Also

- ttOptEstimateStats
- ttOptSetColIntvlStats
- ttOptSetFlag
- ttOptSetOrder
- ttOptSetTblStats
- ttOptUpdateStats
- ttPLSQLMemoryStats

ttOptCmdCacheInvalidate

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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

- 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 `DDL` privilege.

Related Views

This procedure has no related views.

Syntax

```
ttOptCmdCacheInvalidate('tblName', invalidate)
```

Parameters

`ttOptCmdCacheInvalidate` has these parameters:

Parameter	Type	Description
tblname	`TT_CHAR(61)`	The name of the table for which the dependent commands should be invalidated or recompiled.
invalidate	`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

`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

Copyright © 1996, 2026, Oracle and/or its affiliates.

ttOptUpdateStats
ttPLSQLMemoryStats

ttOptEstimateStats

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.

Related Views

This procedure has this related view.

SYS.V$OPT_COL_STATS

Syntax

```
ttOptEstimateStats(['tblName'], [invalidate], 'sampleStr')
```

Parameters

`ttOptEstimateStats` has these parameters:

Parameter	Type	Description
`tblName`	`TT_CHAR(61)`	Name of an application table. Can include table owner. If *`tblName`* 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.
`invalidate`	`TT_INTEGER`	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. The *`invalidate`* parameter is optional and defaults to 0.
`sampleStr`	`TT_VARCHAR (255) NOT NULL`	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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

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.
- For performance reasons, `ttOptEstimateStats` does not hold a lock on tables or rows when computing statistics. However, it holds a lock on the TimesTen system tables. 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.

See Also

ttOptSetColIntvlStats
ttOptSetFlag
ttOptSetOrder
ttOptSetTblStats
ttOptUpdateStats
ttPLSQLMemoryStats

Copyright © 1996, 2026, Oracle and/or its affiliates.

ttOptGetColStats

This procedure returns statistics information in text format.

See ttOptSetColIntvlStats for the layout of the statistics.

Required Privilege

This procedure requires the SELECT privilege on the specified tables.

Related Views

This procedure has this related view.

SYS.V$OPT_COL_STATS

Syntax

```
ttOptGetColStats('tblName', 'colName')
```

Parameters

ttOptGetColStats has these parameters:

Parameter	Type	Description
tblName	TT_CHAR (61)	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.
colName	TT_CHAR (30)	Name of the column for which statistics should be returned. If NULL is passed, statistics for all columns in the specified table are returned.

Result Set

ttOptGetColStats returns the result set:

Column	Type	Description
tblName	TT_CHAR (30)	Name of the table. Using a synonym to specify a table name is not supported.
colName	TT_CHAR (30)	Name of the column.
stats	TT_VARCHAR (409600) NOT NULL	Statistics in text form.

Examples

```
CALL ttOptGetColStats ();
< T1 , X1, (2, 10, 10, 100 (,4, 40, 10 ,1, 10, 5) ,
(4, 20, 20 ,11, 20, 15) )>
```

See Also

ttOptSetColStats

Copyright © 1996, 2026, Oracle and/or its affiliates.

ttOptSetColIntvlStats

ttOptGetFlag

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.get`*`XXX`*`()` 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.

Related Views

This procedure has no related views.

Syntax

```
ttOptGetFlag('flagName')
```

Parameters

`ttOptGetFlag` has the parameter:

Parameter	Type	Description
`flagName`	`TT_CHAR (32)`	Name of the flag whose value is to be returned. If `NULL` is passed, the values of all flags are returned.

Result Set

`ttOptGetFlag` returns the result set:

Column	Type	Description
`flagName`	`TT_VARCHAR (32) NOT NULL`	Name of the flag. See "ttOptSetFlag" for a description of possible flag values.
`value`	`TT_INTEGER NOT NULL`	Current flag value, either `0` or `1`.

Examples

```
CALL ttOptGetFlag('TmpHash');
```

See Also

ttOptSetFlag

Copyright © 1996, 2026, Oracle and/or its affiliates.

ttOptGetMaxCmdFreeListCnt

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.

Related Views

This procedure has this related view.

SYS.V$OPT_MAX_CMD_FREELIST_CNT

Syntax

```
ttOptGetMaxCmdFreeListCnt()
```

Parameters

`ttOptGetMaxCmdFreeListCnt` has no parameters.

Result Set

`ttOptGetMaxCmdFreeListCnt` returns the results.

Column	Type	Description
retVal	`TT_VARCHAR (200) NOT NULL`	The size of the SQL compiled command cache.

Example

```
CALL ttOptGetMaxCmdFreeListCnt();
```

See Also

ttOptSetMaxPriCmdFreeListCnt
ttOptSetMaxCmdFreeListCnt

ttOptGetOrder

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.get`*XXX*`()` 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.

Related Views

This procedure has this related view.

Copyright © 1996, 2026, Oracle and/or its affiliates.

SYS.V$OPT_ORDER

Syntax

```
ttOptGetOrder()
```

Parameters

`ttOptGetOrder` has no parameters.

Result Set

`ttOptGetOrder` returns the result set:

Column	Type	Description
joinOrder	`TT_VARCHAR(1024) NOT NULL`	Optimizer join order for the current transaction.

Example

```
CALL ttOptGetOrder ();
```

See Also

ttOptSetOrder

ttOptSetColIntvlStats

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).

Related Views

This procedure has no related views.

Syntax

```
ttOptSetColIntvlStats('tblName', 'colName', invalidate, (stats))
```

Parameters

`ttOptSetColIntvlStats` has these parameters:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Parameter	Type	Description
tblName	TT_CHAR(61) NOT NULL	Name of an application table. Can include table owner. Using a synonym to specify a table name is not supported.
colName	TT_CHAR(30) NOT NULL	Name of a column in that table.
invalidate	TT_INTEGER	0 (no) or 1 (yes). If *invalidate* is 1, all commands that reference the affected tables are automatically prepared again when rerun. This includes commands prepared by other users. If *invalidate* is 0, the statistics are not considered to have been modified and existing commands are not reprepared.

stats	VARBINARY (409600) NOT NULL	Sets stats for the column, using the format:
		`(numInterval integer, numNull integer, totUniq integer, totTups integer,`
		`/* information for interval 1 */`
		`(numUniq integer, numTups integer, frequency of most occurred value integer, minVal, maxVal, modalVal),`
		`/* information for interval 2 */...)`
		The numUniq value is the number of unique values minus 1.
		The numTups value is the number of rows whose value is not the modal value.
		The modal value (modalVal) is the value that occurs most often in a specified interval.
		Because this parameter is a compound structure it cannot be parameterized using ODBC functions or described using the ttIsql describe command. For example, a statement like the following fails: SQLPrepare(hstmt, "call ttOptSetColIntvlStats('t1', 'c1', 1, ?)", SQL_NTS)).

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)

Copyright © 1996, 2026, Oracle and/or its affiliates.

- `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:

```
CALL ttOptSetColIntvlStats('t1', 'x1', 1, (2, 10, 10, 100,
(4, 40, 10, 1, 10, 5), (4, 20, 20, 11, 20, 15)));
```

> **Note**
>
> 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
ttOptSetColStats
ttOptSetTblStats
ttOptUpdateStats

ttOptSetColStats

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).

Related Views

This procedure has no related views.

Syntax

```
ttOptSetColStats('tblName', 'colName', numUniq, minVal,maxVal,
 invalidate, numNull)
```

Parameters

`ttOptSetColStats` has these parameters:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Parameter	Type	Description
tblName	TT_CHAR(61) NOT NULL	Name of an application table. Can include table owner. Using a synonym to specify a table name is not supported.
colName	TT_CHAR(30) NOT NULL	Name of a column in that table.
num_Uniq	TT_INTEGER NOT NULL	Number of unique values in the column.
minVal	VARBINARY(1024) NOT NULL	Minimum value in the column (possibly truncated).
maxVal	VARBINARY(1024) NOT NULL	Maximum value in the column (possibly truncated).
invalidate	TT_INTEGER	0 (no) or 1 (yes). If *invalidate* is 1, all commands that reference the affected tables are automatically prepared again when rerun. This includes commands prepared by other users. If *invalidate* is 0, the statistics are not considered to have been modified and existing commands are not reprepared.
num_Null	TT_INTEGER	Indicates the total number of NULLs in the column.

Result Set

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

ttOptSetFlag

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

Copyright © 1996, 2026, Oracle and/or its affiliates.

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.

Related Views

This procedure has this related view.

SYS.V$OPT_FLAG

Syntax

```
ttOptSetFlag('optFlag', optVal)
```

Parameters

`ttOptSetFlag` has these parameters:

Parameter	Type	Description
optFlag	`TT_CHAR(32) NOT NULL`	Name of optimizer flag.
optVal	`TT_INTEGER NOT NULL`	The value of the optimizer flag. The value is generally `0` (disable/disallow) or `1` (enable/allow), except as described under Optimizer Flags below.

Optimizer Flags

When setting the optimizer flags, use the following character strings, which are not case sensitive:

Flag	Description
`BranchAndBound`	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.
`DynamicLoadEnable`	Enables or disables dynamic load of a single cache instance from an Oracle database to a TimesTen dynamic cache group. By default, dynamic load of data from an Oracle database is enabled.
`DynamicLoadErrorMode`	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.
`DynamicLoadMultiplePKs`	Enables or disables dynamic load of multiple cache instances (providing multiple primary keys) from an Oracle database to a TimesTen dynamic cache group that contains only a root table. By default, dynamic load multiple cache instances using multiple primary keys from an Oracle database is enabled.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Flag	Description
DynamicLoadRootTbl	Enables or disables dynamic load of multiple cache instances (without providing multiple primary keys) from an Oracle database to a TimesTen dynamic cache group that contains only a root table. By default, dynamic load multiple cache instances from an Oracle database is disabled.
FirstRow	Enables or disables first row optimization in a SELECT, UPDATE or DELETE statement. If the SQL keyword FIRST is used in the SQL statement, it takes precedence over this optimizer hint. The FIRST keyword enables first row optimization.
ForceCompile	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.
GenPlan	Enables or disables the creation of entries in the PLAN table for the rest of the transaction. For an example, see Instruct TimesTen to Store the Plan in the System PLAN Table in *Oracle TimesTen In-Memory Database Operations Guide*.
Hash	Enables or disables the use of existing hash indexes in indexed table scans.
HashGb	Enables or disables the use of hash groups.
IndexedOR	Enables or disables serialized table scans. If disabled, TimesTen uses serialized table scans for IN...list conditions, else TimesTen uses multiple index scans for an OR condition.
MergeJoin	Enables or disables the use of merge joins.
NestedLoop	Refers to a common way of joining two tables.
NoRemRowIdOpt	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 SELECT list.
PassThrough	Temporarily changes the pass through level for TimesTen Cache applications. The pass through level can be set at any time and takes effect immediately. Supported values for this flag are: 0 - (default) - SQL statements are run only on TimesTen. 1 - INSERT, UPDATE and DELETE statements are run 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 run 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.
Range	Enables or disables the use of existing range indexes in indexed table scans.
Rowid	Enables or disables the use of Row IDs.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Flag	Description
RowLock	Allows or disallows the optimizer to consider using row locks.
Scan	Refers to full table scans.
ShowJoinOrder	Shows the join order of the tables in an optimizer scan.
TblLock	Enables or disables the optimizer to consider using table locks.
TmpHash	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.
TmpRange	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.
TmpTable	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.
UseBoyerMooreStringSearch	Enables or disables the Boyer-Moore string search algorithm. If enabled, Boyer-Moore string search algorithm is enabled. This can improve performance of LIKE operations.

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.

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:

```
ttOptSetFlag ( 'TmpTable', 0 )
```

Similarly, an application can specify a preference for MergeJoin:

```
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

Copyright © 1996, 2026, Oracle and/or its affiliates.

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 `DynamicLoadMultiplePKs` and `DynamicLoadRootTbl` are enabled, `DynamicLoadMultiplePKs` has precedence.

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:

TblLock status	RowLock status	Effect on the optimizer
Disabled	Disabled	Use row-level locking.
Enabled	Disabled	Use table-level locking.
Disabled	Enabled	Use row-level locking.
Enabled	Enabled	Optimizer chooses row-level or table-level locking.

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

```
CALL ttOptSetFlag ('TmpHash', 1);
```

> **Note**
>
> 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 Optimizer Hints to understand the behavior of each style of hint.

See Also

- ttOptEstimateStats
- ttOptGetFlag
- ttOptGetOrder
- ttOptSetColIntvlStats
- ttOptSetOrder
- ttOptSetTblStats
- ttOptUpdateStats
- ttPLSQLMemoryStats

Copyright © 1996, 2026, Oracle and/or its affiliates.

ttOptSetMaxCmdFreeListCnt

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:

Parameter	Type	Description
maxCnt	`TT_INTEGER NOT NULL`	The max number of free SQL compiled commands for regular tables.

Result Set

`ttOptSetMaxCmdFreeListCnt` returns no results.

Examples

```
CALL ttOptSetMaxCmdFreeListCnt(40);
```

See Also

ttOptGetMaxCmdFreeListCnt

ttOptSetMaxPriCmdFreeListCnt

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.

Related Views

This procedure has no related views.

Syntax

```
ttOptSetMaxPriCmdFreeListCnt(maxCnt)
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

Parameters

`ttOptSetMaxPriCmdFreeListCnt` has the required parameter:

Parameter	Type	Description
maxCnt	`TT_INTEGER NOT NULL`	The size of the SQL compiled command cache.

Result Set

`ttOptSetMaxPriCmdFreeListCnt` returns no results.

Examples

```
CALL ttOptSetMaxPriCmdFreeListCnt(40);
```

See Also

ttOptGetMaxCmdFreeListCnt
ttOptSetMaxCmdFreeListCnt

ttOptSetOrder

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.

Related Views

This procedure has no related views.

Syntax

```
ttOptSetOrder('joinOrder')
```

Parameters

`ttOptSetOrder` has the required parameter:

Parameter	Type	Description
joinOrder	`TT_VARCHAR(1024)`	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 *joinOrder* is not specified the query optimizer reverts to its default behavior.

Result Set

`ttOptSetOrder` returns no results.

Copyright © 1996, 2026, Oracle and/or its affiliates.

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 performing 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.prepareCall` 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;

Copyright © 1996, 2026, Oracle and/or its affiliates.

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, Times 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 Optimizer Hints to understand the behavior of each style of hint.

See Also

ttOptEstimateStats
ttOptGetFlag
ttOptGetOrder
ttOptSetColIntvlStats
ttOptSetFlag
ttOptSetTblStats
ttOptUpdateStats
ttPLSQLMemoryStats

ttOptSetTblStats

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).

Related Views

This procedure has no related views.

Syntax

```
ttOptSetTblStats('tblName', numRows, invalidate)
```

Parameters

`ttOptSetTblStats` has these parameters:

Parameter	Type	Description
tblName	`TT_CHAR(61) NOT NULL`	Name of an application table. Can include table owner. Using a synonym to specify a table name is not supported.
num_Rows	`TT_INTEGER NOT NULL`	Number of rows in the table.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Parameter	Type	Description
invalidate	TT_INTEGER	0 (no) or 1 (yes). If *invalidate* is 1, all commands that reference the affected tables are automatically prepared again when rerun, 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.

Result Set

ttOptSetTblStats returns no results.

Examples

```
CALL ttOptSetTblStats ( 'ACCTS', 10000, 0 );
```

> **Note**
>
> 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

This procedure returns the join order of the last prepared or run 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.

Related Views

This procedure has this related view.

Copyright © 1996, 2026, Oracle and/or its affiliates.

SYS.V$OPT_JOIN_ORDER

Syntax

```
ttOptShowJoinOrder()
```

Parameters

`ttOptShowJoinOrder` has no parameters.

Result Set

`ttOptShowJoinOrder` returns the result:

Column	Type	Description
joinOrder	`TT VARCHAR (4096) NOT NULL`	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.

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
ttOptGetOrder
ttOptSetColIntvlStats
ttOptSetFlag
ttOptSetOrder
ttOptSetTblStats
ttOptUpdateStats
ttPLSQLMemoryStats

ttOptStatsExport

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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Related Views

This procedure has this related view.

SYS.V$OPT_STATS

Syntax

```
ttOptStatsExport('tblName')
```

Parameters

`ttOptStatsExport` has the parameter:

Parameter	Type	Description
tblName	TT_CHAR (61)	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.

Result Set

`ttOptStatsExport` returns the result set:

Column	Type	Description
stmt	TT_VARCHAR (8300) NOT NULL	The set of statements required to restore the table(s) statistics to the current state.

Examples

```
CALL ttOptStatsExport('MyTable');
```

See Also

Create Script to Regenerate Current Table Statistics in the *Oracle TimesTen In-Memory Database Operations Guide*.

ttOptUpdateStats

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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

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.

Related Views

This procedure has no related views.

Syntax

```
ttOptUpdateStats(['tblName'], [invalidate], [option])
```

Parameters

ttOptUpdateStats has these parameters:

Parameter	Type	Description
tblName	TT_CHAR(61)	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.
invalidate	TT_INTEGER	0 (no) or 1 (yes). If *invalidate* is 1, marks all commands for reprepare on next run except ALTER TABLE DROP TABLE, and the ALTER TABLE ADD COLUMN FOR SELECT * FROM TABLE statements. These exceptions require manual reprepare. If *invalidate* is 0, the statistics are not considered to have been modified and existing commands are not reprepared. The *invalidate* parameter is optional and defaults to 0.
option	TT_INTEGER	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 *option* parameter is optional and defaults to 0. See the notes below for more information.

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 run.

Copyright © 1996, 2026, Oracle and/or its affiliates.

```
CALL ttOptUpdateStats('', 1);
```

Updates all the current user's tables and causes commands on those tables to be reprepared when they are next run.

```
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
- 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, it holds a lock on the TimesTen system tables. Computing statistics can still slow performance. Estimating statistics generally provides better performance than computing exact statistics. See ttOptEstimateStats 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

Copyright © 1996, 2026, Oracle and/or its affiliates.

ttOptUseIndex

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.

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 `TT_VARCHAR(1024)` with these components:

Component	Description
`IndexName`	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.
`CorrelationName`	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.
`0 | 1`	Disables(`0`) or enables (`1`) the use of the index specified by *`IndexName`*.

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');
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

> **Note**
>
> 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

ttPageLevelTableInfo

TimesTen frees empty pages and reuses empty slots on non-full pages. The `ttPageLevelTableInfo` built-in procedure shows the page allocation for each table to determine when TimesTen is reusing empty slots and freeing empty pages or if new pages are allocated to store new rows.

Required Privilege

This procedure requires `SELECT` privilege to see the current values.

Related Views

This procedure has no related views.

Syntax

```
ttPageLevelTableInfo( [ tblOwner tt_char(31) ], [ tblName tt_char(31)])
```

Parameters

`ttPageLevelTableInfo` has these optional parameters:

Parameter	Type	Description
`tblOwner`	`TT_CHAR (31)`	TimesTen table owner. If not specified, uses the current user.
`tblName`	`TT_CHAR (31)`	Name of an application table. Only user tables can be specified. No global temporary tables can be specified.

Result Set

`ttPageLevelTableInfo` returns these results:

Parameter	Type	Description
`tblOwner`	`TT_CHAR (31) NOT NULL`	TimesTen table or cache group owner.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Parameter	Type	Description
tblName	TT_CHAR (31) NOT NULL	Name of an application table. Provide the cache root table for a cache group.
card	TT_BIGINT NOT NULL	Table tuple count
logical_pgcnt	TT_BIGINT NOT NULL	Count of logical pages
logical_nonFullPages	TT_BIGINT NOT NULL	Count of logical pages with empty slots
logical_freeSlots	TT_BIGINT NOT NULL	Total count of empty slots in logical pages
physical_pgcnt	TT_BIGINT NOT NULL	Count of physical pages
physical_nonFullPages0	TT_BIGINT NOT NULL	Count of physical pages with empty slots in directory list 0
physical_nonFullPages1	TT_BIGINT NOT NULL	Count of physical pages with empty slots in directory list 1
physical_nonFullPages2	TT_BIGINT NOT NULL	Count of physical pages with empty slots in directory list 2
physical_nonFullPages3	TT_BIGINT NOT NULL	Count of physical pages with empty slots in directory list 3
physical_freeSlots0	TT_BIGINT NOT NULL	Count of empty physical slots in directory list 0
physical_freeSlots1	TT_BIGINT NOT NULL	Count of empty physical slots in directory list 1
physical_freeSlots2	TT_BIGINT NOT NULL	Count of empty physical slots in directory list 2
physical_freeSlots3	TT_BIGINT NOT NULL	Count of empty physical slots in directory list 3

Examples

The following demonstrates the output received for `table1` and `table2` using the `ttPageLevelTableInfo` built-in procedure:

```
Command> vertical 1;
Command> call ttPageLevelTableInfo;

  TBLOWNER:                USER1
  TBLNAME:                 TABLE1
  CARD:                    6
  LOGICAL_PGCNT:           1
  LOGICAL_NONFULLPAGES:    1
  LOGICAL_FREESLOTS:       250
  PHYSICAL_PGCNT:          2
  PHYSICAL_NONFULLPAGES0:  0
  PHYSICAL_NONFULLPAGES1:  0
  PHYSICAL_NONFULLPAGES2:  1
  PHYSICAL_NONFULLPAGES3:  1
  PHYSICAL_FREESLOTS0:     0
  PHYSICAL_FREESLOTS1:     0
  PHYSICAL_FREESLOTS2:     253
  PHYSICAL_FREESLOTS3:     253
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

```
  TBLOWNER:                USER1
  TBLNAME:                 TABLE2
  CARD:                    3
  LOGICAL_PGCNT:           1
  LOGICAL_NONFULLPAGES:    1
  LOGICAL_FREESLOTS:       253
  PHYSICAL_PGCNT:          1
  PHYSICAL_NONFULLPAGES0:  0
  PHYSICAL_NONFULLPAGES1:  1
  PHYSICAL_NONFULLPAGES2:  0
  PHYSICAL_NONFULLPAGES3:  0
  PHYSICAL_FREESLOTS0:     0
  PHYSICAL_FREESLOTS1:     253
  PHYSICAL_FREESLOTS2:     0
  PHYSICAL_FREESLOTS3:     0

2 rows found.
```

The following demonstrates the output received for `table1` using the `ttPageLevelTableInfo` built-in procedure. The output uses vertical 0 (which is the default).

```
Command> call ttPageLevelTableInfo(user1, table1);
< USER1, TABLE1, 6, 1, 1, 250, 2, 0, 0, 1, 1, 0, 0, 253, 253 >
```

Notes

- If no parameters are supplied, this procedure returns all user tables, except for global temporary tables, on which the current user has `SELECT` privileges.
- If the `tblOwner` is not provided and the `tblName` is provided, then the current user is used.

See Also

ttAgingLRUConfig
ttAgingTableLRUConfig
ttAgingScheduleNow
Implementing an Aging Policy in Your Tables in the *Oracle TimesTen In-Memory Database Operations Guide*

ttPLSQLMemoryStats

This procedure returns result statistics about PL/SQL library cache performance and activity.

Required Privilege

This procedure requires no privilege.

Related Views

This procedure has this related view.

SYS.V$PLSQL_MEMORY_STATS

Syntax

```
ttPLSQLMemoryStats()
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

Parameters

`ttPLSQLMemoryStats` takes no parameters.

Result Set

`ttPLSQLMemoryStats` returns the results in the following columns:

Column	Type	Description
paramName	`TT_VARCHAR(30) NOT NULL`	The name of the result statistic returned in this row.
paramValue	`BINARY_FLOAT NOT NULL`	The value of the result statistic returned in this row.

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

```
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, 485.00000 >
< GetHits, 444.000000 >
< GetHitRatio, .9154639 >
< Pins, 260.00000 >
< PinHits, 178.000000 >
< PinHitRatio, .6846154 >
< Reloads, 4.000000 >
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

```
< Invalidations, 0.000000e+00 >
< CurrentConnectionMemory, 56.00000 >
9 rows found.
```

ttRamPolicyAutoReloadGet

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.

Related Views

This procedure has no related views.

Syntax

```
ttRamPolicyAutoReloadGet()
```

Parameters

`ttRamPolicyAutoReloadGet` has no parameters.

Result Set

`ttRamPolicyAutoReloadGet` returns the results:

Column	Type	Description
flag	`TT_INTEGER`	The policy used to determine if the database is reloaded into RAM after an invalidation. Valid values are: `0` - The database is not automatically reloaded into memory after an invalidation. This is the equivalent of the command `ttAdmin -noAutoReload`. `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.

Examples

To view the RAM autoreload policy, use:

```
CALL ttRamPolicyAutoReloadGet();
```

See Also

ttRamPolicyAutoReloadSet
ttAdmin

Copyright © 1996, 2026, Oracle and/or its affiliates.

ttRamPolicyAutoReloadSet

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.

Related Views

This procedure has no related views.

Syntax

`ttRamPolicyAutoReloadSet(`*`flag`*`)`

Parameters

`ttRamPolicyAutoReloadSet` has the parameters:

Parameter	Type	Description
`flag`	`TT_INTEGER NOT NULL`	The policy used to determine if the database is reloaded into RAM after an invalidation. Valid values are: `0` - The database is not automatically reloaded into memory after an invalidation. This is the equivalent of the command `ttAdmin -noAutoReload`. `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.

Result Set

`ttRamPolicyAutoReloadSet` returns no results.

Examples

To automatically reload a database into RAM after an invalidation, use:

```
CALL ttRamPolicyAutoReloadSet(1);
```

See Also

ttRamPolicyAutoReloadGet
ttAdmin

ttRamPolicyGet

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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Related Views

This procedure has no related views.

Syntax

```
ttRamPolicyGet()
```

Parameters

`ttRamPolicyGet` has no parameters.

Result Set

`ttRamPolicyGet` returns the results:

Column	Type	Description
ramPolicy	`TT_VARCHAR (10)`	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 ttAdmin `-ramLoad` 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.
ramGrace	`TT_INTEGER`	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`.

Parameters

`ttRamPolicyGet` has no parameters.

Examples

To view the RAM policy, use:

```
CALL ttRamPolicyGet();
```

See Also

ttRamPolicySet
ttAdmin
Specifying a RAM Policy in *Oracle TimesTen In-Memory Database Operations Guide*

Copyright © 1996, 2026, Oracle and/or its affiliates.

ttRamPolicySet

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.

Related Views

This procedure has no related views.

Syntax

```
ttRamPolicySet('ramPolicy', [ramGrace])
```

Parameters

`ttRamPolicySet` has the parameters:

Parameter	Type	Description
`ramPolicy`	`TT_VARCHAR (10) NOT NULL`	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 ttAdmin `-ramLoad` 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.
`ramGrace`	`TT_INTEGER`	Sets the number of seconds the database is kept in RAM after the last application has disconnected. This number is only effective if *`ramPoliy`* is `inUse`. This parameter is optional, and when omitted or set to `NULL`, the existing *`ramGrace`* period is left unchanged.

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:

```
CALL ttRamPolicySet('inUse', 10);
```

See Also

ttRamPolicyGet

Copyright © 1996, 2026, Oracle and/or its affiliates.

ttAdmin
Specifying a RAM Policy in *Oracle TimesTen In-Memory Database Operations Guide*

ttRedundantIndexCheck

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.

Related Views

This procedure has this related view.

SYS.V$REDUNDANT_INDEX

Syntax

```
ttRedundantIndexCheck('tblname')
```

Parameters

`ttRedundantIndexCheck` has the parameter:

Parameter	Type	Description
tblName	TT_CHAR(61)	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.

Result Set

`ttRedundantIndexCheck` returns the result:

Column	Type	Description
redundancy	TT_VARCHAR (1024) NOT NULL	The names of redundant indexes and a suggestion for which index to drop.

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.
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

```
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.
```

ttRekeyStatus

This procedure displays the currently running or last completed rekey operation. If a rekey operation is in progress, it displays information about the ongoing rekey operation.

Required Privilege

This procedure requires `ADMIN` privileges.

Related Views

This procedure has this related view.

SYS.V$REKEY_STATUS

Syntax

```
ttRekeyStatus()
```

Parameters

`ttRekeyStatus` has no parameters.

Result Set

`ttRekeyStatus` returns the result set:.

Column	Type	Description
`status`	`TT_VARCHAR (200) NOT INLINE NOT NULL`	Information about the progress or completion of the last rekey operation. Possible values include the following: • `Never rekeyed` • `In progress` • `Rekey done` • `Rekey failed`
`startTime`	`TT_TIMESTAMP`	Time when the rekey operation started.
`endTime`	`TT_TIMESTAMP`	Time when the rekey operation completed. If this column is `NULL` and the `startTime` column is not `NULL`, a rekey operation is in progress.
`rekeyCount`	`TT_INTEGER`	Count of how many rekey operations have completed.

Examples

The following example shows an example result set for a call to `ttRekeyStatus`.

```
Command> CALL ttRekeyStatus();
< Rekey done, 2025-09-11 22:19:39.044642, 2025-09-11 22:19:39.997542, 1 >
1 row found.
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

ttRepDeactivate

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.

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

This procedure returns the status of one or more replication peer databases.

Required Privilege

This procedure requires no privilege.

Related Views

This procedure has no related views.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Syntax

```
ttReplicationStatus(['subscriber'], ['hostname'])
```

Parameters

`ttReplicationStatus` has the optional parameters:

Parameter	Type	Description
`subscriber`	`TT_VARCHAR (200)`	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.
hostname	`TT_VARCHAR (200)`	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.

Result Set

`ttReplicationStatus` returns the result set:

Column	Type	Description
subscriber	`TT_VARCHAR(200) NOT NULL`	Subscriber name.
hostName	`TT_VARCHAR(200) NOT NULL`	Name of the system that hosts the subscriber.
port	`TT_INTEGER NOT NULL`	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.
pState	`TT_CHAR(10) NOT NULL`	Current replication state of the subscriber with respect to its master database. The values of the result column are: `start` - Replication is enabled to this peer. `pause` - Replication is temporarily paused to this peer. TimesTen preserves updates. See Set the Replication State of Subscribers in *Oracle TimesTen In-Memory Database Replication Guide* for more information. `stop` - Replication updates are `NOT` being collected for this peer. `failed` - Replication to a subscriber is considered failed because the threshold limit (log data) has been exceeded. This state is set by the system.
logs	`TT_INTEGER NOT NULL`	Number of transaction log files the master database is retaining for a subscriber.
lastMsg	`TT_INTEGER`	Seconds since last interaction or `NULL`.
replicationName	`TT_CHAR(30) NOT NULL`	Name of replication scheme.
replicationOwner	`TT_CHAR(30) NOT NULL`	Owner of replication scheme.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Examples

```
Command> call ttReplicationStatus();
< MASTER2, HOST1, 0, start     , 1, 257142, \
     _ACTIVESTANDBY         , TTREP       >
1 row found.

Command> 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

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.

Related Views

This procedure has no related views.

Syntax

```
ttRepPolicyGet()
```

Parameters

ttRepPolicyGet has no parameters.

Result Set

ttRepPolicyGet returns the results:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Column	Type	Description
repPolicy	TT_VARCHAR (10)	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.

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 Oracle TimesTen In-Memory Database C Developer's Guide*

ttRepPolicySet

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.

Related Views

This procedure has no related views.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Syntax

```
ttRepPolicySet('repPolicy')
```

Parameters

`ttRepPolicySet` has this parameter:

Parameter	Type	Description
repPolicy	`TT_VARCHAR (10) NOT NULL`	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.

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

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

Copyright © 1996, 2026, Oracle and/or its affiliates.

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.

Related Views

This procedure has no related views.

Syntax

```
ttRepQueryThresholdGet()
```

Parameters

`ttRepQueryThresholdGet` has no parameters.

Result Set

`ttRepQueryThresholdGet` returns the result:

Column	Type	Description
repQueryThreshold	TT_INTEGER	The number of seconds that a replication query executes before returning an error.

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

This procedure specifies the number of seconds that a query can be executed by the replication agent before TimesTen writes a warning to the daemon log. The specified value takes effect the next time the replication agent is started. The query threshold for the

Copyright © 1996, 2026, Oracle and/or its affiliates.

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.

Related Views

This procedure has no related views.

Syntax

```
ttRepQueryThresholdSet(seconds);
```

Parameters

`ttRepQueryThresholdSet` has the parameter:

Parameter	Type	Description
seconds	`TT_INTEGER NOT NULL`	Number of seconds a SQL statement can be run by the replication agent before TimesTen writes a warning to the daemon log. The value must be greater than or equal to 0. Default is `0` and indicates that TimesTen does not write any warnings.

Result Set

`ttRepQueryThresholdSet` returns no results.

Examples

To set the replication query threshold value to four seconds, use:

```
CALL ttRepQueryThresholdSet(4);
```

See Also

- ttRepDeactivate
- ttReplicationStatus
- ttRepPolicyGet
- ttRepQueryThresholdGet
- ttRepStart
- ttRepStop
- ttRepSubscriberStateSet
- ttRepSubscriberWait
- ttRepSyncGet
- ttRepSyncSet
- ttRepTransmitSet
- ttRepDuplicateEx in *Oracle TimesTen In-Memory Database C Developer's Guide*

Copyright © 1996, 2026, Oracle and/or its affiliates.

ttRepStart

This procedure starts the TimesTen replication agent for the connected database.

Required Privilege

This procedure requires the `CACHE_MANAGER` privilege.

Related Views

This procedure has no related views.

Syntax

```
ttRepStart()
```

Parameters

`ttRepStart` has no parameters.

Result Set

`ttRepStart` returns no results.

Examples

To start the replication agent, use:

```
CALL ttRepStart();
```

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 (LockLevel=1).

See Also

ttRepDeactivate
ttRepTransmitGet
ttRepTransmitSet
ttReplicationStatus
ttRepPolicySet
ttRepStop
ttRepSubscriberStateSet
ttRepSubscriberWait
ttRepSyncSet
ttRepSyncGet
ttRepDuplicateEx in *Oracle TimesTen In-Memory Database C Developer's Guide*

Copyright © 1996, 2026, Oracle and/or its affiliates.

ttRepStateGet

This procedure returns the current replication state of a database in an active standby pair.

Required Privilege

This procedure requires no privilege.

Related Views

This procedure has no related views.

Syntax

```
ttRepStateGet()
```

Parameters

`ttRepStateGet` has no parameters.

Result Set

`ttRepStateGet` returns the result:

Column	Type	Description
`state`	`TT_VARCHAR (20) NOT NULL`	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.

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

Copyright © 1996, 2026, Oracle and/or its affiliates.

ttReplicationStatus
ttRepPolicySet
ttRepStateSave
ttRepStateSet
ttRepStop
ttRepSubscriberStateSet
ttRepSubscriberWait
ttRepDuplicateEx in *Oracle TimesTen In-Memory Database C Developer's Guide*

ttRepStateSave

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.

Related Views

This procedure has no related views.

Syntax

```
ttRepStateSave('state', 'storeName', 'hostName')
```

Parameters

ttRepStateSave has these parameters:

Parameter	Type	Description
state	TT_VARCHAR (20) NOT NULL	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.
storeName	TT_VARCHAR (200) NOT NULL	Name of the database for which the state is indicated.
hostName	TT_VARCHAR (200)	Name of the host where the database resides.

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');
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

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

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 run in the following situations:

- A database has had a `CREATE ACTIVE STANDBY PAIR` command run 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.

Related Views

This procedure has no related views.

Syntax

```
ttRepStateSet('state')
```

Parameters

`ttRepStateSet` has the parameter:

Parameter	Type	Description
state	`TT_VARCHAR (20) NOT NULL`	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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

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

This procedure stops the TimesTen replication agent for the connected database.

Required Privilege

This procedure requires the `CACHE_MANAGER` privilege.

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();
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

> **Note**
>
> 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

This procedure changes a replicating subscriber's state with respect to the master store.

Required Privilege

This procedure requires the `ADMIN` privilege.

Related Views

This procedure has no related views.

Syntax

```
ttRepSubscriberStateSet('replicationName', 'replicationOwner',
 'subscriberStoreName', 'subscriberHostName', newStateCode)
```

Parameters

`ttRepSubscriberStateSet` has these parameters:

Parameter	Type	Description
replicationName	`TT_CHAR (30)`	The name of the replication scheme on which to operate. May be `NULL` to indicate all replication schemes.
replicationOwner	`TT_CHAR (30)`	The owner of the replication scheme. May be `NULL` to indicate all replication scheme owners.
subscriberStoreName	`TT_VARCHAR (200)`	The name of the subscribing database whose state is to be set. May be `NULL` to indicate all stores on host *subscriberHostName*.
subscriberHostName	`TT_VARCHAR (200)`	The subscriber's host. May be `NULL` to indicate all hosts of subscribing peers.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Parameter	Type	Description
newStateCode	TT_INTEGER	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 "failed.") Set the Replication State of Subscribers in the *Oracle TimesTen In-Memory Database Replication GuideOracle TimesTen In-Memory Database Replication Guide* for more information.

Result Set

ttRepSubscriberStateSet returns no results.

Examples

For the replication scheme named REPL.REPSCHEME, the following directs the master database to set the state of the subscriber database (SUBSCRIBERDS ON SYSTEM1) to Stop (2):

```
CALL ttRepSubscriberStateSet('REPSCHEME', '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

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

Copyright © 1996, 2026, Oracle and/or its affiliates.

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.

The procedure is working as 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.

Related Views

This procedure has no related views.

Syntax

```
ttRepSubscriberWait('replicationName', 'replicationOwner',
'subscriberStoreName', 'subscriberHostName', waitTime)
```

Parameters

ttRepSubscriberWait has these parameters:

Parameter	Type	Description
replicationName	TT_CHAR (30)	The name of the replication scheme on which to operate. May be NULL to indicate all replication schemes.
replicationOwner	TT_CHAR (30)	The owner of the replication scheme. May be NULL to indicate all replication scheme owners.
subscriberStoreName	TT_VARCHAR (200)	The name of the subscribing database whose state is to be set. May be NULL to indicate all stores on host *subscriberHostName*.
subscriberHostName	TT_VARCHAR(200)	The subscriber's host. May be NULL to indicate all hosts of subscribing peers.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Parameter	Type	Description
waitTime	TT_INTEGER NOT NULL	Number of seconds to wait for the specified subscriber(s). A value of -1 indicates to wait forever. This parameter is required and may not be NULL.

Result Set

ttRepSubscriberWait returns the result set:

Column	Type	Description
timeOut	BINARY(1)	0x00 - The wait succeeded within the allotted *waitTime*; the specified subscribers are up to date at the time this procedure was called. TimesTen returns 0x01 if not enough time has been granted.

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:

```
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

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.

Related Views

This procedure has no related views.

Syntax

```
ttRepSyncGet()
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

Parameters

`ttRepSyncGet` has no parameters.

Result Set

`ttRepSyncGet` returns the result set:

Column	Type	Description
`requestReturn`	`BINARY(1)`	`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").
`returnWait`	`TT_INTEGER`	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.
`localAction`	`TT_INTEGER`	The current `LOCAL ACTION` configuration for `RETURN` services. `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.

Examples

To retrieve the caller's *`requestReturn`* value, use:

```
SQLCHAR requestReturn[1];
SQLINTEGER    len;
rc = SQLExecDirect   ( hstmt
                , (SQLCHAR *) "{CALL ttRepSyncGet( NULL )}"
                , SQL_NTS )
rc = SQLBindCol     ( hstmt
                , /* ColumnNumber */      1
                , /* Tarype */      SQL_C_BINARY )
                , /* 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`*.

Copyright © 1996, 2026, Oracle and/or its affiliates.

- 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

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.

Related Views

This procedure has no related views.

Syntax

```
ttRepSyncSet([requestReturn], [returnWait], [localAction])
```

Parameters

`ttRepSyncSet` has these optional parameters:

Parameter	Type	Description
requestReturn	`BINARY(1)`	`0x00` - Turn off the return service for the current transaction. `0x01` - Turn on return services for the current transaction. Committing the transaction resets this attribute to its default value of `0` ("off"). You can use this parameter to turn on or turn off return services only when the replication subscribers have been configured with `RETURN RECEIPT BY REQUEST` or `RETURN TWOSAFE BY REQUEST`.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Parameter	Type	Description
returnWait	TT_INTEGER	Specifies the number of seconds to wait for return service acknowledgment. The default value is 10. A value of 0 means there is no wait time. This timeout value overrides the value set by the RETURN WAIT TIME attribute in the CREATE REPLICATION or ALTER REPLICATION statement. The timeout set by this parameter persists across transaction boundaries and applies to all return services independent of the BY REQUEST option.
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:

```
rc = SQLExecDirect ( hstmt,
 (SQLCHAR *)"{CALL ttRepSyncSet( 0x01 )}",
                   SQL_NTS )
```

> **Note**
>
> 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

Copyright © 1996, 2026, Oracle and/or its affiliates.

ttRepSubscriberWait
ttRepSyncGet
ttRepDuplicateEx in *Oracle TimesTen In-Memory Database C Developer's Guide*

ttRepSyncSubscriberStatus

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.

Related Views

This procedure has no related views.

Syntax

```
ttRepSyncSubscriberStatus('subscriber', 'hostName')
```

Parameters

`ttRepSyncSubscriberStatus` has these parameters:

Parameter	Type	Description
`subscriber`	`TT_VARCHAR (200) NOT NULL`	The name of the subscribing database to be queried.
`hostName`	`TT_VARCHAR (200)`	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.

Result Set

`ttRepSyncSubscriberStatus` returns:

Column	Type	Description
`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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

> **Note**
>
> 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

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.

Related Views

This procedure has no related views.

Syntax

```
ttRepTransmitGet()
```

Parameters

`ttRepTransmitGet` has no parameters.

Result Set

`ttRepTransmitGet` returns the result:

Column	Type	Description
transmit	TT_INTEGER	0 - Updates are not being transmitted to any subscribers for the remainder of the transaction on the connection. 1 (default) - Updates are being transmitted to subscribers on the connection.

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

Copyright © 1996, 2026, Oracle and/or its affiliates.

ttRepTransmitSet
ttRepDuplicateEx in *Oracle TimesTen In-Memory Database C Developer's Guide*

ttRepTransmitSet

This procedure stops subsequent updates on the connection it is run 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.

Related Views

This procedure has no related views.

Syntax

```
ttRepTransmitSet(transmit)
```

Parameters

`ttRepTransmitSet` has the parameter:

Parameter	Type	Description
transmit	`TT_INTEGER NOT NULL`	When set to 1, updates are transmitted to subscribers on the connection after the built-in is run. (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.

Result Set

`ttRepTransmitSet` returns no results.

Examples

To activate the active database in an active standby pair, use:

```
CALL ttRepTransmitSet(1);
```

To deactivate the active database in an active standby pair, use:

```
CALL ttRepTransmitSet(0);
```

See Also

- ttRepDeactivate
- ttReplicationStatus
- ttRepPolicySet
- ttRepStateSave
- ttRepStateSet
- ttRepStop

Copyright © 1996, 2026, Oracle and/or its affiliates.

ttRepSubscriberStateSet
ttRepSubscriberWait
ttRepTransmitGet
ttRepDuplicateEx in *Oracle TimesTen In-Memory Database C Developer's Guide*

ttRepXactStatus

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.

Related Views

This procedure has no related views.

Syntax

```
ttRepXactStatus(xactID)
```

Parameters

`ttRepXactStatus` has the parameter:

Parameter	Type	Description
xactID	VARBINARY (10000)	If no parameter is specified, status is returned for one of the following: • If called in a transaction that has begun, but not completed, commit processing, it returns the status of the transaction. • 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.

Result Set

`ttRepXactStatus` returns the result set:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Column	Type	Description
subscriberName	TT_CHAR (61)	The name of the database that subscribes to tables updated in the transaction. The name returns as: *store_name@host_name*.
state	TT_CHAR (2)	The state of the transaction with respect to the subscribing database. The return values are one of the following: 'NS' - Transaction not sent to the subscriber. 'RC' - Transaction received by the subscriber agent. 'CT' - Transaction applied at the subscriber store. (Does not convey whether the transaction ran into an error when being applied.) 'AP' - Transaction has been durably applied on the subscriber.
errorString	TT_VARCHAR (2000)	Error string retuned 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.

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

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 run in any subsequent transaction or in the same transaction after commit processing has begun for a transaction in RETURN TWOSAFE replication.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Required Privilege

This procedure requires no privilege.

Related Views

This procedure has no related views.

Syntax

```
ttRepXactTokenGet('typ')
```

Parameters

`ttRepXactTokenGet` has these parameters:

Parameter	Type	Description
typ	`TT_CHAR (2) NOT NULL`	The type of transaction desired: `'RR'` - Return receipt. `'R2'` - Return twosafe.

Result Set

`ttRepXactTokenGet` returns the result set:

Column	Type	Description
token	`VARBINARY (10000)`	A `VARBINARY` token used to represent the transaction desired.

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

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-

Copyright © 1996, 2026, Oracle and/or its affiliates.

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.

Related Views

This procedure has no related views.

Syntax

```
ttSetUserColumnID('tblName', 'colName', repID)
```

Parameters

`ttSetUserColumnID` has these 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

`ttSetUserColumnID` returns no results.

Examples

```
CALL ttSetUserColumnID('APP.SESSION', 'SESSIONID', 15);
```

See Also

ttSetUserTableID

ttSetUserTableID

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.

Related Views

This procedure has no related views.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Syntax

```
ttSetUserTableID('tblName', repID)
```

Parameters

`ttSetUserTableID` has these parameters:

Parameter	Type	Description
tblName	`TT_CHAR (61) NOT NULL`	Table name. Using a synonym to specify a table name is not supported.
repID	`BINARY(8) NOT NULL`	Integer identifier.

Result Set

`ttSetUserTableID` returns no results.

Examples

```
CALL ttSetUserTableID('APP.SESSION', 0x123456);
```

See Also

- ttSetUserColumnID

ttSize

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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Related Views

This procedure has no related views.

Syntax

```
ttSize(['tblName'], [nRows], frac)
```

Parameters

ttSize has these parameters:

Parameter	Type	Description
tblName	TT_CHAR(61)	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.
nRows	TT_BIGINT	Number of rows to estimate in a table. This parameter is optional.
frac	BINARY_DOUBLE	Estimated average fraction of VARBINARY or VARCHAR column sizes. This parameter is optional.

Result Set

ttSize returns the following result set.

Column	Type	Description
size	BINARY_DOUBLE NOT NULL	Estimated size of the table, in bytes.

Examples

```
CALL ttSize('ACCTS', 1000000, NULL);

CALL ttSize('ACCTS', 30000, 0.8);

CALL ttSize('SALES.FORECAST', NULL, NULL);
```

When using ttSize, you must first run the command and then fetch the results. For example:

ODBC

```
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

```
. . . . . .
String URL="jdbc:timesten:MyDataStore";
Connection con;
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

```
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();

. . . . . .
```

> **Note**
>
> The `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

ttComputeTabSizes

ttSQLCmdCacheInfo

This procedure returns information about all prepared SQL statements in the TimesTen SQL command cache.

Required Privilege

This procedure requires the `ADMIN` privilege.

Related Views

This procedure has this related view.

SYS.V$SQL_CMD_CACHE

Syntax

```
ttSQLCmdCacheInfo([sqlCmdID])
```

Parameters

`ttSQLCmdCacheInfo` has the optional parameter:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Parameter	Type	Description
sqlCmdID	TT_BIGINT	The unique identifier of a SQL command in the TimesTen command cache. If no value is supplied, information is displayed for all commands.

Result Set

ttSQLCmdCacheInfo returns the result set:

Column	Type	Description
sqlCmdID	TT_BIGINT NOT NULL	The unique identifier of a command.
privateCommandConnectionID	TT_INTEGER	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.
executions	TT_BIGINT NOT NULL	Counts the number of executions of the command.
prepares	TT_BIGINT NOT NULL	Counts the number of prepares for the command.
reprepares	TT_BIGINT NOT NULL	Counts the number of reprepares for the command.
freeable	TT_TINYINT NOT NULL	Indicates whether this command can be garbage collected by the subdaemon. 1 - Indicates freeable. 0 - Indicates non-freeable.
size	TT_INTEGER NOT NULL	The total space (bytes) allocated for this command in the command cache.
owner	TT_CHAR(31) NOT NULL	The user who created the command.
queryText	TT_VARCHAR (409600) NOT NULL	The full SQL text for the current command.
fetchCount	TT_BIGINT NOT NULL	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.
startTime	TT_TIMESTAMP	The time when the statement was last run. The value is in the form: *YYYY-MM-DD HH:MI:SS.FFF*
maxExecuteTime	NUMBER	The maximum wall clock execute time in seconds for this statement.
lastExecuteTime	NUMBER	Last measured execution time in seconds of the command.
minExecuteTime	NUMBER	If SqlCmdSampleFactor > 0, minimum execute time in seconds, otherwise 0.0.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Column	Type	Description
tempSpaceUsage	TT_BIGINT	The amount of temporary space in bytes used by the last execution of this command.
maxTempSpaceUsage	TT_BIGINT	Expresses the maximum amount of temporary space in bytes used by any previous run of this command.
writeConcurrencyModeCompiled	TT_TINYINT	The optimizer's write concurrency mode under which the command compiles.
writeConcurrencyModeSensitive	TT_TINYINT	The command compiles the same under all modes or compiles differently under different modes.

Examples

To display command information in `ttIsql` for all the current valid commands, use:

```
Command> call ttsqlcmdcacheinfo;
< 45085320, -1, 0, 1, 0, 0, 4464, SYS                              , update sys.user$ u
set u.lcount = u.lcount + 1 where u.user# = :userid, 0, <NULL>, 0, <NULL>, 0, 0, 0, 0, 0
>
< 43977560, -1, 0, 1, 0, 0, 4736, SYS                              , update sys.user$ u
set u.lcount = 0, u.astatus = :astatus where u.user# = :userid, 0, <NULL>, 0, <NULL>, 0,
0, 0, 0, 0 >
< 45094152, -1, 0, 1, 0, 0, 4640, SYS                              , select u.password,
u.password_date from sys.user_history$ u where u.user# = :userid order by
u.password_date desc, 0, <NULL>, 0, <NULL>, 0, 0, 0, 0, 0 >
< 43968568, -1, 0, 1, 0, 0, 4080, SYS                              , select pn.profile#
from sys.profname$ pn where pn.name = :profilename, 0, <NULL>, 0, <NULL>, 0, 0, 0, 0, 0 >
< 35824216, -1, 1, 1, 0, 1, 10504, SYS                              , select
o.owner#,o.obj#,u.name,o.name,o.namespace from sys.user$ u, sys.obj$ o where
u.user#=o.owner#   and o.type#=:1   and not exists   (select p_obj# from sys.dependency$
where p_obj# = o.obj#) order by o.obj# for update, 1, 2026-02-17 04:49:26.547000, 0, 0,
0, 5008, 5008, 0, 0 >
< 35017680, -1, 1, 1, 0, 0, 6849, JOHN                             , call
ttsqlcmdcacheinfo, 6, 2026-02-17 04:49:39.179000, 0, 0, 0, 0, 0, 0, 0 >
< 43972488, -1, 0, 1, 0, 0, 5256, SYS                              , select p.resource#,
p.limit# from sys.profile$ p where p.type# = 0 and p.profile# = :profileid, 0, <NULL>,
0, <NULL>, 0, 0, 0, 0, 0 >
< 35005120, -1, 6, 6, 0, 1, 7088, SYS                              , select
owner#,name,namespace,obj#,type#,ctime,mtime,stime,status,flags from sys.obj$ where
owner#=:1 and name=:2 and namespace=:3, 6, 2026-02-17 04:49:39.179000, 0, 0, 0, 1110,
1110, 0, 0 >
< 43959784, -1, 0, 1, 0, 0, 8744, SYS                              , select 1 from
sys.sysauth$ s where (s.grantee# = :userid or s.grantee# = 1) and (s.privilege# = :priv
or s.privilege# = 67), 0, <NULL>, 0, <NULL>, 0, 0, 0, 0, 0 >
< 45098584, -1, 0, 1, 0, 0, 4472, SYS                              , delete from
user_history$ where user# = :userid and password_date < :pdate, 0, <NULL>, 0, <NULL>, 0,
0, 0, 0, 0 >
< 45089352, -1, 0, 1, 0, 0, 5272, SYS                              , update sys.user$ u
set u.lcount = u.lcount + 1, u.ltime = sysdate, u.astatus = :astatus where u.user#
= :userid, 0, <NULL>, 0, <NULL>, 0, 0, 0, 0, 0 >
< 43953960, -1, 1, 1, 0, 0, 6328, SYS                              , select u.user#,
u.password, u.identification, u.astatus, u.lcount, u.ltime, u.profile# from sys.user$ u
where u.name = :name and u.type# = 1, 1, 2026-02-17 04:49:26.547000, 0, 0, 0, 1400,
1400, 0, 0 >
< 35012432, -1, 1, 1, 0, 1, 2385, JOHN                             , CALL ttOptSetFlag
('passthrough', 0), 0, 2026-02-17 04:49:26.547000, 0, 0, 0, 32, 32, 0, 0 >
< 35014848, -1, 1, 1, 0, 1, 2793, JOHN                             , CALL ttconfiguration
('TTGrid'), 2, 2026-02-17 04:49:26.547000, 0, 0, 0, 1148, 1148, 0, 0 >
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

```
< 35835080, -1, 0, 1, 0, 1, 8240, SYS                          , select null from
sys.obj$ where obj#=:1 and type#=:2 and obj# not in (select p_obj# from dependency$
where p_obj# = sys.obj$.obj#), 0, <NULL>, 0, <NULL>, 0, 0, 0, 0, 0 >
15 rows found.
```

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

ttSQLCmdCacheInfoGet

ttSQLCmdCacheInfoGet

This procedure displays information about the commands in the TimesTen SQL command cache.

Required Privilege

This procedure requires no privilege.

Related Views

This procedure has this related view.

SYS.V$SQL_CMD_CACHE_INFO

Syntax

```
ttSQLCmdCacheInfoGet()
```

Parameters

`ttSQLCmdCacheInfoGet` has no parameters.

Result Set

`ttSQLCmdCacheInfoGet` returns the result set:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Column	Type	Description
cmdCount	TT_INTEGER NOT NULL	Number of commands in the cache.
freeableCount	TT_INTEGER NOT NULL	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.
size	TT_BIGINT NOT NULL	The current total space allocated to store all the cached commands, in bytes.

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

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.

Related Views

This procedure has this related view.

SYS.V$SQL_CMD_QUERY_PLAN

Syntax

`ttSQLCmdQueryPlan([`*`sqlCmdID`*`])`

Parameters

`ttSQLCmdQueryPlan` has the optional parameter:

Parameter	Type	Description
sqlCmdID	TT_BIGINT	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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Result Set

`ttSQLCmdQueryPlan` returns the result set:

Column	Type	Description
sqlCmdID	TT_BIGINT NOT NULL	The unique identifier of a command in the TimesTen command cache.
queryText	TT_VARCHAR(409600)	The first 1024 characters of the SQL text for the current command.
step	TT_INTEGER	The step number of current operation in this run-time query plan.
level	TT_INTEGER	The level number of current operation in this run-time query plan.
operation	TT_CHAR(127)	The operation name of the current step in this run-time query plan.
tableName	TT_CHAR(31)	Name of the table used in this step, if any. Using a synonym to specify a table name is not supported.
tableOwnerName	TT_CHAR(31)	Name of the owner of the table used in this step, if any.
indexName	TT_CHAR(31)	Name of the index used in this step, if any.
indexedPred	TTVARCHAR(1024)	In this step, if an index is used, the indexed predicate is printed if available. Not all expressions can be printed out and the output may be fragmented and truncated. "..." represents the unfinished portion of the expression.
nonIndexedPred	TT_VARCHAR(1024)	In this step, if a non-indexed predicate is used, the non-indexed predicate is printed if available. Not all expressions can be printed out and the output may be fragmented and truncated. "..." represents the unfinished portion of the expression.
miscellaneous	TT_VARCHAR (65536)	The type of constraint and other information about the constraint. Constraint type can be one of: `ForeignKeyInsert` - To insert foreign key. `ForeignKeyDelete` - To delete foreign key. `UniqueKeyInsert` - To insert unique key. `ForeignKeyOrphanChild` - To handle case where parent is lost in case of foreign key constraint. `ForeignKeyCascadeDelete` - To delete corresponding row for cascade delete. `ForeignKeySyncCascadeDelete` - To delete corresponding row for cascade delete from sync replica.

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> >
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

```
< 528078576, <NULL>, 0, 4, RowLkSerialScan , T1 , SAMPLEUSER , , , >
< 528078576, <NULL>, 1, 7, RowLkRangeScan , T2 , SAMPLEUSER , I2 , , >
< 528078576, <NULL>, 2, 7, RowLkRangeScan , T5 , SAMPLEUSER , I2 , , >
< 528078576, <NULL>, 3, 6, NestedLoop , , , , , >
< 528078576, <NULL>, 4, 6, RowLkRangeScan , T3 , SAMPLEUSER , 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 , SAMPLEUSER , 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
 , SAMPLEUSER  , I2 , , NOT(LIKE( tuf ,abc ,NULL )) >
< 528079360, <NULL>, 2, 3, RowLkRangeScan  , T3  , SAMPLEUSER  ,
I2  , , >
< 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  ,  ,  ,  , , >
< 528054656, create table t2(x2 int,y2 int, z2 int), <NULL>,
<NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL> >
< 528066648, insert into t2 select * from t1, <NULL>, <NULL>,
<NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL> >
< 528066648, <NULL>, 0, 0, Insert  , T2  , SAMPLEUSER  ,  , , >
< 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 */
```

ttSQLExecutionTimeHistogram

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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Related Views

This procedure has this related view.

SYS.V$EXECUTION_TIME_HISTOGRAM

Syntax

```
ttSQLExecutionTimeHistogram(sqlCmdID)
```

Parameters

`ttSQLExecutionTimeHistogram` has the optional parameter:

Parameter	Type	Description
sqlCmdID	TT_BIGINT	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.

Result Set

`ttSQLExecutionTimeHistogram` returns the result set:

Column	Type	Description
histogramSamples	TT_BIGINT	The number of SQL command execution time operations have been measured since either the database was started or the ttStatsConfig built-in procedure was used to reset the statistics.
totalExecuteTime	NUMBER	The accumulated wall clock execution time when sampling in seconds.
bucketUpperBound	NUMBER	The upper limit in seconds of execution time.
count	TT_BIGINT	The number of SQL commands with time less than or equal to ExecutionTimeLimit and greater than ExecutionTimeLimit from the previous row or 0.

Examples

The following example shows the output for the `ttSQLExecutionTimeHistogram` built-in procedure:

The following example of the `ttSQLExecutionTimeHistogram` built-in procedure shows that a total of 1919 statements run. The total time for all 1919 statements to run was 1.090751 seconds. This example shows that SQL statements ran in the following time frames:

- 278 statements run in a time frame that was less than or equal to 0.00001562 seconds.
- 1484 statements run in a time frame that was greater than 0.00001562 seconds and less than or equal to 0.000125 seconds.
- 35 statements run in a time frame that was greater than 0.000125 seconds and less than or equal to 0.001 seconds.
- 62 statements run in a time frame that was greater than 0.001 seconds and less than or equal to 0.008 seconds.
- 60 statements run in a time frame that was greater than 0.008 seconds and less than or equal to 0.064 seconds.

Copyright © 1996, 2026, Oracle and/or its affiliates.

```
Command> call ttSQLExecutionTimeHistogram;
< 1919, 1.090751, .00001562, 278 >
< 1919, 1.090751, .000125, 1484 >
< 1919, 1.090751, .001, 35 >
< 1919, 1.090751, .008, 62 >
< 1919, 1.090751, .064, 60 >
< 1919, 1.090751, .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

ttStatsConfig

ttStatsConfig

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.

Related Views

This procedure has no related views.

Syntax

```
ttStatsConfig('param', [value], [option])
```

Parameters

`ttStatsConfig` has the parameters:

Parameter	Type	Description
param	`VARCHAR2(50) NOT NULL`	The name of the parameter to configure.
value	`VARCHAR2(200)`	The value of the specified parameter. If no value is supplied, the built-in procedure displays the current value for the specified parameter.
option	`VARCHAR2 (200)`	The value of a second optional parameter, for example `con_id` for `LatchStats`.

Parameter / Value Pairs

These parameter/value pairs can be set with TimesTen:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Parameter	Value	Description
ConnSampleFactor	C,S 0<=C<=Connections 0<=S<=60000000	Specifies the frequency at which a specific connection is sampled. C – Stands for the connection ID to be sampled. For connection ID, see ttXactIdGet. Connections – Represents the total number of connections to the database. S – Is the sampling factor. The default value is 0 and indicates that sampling is turned off. A value greater than 0 indicates that a sample is taken at that interval of statements for the connection specified.
LatchStats	scope,level scope=conn|db|*con_id* level=NONE|TYPICAL|ALL|BASIC	Specifies the scope, *scope*, and the level, *level*, for collection for latch statistics. The scope value determines at what level TimesTen should collect latch statistics: • conn - Collects latch statistics for your current connection. • db - Collects latch statistics for your database. • con_id - Collects latch statistics for the connection id that you specify. The level value determines the level at which TimesTen collects statistics: • NONE - Disables the collection of latch statistics. • TYPICAL - Ensures the collection of major useful latch statistics. • ALL - Additional statistics are added to the set of statistics collected with the TYPICAL setting. The additional statistics include internal and debugging statistics. • BASIC - Disables the collection of many of the important latch statistics. If latch statistics are enabled, TimesTen allocates around 100KB from temporary memory to store these statistics. Once you have configured the LatchStats parameter, you can use the ttLatchStatsGet built-in procedure to view latch statistics. See ttLatchStatsGet for more information.
SQLCmdHistogramReset	0 or not	The existing SQL run time statistics are reset if the specified value is nonzero.
SQLCmdSampleFactor	0 <= *value* <= 60000000	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 run, the wall clock time of that run is captured.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Parameter	Value	Description
StatsLevel	NONE TYPICAL ALL BASIC	Specifies the level of collection for database statistics. TimesTen stores theses statistics in system tables. Setting the StatsLevel parameter to NONE disables the collection of system statistics. The default setting of TYPICAL ensures collection of major useful statistics and should be adequate for most environments. When the StatsLevel parameter is set to ALL, additional statistics are added to the set of statistics collected with the TYPICAL setting. The additional statistics include internal and debugging statistics. Setting the StatsLevel parameter to BASIC disables the collection of many of the important statistics required by many TimesTen features.

Note

There are two ttStatsConfig calls that configure the collection of histogram data for SQL commands.

- ttStatsConfig('SQLCmdSampleFactor', S) enables SQL histogram collection for all connections and the frequency at which a SQL command sample is taken. The value of the sampling factor S should be greater than 0.
- ttStatsConfig('ConnSampleFactor', 'C, S') enables SQL histogram collection for a connection to the database (C) and the frequency at which a specific connection is sampled. The value of the sampling factor S should be greater than 0.

If the sampling factor value is greater than 0 for both calls, then the value for ttStatsConfig('ConnSampleFactor', 'C, S') takes precedence. If the sampling factor value is set to 0 for both parameters, then the sampling is turned off.

Result Set

ttStatsConfig returns the result set:

Column	Type	Description
param	VARCHAR2(50) NOT NULL	The name of the parameter that was configured.
value	VARCHAR2(200)	The value of the specified parameter. If no value is supplied, the built-in procedure displays the current value for the specified parameter.

Examples

These are supported examples:

Sample every command:

Copyright © 1996, 2026, Oracle and/or its affiliates.

```
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.
```

See Also

- ttStats
- ttStatsConfigGet

ttStatsConfigGet

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 no privilege.

Related Views

This procedure has this related view.

SYS.V$STATS_CONFIG

Syntax

```
ttStatsConfigGet()
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

Parameters

`ttStatsConfigGet` has no parameters:

Result Set

`ttStatsConfigGet` returns the result set:

Column	Type	Description
param	VARCHAR2(50)	The name of the parameter.
value	VARCHAR2(200)	The current value of the parameter.

Parameter / Value Pairs

These are the return parameter/value pairs in TimesTen:

These parameter/value pairs can be returned in the result set in TimesTen:

Parameter	Description
SQLCmdSampleFactor	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 run, the wall clock time of that run is captured.
ConnSampleFactor	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.
StatsLevel	The existing SQL run time statistics are reset if the specified value is nonzero.

Examples

These are the supported examples:

View the configuration settings of `ttStatsConfig`:

```
Command> call ttStatsConfigGet();
< SQLCMDSAMPLEFACTOR, 1 >
< CONNSAMPLEFACTOR, 2047,0 >
< STATSLEVEL, TYPICAL >
3 rows found.
```

See Also

- ttStats
- ttStatsConfig

Copyright © 1996, 2026, Oracle and/or its affiliates.

ttTableSchemaFromOraQueryGet

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 run. 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 Without Cache in the *Oracle TimesTen In-Memory Database Operations GuideOracle TimesTen In-Memory Database Operations Guide*.

Required Privilege

This procedure requires no privileges. The session user must have all required privileges to run the query on the Oracle database.

Related Views

This procedure has no related views.

Syntax

```
ttTableSchemaFromOraQueryGet(['tblOwner'], 'tblName', 'selectSQL')
```

Parameters

ttTableSchemaFromOraQueryGet has the parameters:

Parameter	Type	Description
tblOwner	TT_CHAR (30)	TimesTen table owner (optional). If not provided, the connection ID is used.
tblName	TT_CHAR (30) NOT NULL	Table name for the CREATE TABLE 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.
selectSQL	TT_VARCHAR (409600) NOT NULL	A SELECT query on an Oracle database to derive the table column definition. Any expressions in the SELECT list should be provided with a column alias; otherwise, an implementation dependent column name is assumed and the expression is not evaluated.

Result Set

ttTableSchemaFromOraQueryGet returns the result set:

Column	Type	Description
createSQL	TT_VARCHAR (409600) NOT NULL	A CREATE TABLE statement that matches the result set of the SELECT query on an Oracle database.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Examples

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

The `ttVersion` built-in procedure returns the five parts of the TimesTen release number.

Required Privilege

This procedure requires no privilege.

Related Views

This procedure has this related view.

SYS.V$VERSION

Syntax

```
ttVersion()
```

Parameters

`ttVersion` has no parameters.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Result Set

`ttVersion` returns the result set:

Column	Type	Description
`major`	`TT_INTEGER NOT NULL`	The first part of the five-part release number (22 for release 26.1.1.1.0), indicating the last two digits of the year of the major release. A change in `major1` indicates major infrastructure and functionality changes.
`minor`	`TT_INTEGER NOT NULL`	The second part of the five-part release number (1 for release 26.1.1.1.0). A change in only `major2` indicates a version with new functionality and possibly some infrastructure changes. Releases with the same `major1.major2` are patch-compatible. Data can be unloaded from a database from one release and loaded into a database from the other without the migration process.
`patchset`	`TT_INTEGER NOT NULL`	The third part of the five-part release number (1 for release 26.1.1.1.0). A change in only `patchset` indicates a release that contains bug fixes and possibly some feature enhancements.
`patch`	`TT_INTEGER NOT NULL`	The fourth part of the five-part release number (9 for release 26.1.1.1.0). A change in only `patch` indicates a release with only critical bug fixes.
`reserved`	`TT_INTEGER NOT NULL`	The fifth part of the five-part release number (0 for release 26.1.1.1.0). Reserved for future use.

Examples

Return for release 26.1.1.1.0:

```
Command>  call ttVersion();
< 22, 1, 1, 9, 0 >
1 row found.
```

ttWarnOnLowMemory

This procedure enables applications to specify that operations run 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.

Related Views

This procedure has no related views.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Syntax

```
ttWarnOnLowMemory(permanent, temporary)
```

Parameters

`ttWarnOnLowMemory` has these parameters:

Parameter	Type	Description
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 temporary data partition `0` - Disable warnings for the temporary data partition

Result Set

`ttWarnOnLowMemory` returns no results.

Examples

```
CALL ttWarnOnLowMemory(1, 0);
```

Enables low memory warnings for the permanent data partition only.

> **Note**
>
> 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

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. The first column represents the connection ID and the second column represents the increasing number of transactions for the same transaction ID. 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.

Related Views

This procedure has this related view.

Copyright © 1996, 2026, Oracle and/or its affiliates.

SYS.V$XACT_ID

Syntax

```
ttXactIdGet()
```

Parameters

`ttXactIdGet` has no parameters.

Result Set

`ttXactIdGet` returns the result set:

Column	Type	Description
xactID	TT_INTEGER	Connection ID.
counter	TT_BIGINT	An increasing number that distinguish successive transactions of the same transaction ID.

Examples

```
Command> autocommit 0;
Command > call ttXactIdGet();
<2,11>
1 row found.
Command> commit;
Command> call ttXactIdGet();
<3, 12>
1 row found.
```

> **Note**
>
> The output correlates to the values printed in lock error messages and ttXactAdmin lock information output.

See Also

ttXactAdmin
ttXactIdRollback in *Oracle TimesTen In-Memory Database C Developer's Guide*

ttXlaBookmarkCreate

This procedure creates the specified bookmark.

Required Privilege

This procedure requires the `XLA` privilege.

Related Views

This procedure has no related views.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Syntax

```
ttXlaBookmarkCreate('bookmark', 'replicated')
```

Parameters

`ttXlaBookmarkCreate` has the parameters:

Parameter	Type	Description
bookmark	`TT_CHAR (31) NOT NULL`	The name of the bookmark to be created.
replicated	`BINARY(1)`	`0x00` or `NULL` (equivalent) for non-replicated bookmarks (default setting). `0x01` for replicated bookmarks. If `NULL`, non-replicated bookmarks are used.

Result Set

`ttXlaBookmarkCreate` returns no results.

Examples

For non-replicated bookmark, execute the following:

```
Command > call ttXlaBookmarkCreate('mybookmark');
```

or:

```
Command> call ttxlabookmarkcreate('mybkmk2',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* .

> **Note**
>
> 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

Copyright © 1996, 2026, Oracle and/or its affiliates.

ttXlaBookmarkDelete

This procedure deletes the specified bookmark. The bookmark cannot be deleted while it is in use.

Required Privilege

This procedure requires the XLA privilege.

Related Views

This procedure has no related views.

Syntax

```
ttXlaBookmarkDelete('bookmark')
```

Parameters

ttXlaBookmarkDelete has the parameter:

Parameter	Type	Description
bookmark	TT_CHAR (31) NOT NULL	The name of the bookmark to be deleted.

Result Set

ttXlaBookmarkDelete returns no results.

Examples

```
Command > call ttXlaBookmarkDelete('mybookmark');
```

> **Note**
>
> 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

This procedure configures persistent XLA tracking of a table. This procedure cannot be run when the specified bookmark is in use.

Required Privilege

This procedure requires the XLA privilege.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Related Views

This procedure has no related views.

Syntax

```
ttXlaSubscribe('tblName', 'bookmark')
```

Parameters

`ttXlaSubscribe` has the parameters:

Parameter	Type	Description
tblName	`TT_CHAR (61) NOT NULL`	The name of the table to be tracked. Using a synonym to specify a table name is not supported.
bookmark	`TT_CHAR (31) NOT NULL`	The name of the bookmark that the application uses to track this table.

Result Set

`ttXlaSubscribe` returns no results.

Examples

```
Command > call ttXlaSubscribe ('SALLY.ACCTS', 'mybookmark');
```

> **Note**
>
> 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

This procedure stops persistent XLA tracking of a table. This procedure cannot be run when the specified bookmark is in use.

Required Privilege

This procedure requires the `XLA` privilege.

Related Views

This procedure has no related views.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Syntax

```
ttXlaUnsubscribe('tblName', 'bookmark')
```

Parameters

`ttXlaUnsubscribe` has the parameters:

Parameter	Type	Description
tblName	`TT_CHAR (61) NOT NULL`	The name of the table on which XLA tracking should be stopped. Using a synonym to specify a table name is not supported.
bookmark	`TT_CHAR (31) NOT NULL`	The name of the bookmark that the application uses to track this table.

Result Set

`ttXlaSubscribe` returns no results.

Examples

```
Command > call ttXlaUnsubscribe ('SALLY.ACCTS', 'mybookmark');
```

> **Note**
>
> 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

Copyright © 1996, 2026, Oracle and/or its affiliates.

4

Utilities

The reference information for most TimesTen utilities is included in this chapter, beginning with the following introductory sections:

- Overview
- Utilities List

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.

Utilities List

Utilities are described in this chapter.

Utilities Descriptions

Name	Description
ttAdmin	Specifies or changes database policies.
ttAdoptStores	Moves databases from a TimesTen installation to a new TimesTen installation of the same major release, but a different minor release.
ttBackup	Creates a backup copy of a database that can be restored at a later time using the `ttRestore` utility.
ttBulkCp	Copies data between TimesTen tables and ASCII files.
ttCapture	Captures information about the state of TimesTen.
ttCheck	Performs internal consistency checking within a TimesTen database.
ttClientImport	Imports the client wallet and `sys.odbc.in` configuration on the client.
ttCreateCerts	Manages certificates and wallets.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Name	Description
ttCacheInfo	Returns change-log table information for Oracle Database tables and information about Oracle Database objects used to track DDL statements issued on cached Oracle Database tables.
ttCWAdmin	Manages TimesTen active standby pairs that take advantage of the high availability framework of Oracle Clusterware.
ttDaemonAdmin	Starts and stops the TimesTen main daemon and Server.
ttDaemonLog	Controls and displays daemon log messages.
ttDestroy	Destroys a database including all checkpoint files, transaction logs and daemon catalog entries.
ttExporter	Starts or configures the TimesTen Prometheus exporter.
ttInstallationCheck	Examines all files in an installation of TimesTen and generates a signature for the installation,
ttInstallDSN	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.
ttInstanceCreate	Create a new TimesTen instance.
ttInstanceDestroy	Destroys an existing TimesTen instance.
ttInstanceModify	Modifies certain attributes of an instance.
ttIsql	Executes SQL statements interactively.
ttMigrate	Saves and restores TimesTen objects.
ttRepAdmin	Displays, sets, modifies and monitors existing replication definitions and status.
ttRestore	Creates a database from a backup that has been created using the `ttBackup` utility.
ttSchema	Prints out the schema, or selected objects, of a database.
ttShmSize	Returns the required total shared memory size needed for both the database and the PL/SQL shared memory segments.
ttSize	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.
ttStats	Monitors database metrics or takes and compares snapshots of metrics.
ttStatus	Displays information that describes the current state of TimesTen.
ttTail	Fetches TimesTen internal trace information from a database and displays it to `stdout`.
ttTraceMon	Enables and disables the TimesTen internal tracing facilities.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Name	Description
ttUser	Helps to securely store user IDs and passwords in an Oracle Wallet and to obtain a hashed password for the PWDCrypt connection attribute.
ttVersion	Lists the TimesTen release information.
ttXactAdmin	Lists ownership, status, log and lock information for each outstanding transaction.
ttXactLog	Displays a formatted dump of the contents of a TimesTen transaction log.

ttAdmin

This utility specifies or changes database policies. It enables you to:

- Get information about ttAdmin options, version, and settings. See Help, Version, and Query Options.
- Specify settings for database RAM policy, database start, and database stop. See Perform RAM Operations.
- Open or close a database to user connections. See Open or Close a Database.
- Rotate the encryption keys used for encryption at rest. See Encryption at Rest.
- Run a forced disconnect operation for existing database connections. See Force Disconnect.
- 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.
- Specify settings to automatically or manually start and stop replication agents for specified databases. See Set Replication Policies.

Required Privilege

This utility requires no privileges to query the database.

Cache options require the CACHE_MANAGER privilege.

All other options require the ADMIN privilege.

Syntax

```
ttAdmin {-h | -help | -?}
ttAdmin {-V | -version}
ttAdmin -query {-connStr connection_string | DSN}

ttAdmin  [-ramUnload | -ramLoad [-open | -close]]
         [-ramPolicy {always | manual | enduring | inUse [-ramGrace secs]}]
         [-autoreload | -noautoreload]
         [-shmAttach | -shmDetach [-ckpt | -noCkpt] | -shmFree]

         [-disconnect urgency [granularity]] {-connStr connection_string | DSN}]
           urgency: {-transactional | -immediate | -abort}
           granularity: [-users | -unload]
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

```
ttAdmin -clientExportAll {FILE}{-connStr connection_string | DSN}

ttAdmin -certificateList [-json]

ttAdmin -rekey [-encrAlg algorithm] {-connStr connection_string | DSN}

ttAdmin  [-repPolicy always | manual | norestart]
         [-repStart | -repStop]
         [-repQueryThresholdSet secs]
         [-repQueryThresholdGet]

           [-cacheUidGet]
           [-cacheUidPwdSet -cacheUid uid [-cachePwd pwd]]
           [-cachePolicy {always | manual | norestart}]
           [-cacheStart]
           [-cacheStop [-stopTimeout secs]]
           {-connStr connection_string | DSN}
```

Notes

The following notes apply to all modes of `ttAdmin` usage:

You need to set some environment variables. See Environment Variables in *Oracle TimesTen In-Memory Database Installation, Migration, and Upgrade Guide* for more details.
The `ttAdmin` utility is supported only for TimesTen Data Manager DSNs. It is not supported for TimesTen Client DSNs.

Help, Version, and Query Options

Options

`ttAdmin` has these options for help, version, and settings information:

Option	Description
`-h -help` `-?`	Prints usage information, syntax, and option descriptions.
`-V | -version`	Prints the TimesTen release number of `ttAdmin` and exits.
`-query`	Displays a summary of the policy settings for the specified database.
`-connStr` *`connection_string`*	For `-query`, an ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.
`DSN`	For `-query`, an ODBC data source name of the database to be administered.

Examples

To get the version of `ttAdmin`:

```
% ttAdmin -version
TimesTen Release 26.1.1.1.0
```

To get help for `ttAdmin`:

```
% ttAdmin -help
Usage:
  ttAdmin [-h | -help | -?]
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

```
ttAdmin [-V | -version]
ttAdmin [-ramUnload]
        [-shmAttach]
        [-shmFree]
        [-shmDetach [-ckpt | -nockpt]]
        [-ramLoad] [-open | -close]
        [-ramPolicy always | manual | enduring | inUse [-ramGrace seconds]]
        [-autoreload | -noautoreload]
        [-certificateList [-json]]
        [-clientExportAll {file}]
        [-disconnect { -transactional | -immediate | -abort }
          [-users | -unload]
        ]
        [-repPolicy always | manual | norestart]
        [-repStart | -repStop]
        [-repQueryThresholdSet seconds]
        [-repQueryThresholdGet]
        [
          [-cacheUidGet] |
          [-cacheUidPwdSet -cacheUid UID [-cachePwd password]] |
          [-cachePolicy always | manual | norestart] |
          [-cacheStart] |
          [-cacheStop [-stopTimeout seconds]]
        ]
        [-reKey [-encrAlg algorithm]]
        [-query]
        {DSN | [-connstr] connection_string}

options:
[...Options descriptions not shown...]
```

Perform RAM Operations

Options

`ttAdmin` has these options to set the RAM policy, start the database and stop the database:

Option	Description
`-connStr` *`connection_string`*	An ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.
`DSN`	An ODBC data source name of the database to be administered.
`-autoreload | -noautoreload`	If set to `-autoreload` (default), TimesTen reloads the database after an invalidation. If set to `-noautoreload`, TimesTen does not automatically reload the database after an invalidation.
`-ramGrace` *`secs`*	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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
`-ramLoad`	Valid only when `-ramPolicy` is `manual` or `enduring`. Causes the database to be loaded into RAM. If there is an existing database shared memory segment `-ramLoad` destroys it before creating the new one. This enables `-ramLoad` to clean up if an error is returned by `ttAdmin -shmAttach`. The shared memory segment must be free first so that `-ramLoad` can destroy it.
`-ramPolicy` *`policy`*	Defines the policy used to determine when the database is loaded into system RAM. `manual` - Specifies that the database is only to be loaded in system RAM when explicitly loaded by the user with the `-ramLoad` option. Using `-ramUnload` with `-ramPolicy manual` to unload the database, causes the daemon to destroy any remanent detached shared memory segment from a previous in memory database. This is the recommended RAM policy because it avoids unnecessary database loading or unloading. `enduring` - This RAM policy is like the RAM policy *manual*, except that it also enables you to use the `-shmDetach` and `-shmAttach` options to respectively stop and restart the database without destroying the shared memory segment that contains the database. This feature enables you to perform a fast patch upgrade, as you can detach from the shared memory segment instead of unloading the database. Then after the upgrade, you can attach to this shared memory segment avoiding loading the database. The RAM policy *enduring* also enables you to use the `-shmFree` option. `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. `always` - Specifies that the database should remain in system RAM all the time.
`-ramUnload`	Performs a *clean shutdown* of the database by disconnecting the subdaemon and destroying the shared memory segment. The checkpoint file written to disk allows you to reload the database using the `ttAdmin -ramLoad` option. The `-ramUnload` option is valid only with `-ramPolicy manual` or `-ramPolicy enduring`.
`-shmAttach`	Enables you to reattach to a preserved shared memory segment while starting the database. The prerequisites are setting the RAM policy to *enduring* and having the shared memory segment previously detached with the `-shmDetach` option.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
`-shmDetach [-ckpt | -noCkpt]`	Enables you to *stop* the database by detaching (instead of destroying) the shared memory segment. Before disconnecting from the shared memory segment, you can specify the subdaemon to perform a static checkpoint with the `–ckpt` option (which is the default) or to not perform a static checkpoint with the `–noCkpt` option. To use `-shmDetach`, the database must be closed and there must be no connections to the database.
`-shmFree`	The `-shmFree` option destroys a preserved shared memory segment that remained in memory after a `ttAdmin -shmDetach` operation. If you use the `-shmDetach` option but decide you do not want to re-attach the segment, you can free the segment by using the `-shmFree` option. After doing so, you can still load the database from the checkpoint files using the `-ramLoad` option.

Note

RAM policy *enduring* is helpful to perform fast patch upgrades, because you do not need to unload and reload the data which can be time-consuming. The `-ramPolicy enduring` enables you:

1. Use `-shmDetach` to preserve the shared memory segment while *stopping* the database.
2. Perform the fast upgrade.
3. Use `-shmAttach` to attach to the preserved memory segment while *starting* the database.

For more details, see Specifying a RAM Policy and Detaching, Attaching, and Freeing the Shared Memory Segment in the *Oracle TimesTen In-Memory Database Operations Guide*. For more information About Performing a Fast Patch Upgrade see the *Oracle TimesTen In-Memory Database Installation, Migration, and Upgrade Guide*.

Examples

To manually control whether `database1` is loaded into RAM and to load it now, use the following:

```
% ttAdmin -ramPolicy manual -ramLoad 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 set the RAM policy of `database1` to enduring:

```
$ ttAdmin -ramPolicy enduring database1
RAM Residence Policy            : enduring
Manually Loaded In RAM          : False
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

```
Replication Agent Policy        : manual
Replication Manually Started    : False
Cache Agent Policy              : manual
Cache Agent Manually Started    : False
Database state                  : open
```

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
```

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
```

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
Database State                  : Open
```

Notes

If you need to destroy a database, use the `ttDestroy` utility. If the database has a preserved shared memory segment (because it was detached with the `-shmDetach` utility) you must use the `ttDestroy -force` option. For more information on destroying a database, see ttDestroy.

Setting the RAM policy to *inUse* for production systems with large databases is not recommended as the database may unload or reload unexpectedly which can result in long delays and excessive overhead.

RAM policy `always` conflicts with forced disconnect granularity `unload`, described in Force Disconnect. Using both simultaneously results in an error and the disconnect request being ignored.

The *always* RAM policy should be used with caution. When failures occur, it may not be beneficial to have your database automatically reload. In addition, it may affect system startup performance if all databases load at the same time when your system boots.

If the shared memory segment is destroyed manually (for example by using `ipcrm`), `-shmFree` returns an error indicating that the shared memory segment is missing; however, the `-shmFree` command updates the database's DBI file and main daemon bookkeeping. The database can then be successfully loaded with `-ramLoad`, and the database RAM policy can be changed to another policy besides *enduring* by using `ttAdmin -ramPolicy`.

If you try to load a database using `-ramLoad` while a previously detached shared memory segment exists for that database, then the command returns an error and fails. You must either attach the segment with `-shmAttach` (instead of `-ramLoad`) or destroy it using `-shmFree`, and then reload the database using the `-ramLoad` option.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Open or Close a Database

Options

`ttAdmin` has these options for opening or closing a database:

Option	Description
`-close`	Closes a database to user connections. When a database is closed to user connections, new connection attempts will fail, but existing connections are unaffected.
`-connStr` *`connection_string`*	An ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.
`DSN`	An ODBC data source name of the database to be administered.
`-open`	Opens a database to user connections. A database is open to user connections by default upon creation.
`-ramLoad`	Valid only when `-ramPolicy` is `manual`. Causes the database to be loaded into RAM.

Examples

To open the `database1` DSN:

```
% ttAdmin -open database1
```

To load and open the `database1` DSN from an unloaded and closed state:

```
% ttAdmin -ramLoad -open database1
```

To close the `database1` DSN:

```
% ttAdmin -close database1
```

To load and close the `database1` DSN from an unloaded and open state:

```
% ttAdmin -ramLoad -close database1
```

Notes

A database remains closed or open to user connections regardless of its loaded state or RAM policy, unless its closed or open state is modified through `ttAdmin` or other utilities like `ttRestore` or `ttRepAdmin`.

If the `-open` or `-close` options are used in conjunction with any option other than `-ramLoad`, `ttAdmin` returns an error.

Trying to close a closed database or open an open database returns an error.

Transport Layer Security

Options

`ttAdmin` has these options for exporting access information and certificates:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
`-clientExportAll FILE`	Outputs a ZIP file containing the client wallet, a `sys.odbc.ini` file that can be used in accessing the database, and other files (such as `tnsnames.ora` file) as applicable. See (Optional) Task 6: Import Certificates and Configuration in TimesTen in *Oracle TimesTen In-Memory Database Security Guide* for more information. Also see ttClientImport.
`-certificateList [-json]`	Lists certificates that have been created on the server. See *Listing Certificates in TimesTen Classic* in *Oracle TimesTen In-Memory Database Security Guide* for more information.

Encryption at Rest

Options

`ttAdmin` has these options for encryption at rest of checkpoint and transaction log files:

Option	Description
`-connStr` *`connection_string`*	An ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.
`DSN`	An ODBC data source name of the database to be administered.
`-rekey [-encrAlg` *`algorithm`*`]`	On an encrypted database, rotates the master and data encryption keys used to encrypt the checkpoint and transaction log files. Re-encrypts the database files using the new randomly-generated keys. On an unencrypted database, creates a master and data encryption keys to encrypt the checkpoint and transaction log files. Encrypts the database files using the new randomly-generated keys. The `-encrAlg` option defines the AES encryption algorithm for the encryption keys. This option is required if: • You want the database files encrypted with an encryption algorithm different from the one in use. • You want to encrypt an unencrypted database. Valid values for the `-encrAlg` option are: • `AES128` • `AES192` • `AES256` See Re-Keying an Encrypted Database in *Oracle TimesTen In-Memory Database Security Guide*.

Examples

The `database1` encrypted database needs its master and data encryption keys rotated. Use the following `ttAdmin` command to rotate the encryption keys for `database1` and re-encrypt the checkpoint and transaction log files.

```
% ttAdmin -rekey database1
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

You can also use the `-encrAlg` option to change the encryption algorithm used for the database.

```
% ttAdmin -rekey -encrAlg AES256 database1
```

Notes

To rotate the encryption keys, the database must to be loaded into memory.

To encrypt an existing database for the first time, it is not necessary to define the `DbEncrypted` and `DbKeyWallet` data store attributes for the database in the `odbc.ini` file, just run the `ttAdmin` utility with the `-rekey` and `-encrAlg` options to encrypt the database.

Encryption attempts for an unencrypted database fail if the replication agent is running. See Replicating an Encrypted Database in *Oracle TimesTen In-Memory Database Security Guide*.

Force Disconnect

Options

`ttAdmin` has these options for forced disconnect:

Option	Description
`-connStr` *`connection_string`*	An ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.
`DSN`	An ODBC data source name of the database to be administered.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
`-disconnect` *`urgency`* [*`granularity`*]	Asynchronously disconnects connected applications from the database, optionally including those that are idle or unresponsive. 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 client/server processes (`ttcserver`) in order to disconnect them. **Note:** 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 cleanly unloads the database. **Note**: RAM policy `always`, described in Perform RAM Operations, conflicts with forced disconnect granularity `unload`. Using both simultaneously results in an error and the disconnect request being ignored.

Examples

This sample script uses `-disconnect` to disconnect all connections to `database1`, first using transactional urgency level then immediate urgency level:

```
#!/bin/sh

# close the databae
ttAdmin -close database1

# 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
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

```
# 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
------------------------------------------------------------------------
------------------------------------------------------------------------
Closed to user connections
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***
Shared Memory KEY 0x0210679b ID 949092358
PL/SQL Memory KEY 0x0310679b ID 949125127 Address 0x5000000000
Type            PID     Context             Connection Name              ConnID
Process         10484   0x00007f3ddfeb4010  database1 1
...
```

Notes

To enable the capability for forced disconnect, use the TimesTen connection attribute setting `ForceDisconnectEnabled=1`. See ForceDisconnectEnabled.

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`.

Close the database before attempting a forced disconnect process. Any new connection request is rejected by the main daemon during the forced disconnect process. However, after completion of forced disconnect, connection requests are accepted again if the database is not in a closed state.

Set Cache Policies

Options

`ttAdmin` has these options for cache:

Option	Description
`-connStr` *`connection_string`*	An ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.
`DSN`	An ODBC data source name of the database to be administered.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
`-cachePolicy`	Defines the policy used to determine when the cache agent for the database should run. `manual` (default) - Specifies that the cache agent must be manually started and stopped. `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. `norestart` - Specifies that the cache agent for the database is not to be restarted after a failure.
`-cacheStart`	Starts a cache agent for the database.
`-cacheStop`	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.
`-cachePwd`	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 Grant Privileges to the Oracle Cache Administration User in Oracle TimesTen In-Memory Database Cache Guide for more details.
`-cacheUid`	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 Grant Privileges to the Oracle Cache Administration User in the Oracle TimesTen In-Memory Database Cache Guide for more details.
`-cacheUidGet`	Gets the current cache administration user ID for the specified database.
`-cacheUidPwdSet`	Registers the cache administration user name and password for the specified database, using the `-cacheUid` and `-cachePwd` options. Note the following: • The `CacheAdminWallet` connection attribute specifies that credentials for the Oracle cache administration user that are set with the `-cacheUidPwdSet` option are stored in a system-managed Oracle Wallet. With the `CacheAdminWallet` connection attribute set to 0, TimesTen encrypts the credentials that you previously defined and stores them in memory. By setting the value to 1, the default value, TimesTen creates an Oracle Wallet to store the credentials. See CacheAdminWallet. • You only need to register the cache administration user ID and password once for each new database. • For all levels of `DDLReplicationLevel`, you can register the cache administration user name and password with the `-cacheUidPwdSet` option while the cache or replication agents are running. See Making DDL Changes in an Active Standby Pair in the *Oracle TimesTen In-Memory Database Replication 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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
`-stopTimeout` *`secs`*	Specifies that the TimesTen daemon should stop the cache agent if it does not stop within *`secs`* seconds. If set to `0`, the daemon potentially waits forever for the cache agent. The default value is `100` seconds.

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`:

```
% 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:

```
% ttAdmin -cachePolicy always database1
```

To turn off the automatic start of cache agent:

```
% ttAdmin -cachePolicy manual database1
```

To set the cache administration user ID and password, use `-cacheUidPwdSet` with `-cacheUid` and `-cachePwd`. For example:

```
% ttAdmin -cacheUidPwdSet -cacheUid cacheadmin -cachePwd orapwd database1
```

To get the current cache administration user ID for `database1`:

```
% 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 In-Memory Database Cache Guide* for more details.

Set Replication Policies

Options

`ttAdmin` has these options for replication:

Option	Description
`-connStr` *`connection_string`*	An ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.
`DSN`	An ODBC data source name of the database to be administered.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
`-repPolicy`	Defines the policy used to determine when the replication agent starts. `manual` (default) - Specifies that the replication agent must be manually started and stopped. `always` - Specifies that the agent should always be running for the database. This option immediately starts the replication agent. When the daemon restarts, the replication agent is restarted. `norestart` - Specifies that the replication agent for the database is not to be restarted after a failure.
`-repQueryThresholdGet`	Returns the number of seconds that a query can be run by the replication agent before TimesTen writes a warning to the daemon log. A value of `0` indicates that no warning is sent.
`-repQueryThresholdSet` *secs*	This option specifies the number of seconds that a query can be run by the replication agent before TimesTen writes a warning to the daemon 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, `ON DELETE CASCADE` 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.
`-repStart`	Starts the replication agent.
`-repStop`	Stops the replication agent.

Examples

These examples show use of replication options:

```
% ttAdmin -repPolicy always rep1
RAM Residence Policy            : inUse
Replication Agent Policy        : always
Cache Agent Policy              : manual
Cache Agent Manually Started    : False
Database State                  : Open

% ttAdmin -repPolicy manual rep1
RAM Residence Policy            : inUse
Replication Agent Policy        : manual
Replication Manually Started    : True
Cache Agent Policy              : manual
Cache Agent Manually Started    : False
Database State                  : Open

% ttAdmin -repPolicy norestart rep1
RAM Residence Policy            : inUse
Replication Agent Policy        : norestart
Replication Manually Started    : True
Cache Agent Policy              : manual
Cache Agent Manually Started    : False
Database State                  : Open

% ttAdmin -repQueryThresholdSet 100 rep1
RAM Residence Policy            : inUse
Replication Agent Policy        : norestart
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

```
Replication Manually Started    : True
Cache Agent Policy              : manual
Cache Agent Manually Started    : False
Database State                  : Open

% ttAdmin -repQueryThresholdGet rep1
QueryThreshold in seconds : 100
RAM Residence Policy            : inUse
Replication Agent Policy        : norestart
Replication Manually Started    : True
Cache Agent Policy              : manual
Cache Agent Manually Started    : False
Database State                  : Open
```

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:

```
% ttAdmin -repstart repl1
*** [TimesTen][TimesTen 26.1.1.1 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

ttAdoptStores

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 patchset or patch release. For example, you can move a database from TimesTen 26.1.1.1.0 to TimesTen 26.1.1.2.0.

> **Note**
>
> A major release refers to the first two parts of the release number (26.1 above). A patchset release refers to the third part of the release number.

This utility is useful for testing a patchset or patch 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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

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.

Syntax

```
ttadoptstores {-h | -help | -V}
ttadoptstores [-quiet] -dspath path
ttadoptstores [-quiet] -instpath path
```

Options

`ttAdoptStores` has the options:

Option	Description
`-dspath` *`path`*	Adopts a single database. The path argument must be the path to the database files (without any file extensions).
`-h`	Prints a usage message and exits.
`-help`	`-V` prints TimesTen version.
`-V`	
`-instpath` *`path`*	Adopts all databases for an instance. The path argument must be the path to the daemon working directory (the `info` directory). 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.
`-quiet`	Do not return verbose messages.
`-V | -version`	Prints the release number of `ttAdoptStores` and exits.

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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

ttBackup

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 Back Up, 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.

Syntax

```
ttBackup {-h | -help | -?}
ttBackup {-V | -version}
ttBackup -dir directory [-type backupType]
[-fname fileprefix] [-force]
{-connStr connStr | DSN}
```

Options

`ttBackup` has the options:

Option	Description
`-connStr` *`connStr`*	An ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.
`DSN`	Specifies an ODBC data source name of the database to be backed up.
`-dir` *`directory`*	Specifies the directory where the backup files should be stored.
`-fname` *`fileprefix`*	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.
`-force`	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.
`-h -help -?`	Prints a usage message and exits.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
`-type` *`backupType`*	Specifies the type of backup to be performed. Valid values are: `fileFull` (default) - Performs a full file backup to the backup path specified by the *`directory`* and *`fileprefix`* parameters. The resulting backup is not enabled for incremental backup. `fileFullEnable` - Performs a full file backup to the backup path specified by the *`directory`* and *`fileprefix`* parameters. The resulting backup is enabled for incremental backup. `fileIncremental` - Performs an incremental file backup to the backup path specified by the *`directory`* and *`fileprefix`* parameters, if that backup path contains an incremental-enabled backup of the database. Otherwise, an error is returned. `fileIncrOrFull` - Performs an incremental file backup to the backup path specified by the *`directory`* and *`fileprefix`* parameters if that backup path contains an incremental-enabled backup of the database. Otherwise, it performs a full file backup of the database and marks it incremental enabled. `streamFull` - Performs a stream backup to standard out `incrementalStop` - Does not perform a backup. Disables incremental backups for the backup path specified by the *`directory`* and *`fileprefix`* parameters. This prevents transaction log files from accumulating for an incremental backup.
`-V | -version`	Prints the release number of `ttBackup` and exits.

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 26.1.1.1.0 and 26.1.1.2.0. 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.

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

Copyright © 1996, 2026, Oracle and/or its affiliates.

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 TimesTen Database with Cache Groups in *Oracle TimesTen In-Memory Database Cache Guide* for details.

You cannot back up temporary databases.

See Also

ttBulkCp
ttMigrate
ttRestore

ttBulkCp

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`.

> **Note**
>
> Although cross-release compatibility over client/server protocol is supported in TimesTen, the tool `ttBulkCpCS` is not backward and forward release compatible; hence it can be used only for the same version client/server connections.

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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

If authentication information is not supplied in the connection string or DSN, this utility prompts for a user ID and password before continuing.

Syntax

```
Usage:
  ttBulkCp [-h | -help | -?]
  ttBulkCp -helpfull
  ttBulkCp [-V | -version]

  ttBulkCp -i [-cp numTrans|final] [-d errLevel] [-e errorFile] [-restrict]
              [-m maxErrs] [-s c] [-t errLevel] [-u errLevel]
              [-v 0|1] [-xp numRows|rollback] [-C c | -Cnone]
              [-dformat formatStr] [-tformat formatStr]
              [-tsformat formatStr] [-F firstRow] [-L lastRow]
              [-N ncharEncoding] [-Q 0|1] [-S errLevel]
              [-dateMode mode] [-[no]tblLock]
              {DSN | [-connstr] connStr}
              [owner.]tblName [dataFile ...]

  ttBulkCp -directLoad [-cp numTrans|final] [-d errLevel] [-e errorFile] [-restrict]
              [-m maxErrs] [-s c] [-t errLevel] [-u errLevel]
              [-v 0|1] [-xp numRows|rollback] [-C c | -Cnone]
              [-dformat formatStr] [-tformat formatStr]
              [-tsformat formatStr] [-F firstRow] [-L lastRow]
              [-N ncharEncoding] [-Q 0|1] [-S errLevel]
              [-dateMode mode]
              {<DSN> | [-connstr] connStr}
              [owner.]tblName [dataFile ...]

  ttBulkCp -o [-s c] [-v 0|1] [-A 0|1]
              [-C c | -Cnone] [-N ncharEncoding] [-Q 0|1]
              [-dformat formatStr] [-tformat formatStr]
              [-tsformat formatStr] [-tsprec precision]
              [-[no]forceSerializable] [-dateMode mode]
              [-nullFormat formatStr] [-restrict]
              {DSN | [-connstr] connStr}
              [owner.]tblName [dataFile]
```

Options

`ttBulkCp` has the options:

Option	Description
`-Cnone` `-Cc`	`-Cnone` disables the use of comments in the output file. `-Cc` sets 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:`~ ! @ # % ^ & * ( ) = : ; | < > ? , /` This option overrides the `COMMENTCHAR` file attribute.
`-connStr connStr`	An ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.
DSN	Specifies an ODBC data source name of the database to be copied.
`-dformat formatStr` `-D formatStr`	Sets the date format. For a list of supported fixed values, see Fixed Date, Time and Timestamp Formats. This option overrides the `DFORMAT` file attribute. The default is `ODBC`. See also: `-tformat` and `-tsformat`.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
dataFile	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.
-dateMode *dateMode*	Specifies whether ttBulkCp treats an Oracle database DATE type as a 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.
-directLoad	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.
-h -help -?	Prints a short usage message and exits.
-helpfull	Prints a longer usage message and exits.
-i	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.
-N *ncharEncoding*	Specifies the input and output character encoding for NCHAR types. Valid values are UTF8, UTF-8 or ASCII.
-o	Selects copy-out mode.
owner	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.
-Q [0 | 1]	Indicates whether character-string values should be enclosed in double quotes. 0 - Indicates that strings should not be quoted. This document refers to this mode as "no quote mode." 1 (default) - Indicates that strings should be quoted.This option overrides the QUOTES file attribute. This document refers to this mode as "quote mode."
-s *c*	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:~ ! @ # % ^ & * ( ) = : ; | < > ? , / This option overrides the FSEP file attribute.
tableName	Specifies the name of the table to be saved or loaded. This parameter is case-insensitive.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
`-tformat` *timeFormat*	Sets the time format. For a list of supported fixed values, see Fixed Date, Time and Timestamp Formats. The default value is `ODBC`. This option overrides the `TSFORMAT` file attribute. See also: `-D | -dformat` and `-tsformat`.
`-tsformat` *timestampFormat*	Sets the timestamp format. For a list of supported fixed values, see Fixed Date, Time and Timestamp Formats. 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 | -dformat` and `-tformat`.
`-V | -version`	Prints the release number of `ttBulkCp` and exits.
`-v [0 | 1]`	Sets the verbosity level. `0` - Suppresses the summary. `1` (default) - Prints a summary of rows copied upon completion.

Use the following options in copy-out (`-o`) mode only. You must have `SELECT` privileges on the specified tables.

Option	Description
`-A [0 | 1]`	Indicates whether `ttBulkCp` should suppress attribute lines in the output file. `0` (default) - `ttBulkCp` may write attribute lines into the output file. `1` - Suppresses output of attribute lines.
`-forceSerializable -noForceSerializable`	The `-forceSerializable` option indicates that `ttBulkCp` should use serializable isolation regardless of the DSN or connection string settings. This is the default behavior. `-noForceSerializable` indicates that `ttBulkCp` should honor the isolation level in the DSN or connection string. 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: `Warning: This output was produced using a non-serializable isolation level. It may therefore not reflect a transaction-consistent state of the table.` For more information on isolation modes, see Transaction Isolation Levels in *Oracle TimesTen In-Memory Database Operations Guide*.
`-nullFormat` *formatStr*	Specifies the format in which `NULL` values are printed. Valid values are: `null` (default) - The word `NULL` is printed for null fields. `empty` - Nothing is printed for null fields. 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.
`-restrict`	Prevents from overwriting an existing file while exporting data from the table to a file. See Use ttBulkCp with Restrict Mode in the *Oracle TimesTen In-Memory Database Security Guide*.
`-tsprec` *precision*	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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Use the following options in copy-in (`-i`) and directload (`-directload`) modes only. You must have `INSERT` privileges on the specified tables.

Option	Description
`-cp` *`numTrans`*	Sets the checkpoint policy for the copy in.
`-cp final`	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.

`-d error`
`-d warn`
`-d ignore`
	By default, `ttBulkCp` does not consider rows that are rejected because of constraint violations in a unique column or index to be errors.
	`-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` - 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), 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`.
`-restrict`	Prevents from importing data if the specified error file already exists. See Use ttBulkCp with Restrict Mode in the *Oracle TimesTen In-Memory Database Security Guide*.
`-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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description

`-S error`
`-S warn`
`-S ignore`
	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.

`-t error`
`-t warn`
`-t ignore`
	By default, `ttBulkCp` issues an error when a `CHAR`, `VARCHAR2`, `NCHAR`, `NVARCHAR2`, `BINARY`, `VARBINARY`, `BLOB`, `CLOB`, or `NLOB` value is longer than its maximum column width.
	`-t error` (default) - Specifies that rows containing long string or binary attributes should not be inserted into the TimesTen table and that an error should be placed into the error file.
	`-t warn` - 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.
	`-t ignore` - Specifies that long string or binary attributes should be silently truncated to the maximum column length before being inserted into the 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`
`-u warn`
`-u ignore`
	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 error` (default) - Specifies that rows containing a real, float or double value that underflow should not be inserted into the TimesTen table and that an error should be placed into the error file.
	`-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`* `-xp rollback`	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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

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.
- 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). 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 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 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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

- DATEMODE: Specifies whether an Oracle database DATE type is specified as 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 supported values, see Fixed Date, Time and Timestamp Formats. 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 supported values, see Fixed Date, Time and Timestamp Formats. 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 supported values, see Fixed Date, Time and Timestamp Formats. 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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

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.
- \u*xyzw* (NCHAR and NVARCHAR2 only) The character whose unicode value is *xyzw*, where *xyzw* is a four-digit hexadecimal number, as in\ufe4a. The \u*xyzw* 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
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

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):

String	Value
`NAN`	`NaN`
`[+]INF`	`Inf`
`-INF`	`-Inf`

TimesTen outputs the values as: `NAN`, `INF` and `-Inf`.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Fixed Date, Time and Timestamp Formats

For date values, the fixed formats are:

Format	Description
ODBC	*YYYY-MM-DD* Example: 2011-01-03 (default value)
Oracle	*DD-Mon-YYYY* Example: 03-Jan-2011
SYBASE1	*MM/DD/YYYY* Example: 01/03/2011
SYBASE2	*DD-MM-YYYY* Example: 03-01-2011
SYBASE3	*Mon*DD*YYYY* Example: Jan 03 2011

For time values, the only fixed format is ODBC:

Format	Description
ODBC	*HH24:MI:SS* Example: 07:47:23

For timestamp values, the fixed formats are:

Format	Description
ODBC	*YYYY-MM-DD*HH24:MI:SS+FF* Example: 2011-01-03 07:47:23
Oracle	*DD-Mon-YYYY*HH24:MI:SS+FF* Example: 03-Jan-2011 07:47:23
SYBASE1	*MM/DD/YYYY*HH24:MI:SS+FF* Example: 01/03/2011 07:47:23
SYBASE2	*DD-MM-YYYY*HH24:MI:SS+FF* Example: 03-01-2011 07:47:23
SYBASE3	*Mon*DD*YYYY*HH24:MI:SS+FF* Example: Jan 03 2011 07:47:23

The default timestamp value is: '*DF*TF+FF*'

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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

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.

Specifier	Descriptions and restrictions
Q	Quarter. Cannot be used in copy-in mode.
YYYY	Year (four digits).
Y,*YYY*	Year (with comma as shown).
YYY	Year (last three digits). Cannot be used in copy-in mode.
Y	Year (last digit). Cannot be used in copy-in mode.
MONTH	Month (full name, blank-padded to 9 characters, case-insensitive).
MON	Month (three character prefix, case-insensitive).
MM	Month (01 through 12).
DD	Day of the month (01 through 31).
HH24	Hour (00 through 23).
HH12	Hour (01 through 12). Must be used with AM/PM for copy-in mode.
HH	Hour (01 through 12). Must be used with AM/PM for copy-in mode.
MI	Minute (00 through 59).
SS	Second (00 through 59).
FF	Fractional seconds.Six digits, unless overridden with the `-tsprec` option.
FFn	Fractional seconds (number of digits specified by *n*).
+FF	In copy-in mode, matches, optional decimal point plus one or more fractional seconds. In copy-out mode, same as `.FF`.
+FFn	In copy-in mode, same as +FF. In copy-out mode, same as `.FF`*n*.
AM PM	Meridian indicator without dots. In copy-in mode, this must be used with *HH* or *HH12*, but not *HH24*.
A.M. P.M.	Meridian indicator with dots. In copy-in mode, this must be used with *HH* or *HH12*, but not *HH24*.
DF	Current date format (can only be used in timestamp format).
TF	Current time format (can only be used in timestamp format).
- / ; :	Punctuation that are matched in copy-in mode or output in copy-out mode.
"*text*"	Text that is matched in input mode or output in copy-out mode.
*	Matches 0 or more whitespace characters (space or tab) in copy-in mode or outputs 1 space in copy-out mode.

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.

```
##ttBulkCp
# This is a comment.
###### So is this.
# The following line is a blank line.
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

```
"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 '|':

```
##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.

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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

- 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 passthrough 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

ttCapture

The `ttCapture` utility 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.

Syntax

```
ttCapture {-h | -help | -?}
ttCapture {-V | -version}
ttCapture [-noinstinfo] [-nosysinfo] [-stdout | -dest dir] [-logdir dir]
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

```
 [dspath | DSN]
ttCapture [-noinstinfo] [-nosysinfo] [-stdout | -dest dir] [-logdir dir]
 [-noconnect] [dspath | DSN]
ttCapture -noconnect [dspath | DSN]
```

Options

`ttCapture` has the options:

Option	Description
`-dest` *dir*	Writes the output file to the designated directory.
DSN	Specifies an ODBC data source name of the database to be checked.
dspath	Specifies the fully qualified name of the database to be evaluated. This is not the DSN associated with the connection but the fully qualified database path name associated with the database as specified in the `DataStore=` parameter of the database's ODBC definition. For example, for a database consisting of files `/home/payroll/2011.ds0`, `/home/payroll/2011.ds1`, and several transaction log files `/home/payroll/2011.log`*n*, *dspath* is `/home/payroll/2011`. NOTE: The *DSN* and *dspath* options are mutually exclusive. If you do not supply either option, `ttCapture` does not provide any database information.

`-h`
`-help`
`-?`
	Prints a usage message and exits.

`-logdir` *dir*	Specifies the location of the log directory. Must be used with the `-dspath` option. If not specified, the log directory may not be available.
`-noconnect`	Specifies that the utility should capture information on the DSN without connecting to it. If specified, some information, such as `ttConfiguration` output and replication schemes, is not included in the output. This option is useful if you do not want to load a large database or if you are reporting a problem where connections are failing.
`-noinstinfo`	Indicates that `ttCapture` should not capture any installation information.
`-nosysinfo`	Indicates that `ttCapture` should not capture any system information.
`-stdout`	On UNIX and Linux systems, `ttCapture` writes all output to `stdout`, instead of writing the output to a file. On Windows, `ttCapture` writes to a `Command` prompt.
`-V | -version`	Prints the release number of `ttCapture` and exits.

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
```

> **Note**
>
> This utility is supported only where the TimesTen Data Manager is installed.

Copyright © 1996, 2026, Oracle and/or its affiliates.

ttCheck

The `ttCheck` utility 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.

Syntax

```
ttCheck {-h | -help | -?}
ttCheck {-V | -version}
ttCheck [ [-blkDir] [-compHeap] [-header] [-heap] [-indexHeap] [-log]
[-permBlkDir] [-permHeap] [-tempBlkDir] [-tmpHeap]
[-tables tblName [...]] [-users userName [...]]
[-level levelNum] ] [...]
[-m maxErrors] [-f outFile] [-v verbosity]
{DSN | [-connstr] connStr | dspath}
```

Options

`ttCheck` has the options:

Option	Description
`-blkDir`	Checks all the block directories.
`-compHeap`	Checks the compilation heap structure.
`-connStr` *`connStr`*	An ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.
`DSN`	Specifies an ODBC data source name of the database to be checked.
`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.log`*`n`*, *`dspath`* is `/home/payroll/2011`.
`-f` *`outFile`*	Specifies the output file name; defaults to `stdout`.
`-h`	Prints a usage message and exits.
`-help`	
`-?`	
`-header`	Checks the content of the database header.
`-heap`	Checks all heap structures.
`-indexHeap`	Checks the index heap structure.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
`-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 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
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

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.

ttClientImport

For Transport Layer Security, use the `ttClientImport` utility on the client to import the client wallet, `sys.odbc.ini` configuration information, and other information as applicable. This information that was exported into a ZIP file using the `ttAdmin -clientExportAll` option. The wallet is placed according to where it was located on the server. The utility updates the `sys.odbc.ini` file on UNIX or the registry on Windows with the DSN definitions.
Also see (Optional) Task 6: Import Certificates and Configuration in TimesTen in *Oracle TimesTen In-Memory Database Security Guide* for additional information and examples.

Required Privilege

This utility requires the instance administrator privilege.

Syntax

```
ttClientImport [-h | -help | -verbose | -version]

ttClientImport [options] [exportfile]
```

Options

`ttClientImport` has thes options:

Option	Description
`-h`	Displays help information.
`-help`	
`-verbose`	Shows additional output from execution of the utility.
`-version`	Displays the TimesTen release number.
`exportfile`	The path to where the export ZIP file was created on the server, to be imported into the client.

Copyright © 1996, 2026, Oracle and/or its affiliates.

ttCreateCerts

The `ttCreateCerts` utility creates Oracle wallets and certificates that are used when configuring Transport Layer Security (TLS) in TimesTen client/server and replication environments. Oracle wallets and certificates are created manually using `ttCreateCerts` for client/server TLS communications when you have multiple databases and need different certificates for each, or when you require certificates for both client/server connections and replication.

TimesTen supports several types of certificates, including self-signed certificates. A self-signed certificate is a digital certificate that is created, issued, and digitally signed by itself, rather than by a publicly trusted third-party Certificate Authority (CA). A certificate is self-signed if the issuer's name (who signed it) and the subject's name (who it belongs to) are the same.

The `ttCreateCerts` utility creates a self-signed root, client, and server certificates and uses the root certificate to sign the server's certificate for both client/server and replication. TimesTen stores the server certificate and self-signed root certificate in a server wallet. In a TimesTen client/server environment, for a client to be able to trust the server certificate, a copy of the self-signed root certificate is needed by the client. The `ttCreateCerts` utility creates a client wallet and stores the self- signed root certificate in it. See Task 2A: Create Self-Signed Certificates in *Oracle TimesTen In-Memory Database Security Guide*.

The utility creates three Oracle Wallets: `rootWallet`, `clientWallet`, and `serverWallet`.

> **Note**
>
> You must have Java JDK or JRE version 8 or higher on your system to use `ttCreateCerts`. The utility searches for the environment variables `JRE_HOME`, `JAVA_HOME`, and the `PATH` settings.

Required Privilege

This utility requires no privileges. However, depending on the specified options, it may write files to directories that require instance administrator privileges.

Syntax

```
% ttCreateCerts -h
usage: ttCreateCerts [-dir WALLETDIR] [options...]
       ttCreateCerts [-h | -help]
       ttCreateCerts [-V | -version]
       ttCreateCerts [-showWallet WALLETDIR] [-json]
       ttCreateCerts [-run "ORAPKICOMMAND"]
options:
  -f | -force
  -verbose
  -validity DAYS | -valid_from mm/dd/yyyy -valid_until mm/dd/yyyy
  -dryrun
  -fips140_mode
  -client | -noclient
  -sign_alg  sha256 | sha384 | sha512 | ecdsasha256 | ecdsasha384 | ecdsasha512
  -asym_alg ECC -eccurve p256 | p384 | p521
  -eccurve p256 | p384 | p521
  -asym_alg RSA -keysize 2048 | 4096
  -keysize  2048 | 4096
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

Options

The following table shows the supported options for `ttCreateCerts`. The second column indicates if the option is supported for ECC or RSA keys or both.

Option	ECC or RSA Support	Description
`-dir`	Supported for both ECC and RSA	This is an optional parameter. If the `-dir` parameter is not specified, the wallets are created in the default *`timesten_home`*`/conf` directory. If the `-dir` parameter is specified, the wallets are placed in the directory that already exist. Use the `-force` option to overwrite, regardless of whether the optional `-dir` parameter is used.
`-h` `-help`	Supported for both ECC and RSA	Shows help (showing the above syntax).

`-v`
`-V`
`-version`
	Supported for both ECC and RSA	Displays the TimesTen release number.

`-showWallet` *`WALLETDIR`* `[-json]`	Supported for both ECC and RSA	Shows information about the certificate in the specified `WALLETDIR` directory. If the optional `-json` parameter is specified, `ttCreateCerts` displays the output in JSON format.
`-run` *`"ORAPKICOMMAND"`*	Supported for both ECC and RSA	Runs Oracle's `orapki` utility with the commands supplied in the parameter to manage Public Key Infrastructure (PKI) elements.
`-f` `-force`	Supported for both ECC and RSA	Overwrites any previous wallets in the specified directory. **Note** The `-force` option is ignored when the`-run` option is used.
`-verbose`	Supported for both ECC and RSA	Shows additional output from execution of the utility.
`-validity`	Supported for both ECC and RSA	One of two ways to specify the lifetime of the wallets that are created, expressed as a number of days from creation. The default is 365 days, which can be overridden by setting either `-validity` or `-valid_from` and `-valid_until`.
`-valid_from` `-valid_until`	Supported for both ECC and RSA	The other way to specify the lifetime of the wallets that are created, expressed as a start and an end date in *mm/dd/yyyy* format.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	ECC or RSA Support	Description
`-dryrun`	Supported for both ECC and RSA	Echoes all the commands to be executed by `ttCreateCerts` to create the certificates as you specified, but without executing them. For options you do not set, you can use this to confirm what the default values are.
`-fips140_mode`	Supported for both ECC and RSA	Instructs `ttCreateCerts` to use FIPS mode when generating certificates. By default, `ttCreateCerts` does not run in FIPS mode.
`-client | -noclient`	Supported for both ECC and RSA	Indicates if `ttCreateCerts` adds a client certificate to the client wallet. • If `-client` is specified, `ttCreateCerts` adds a client certificate to the client wallet. • If `-noclient` is specified, `ttCreateCerts` does not add a client certificate to the client wallet. The default is `-client`.
`-sign_alg sha256 |sha384 | sha512 | ecdsasha256 | ecdsasha384 | ecdsasha512`	Supported for both ECC and RSA	Specifies the signature algorithm used to sign a certificate. Valid values are as follows: • For ECC keys, you must use one of the following elliptic curve signing algorithms: – `ecdsasha256` – `ecdsasha384` – `ecdsasha512` (default for ECC keys) • For RSA keys, you must use one of the following RSA signing algorithms: – `sha256` – `sha384` – `sha512` (default for RSA keys)
`-asym_alg ECC | RSA`	Supported for ECC or RSA	This is an optional parameter. If not specified, the default is `ECC`. The `-asym_alg` option indicates the asymmetric cryptographic algorithm to use when generating a key pair. Valid values are `ECC` (default) and `RSA`. Ensure you use uppercase (for example, `ECC` or `RSA`). The valid `-eccurve` and `-keysize` options depend on whether you are using `ECC` or `RSA`. See the Usage Rules section for more information about using the `-asym_alg` option.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	ECC or RSA Support	Description
`-eccurve p256 | p384 | p521`	Supported for ECC	`-eccurve` option is specific to ECC: • Refers to the Elliptic Curve Digital Signature Algorithm (ECDSA), which is a public-key asymmetric cryptographic algorithm that uses math from elliptic curves to create strong digital signatures. • If ECC keys and the ECC asymmetric cryptographic algorithm are used to generate certificates, then the `-eccurve` option is valid. The `-eccurve` option specifies the curve on which an ECC key is generated. • Valid values for `-eccurve` are `p256` or `p384` or `p521` (default).
`-keysize 2048 | 4096`	Supported for RSA	`keysize` option is specific to RSA: • RSA refers to a pair of cryptographic keys, a public key and a private key that are used for public-key encryption and digital signatures. • If RSA keys and the RSA asymmetric cryptographic algorithm are used to generate certificates, then the `-keysize` option is valid. The `-keysize` option sets the key size (expressed in bits) for the RSA key. Supported key sizes are `2048` or `4096` (default).

Usage Rules for ECC and RSA keys

Note the following usage rules:

- The `-asym_alg` option specifies which asymmetric cryptographic algorithm to use when generating a key pair.
  - If the algorithm is RSA, you can specify `-keysize` to set the size of the RSA key to be generated.
  - If the algorithm is ECC (default), you can specify `-eccurve` to set the ECC curve on which the key is generated.
- If you do not specify `-asym_alg`, the following rules apply:
  - If you specify `-eccurve` and `-keysize`, the `ttCreateCerts` utility returns an error.
  - If you only specify `-eccurve`, `ttCreateCerts` uses the ECC algorithm.
  - If you only specify `-keysize`, `ttCreateCerts` uses the RSA algorithm.
  - If you do not specify `-eccurve` or `-keysize`, `ttCreateCerts` uses the ECC algorithm.
- If you do not specify `-sign_alg`, the following rules apply:
  - If the algorithm is ECC (default), the `sign_alg` defaults to `ecdsasha512`.
  - If the algorithm is RSA, the `sign_alg` defaults to `sha512`.

Copyright © 1996, 2026, Oracle and/or its affiliates.

The `ttCreateCerts` utility also has a `-run` option that allows you to run commands of the Oracle `orapki` utility to manage Public Key Infrastructure (PKI) elements. Place the `orapki` command in quotes, such as in this example:

```
% ttCreateCerts -run "wallet create -wallet serverWallet -auto_login_only"
```

For more information on the discussion of using certificates signed by a certificate authority, see Task 2B: Create CA-Signed Certificates in *Oracle TimesTen In-Memory Database Security Guide*.

In addition to using `ttCreateCerts` manually, you can configure a TimesTen instance for TLS. When you do this, TimesTen runs `ttCreateCerts` to create Oracle Wallets and certificates for use at the instance level. See ttInstanceCreate.

Examples Illustrating the Use of ECC and RSA

Example 1:

Use the `ttCreateCerts` utility to create Oracle Wallets and certificates using the RSA asymmetric cryptographic algorithm with a key size of `2048` and a signing algorithm of `sha384`. This example assumes you have previously created the `wallets/rsa` subdirectory from a directory of your choice.

```
$ ttCreateCerts -dir wallets/rsa -asym_alg RSA -keysize 2048 -sign_alg sha384
ttCreateCerts : certificates created in wallets/rsa
```

Review the directory.

```
$ ls wallets/rsa
client1.cert  clientWallet  root.cert  rootWallet  server1.cert  serverWallet
```

Show Oracle Wallet information.

```
$ ttCreateCerts -showWallet wallets/rsa/rootWallet
NAME                                                    HOLDER
EXPIRATION
wallets/rsa/rootWallet/cwallet.sso                      CN=ecRoot,C=US
Sun Dec 09 12:27:23 UTC 2035


$ ttCreateCerts -showWallet wallets/rsa/serverWallet
NAME
HOLDER           EXPIRATION
wallets/rsa/serverWallet/cwallet.sso
CN=server1,C=US  Sun Dec 09 12:27:32 UTC 2035


$ ttCreateCerts -showWallet wallets/rsa/clientWallet
NAME
HOLDER           EXPIRATION
wallets/rsa/clientWallet/cwallet.sso
CN=client1,C=US  Sun Dec 09 12:27:40 UTC 2035
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

Additional Examples

This section provides `ttCreateCerts` examples that place the wallets in a `wallets` subdirectory under *timesten_home*`/conf`, where *timesten_home* is the full path to the TimesTen instance home directory.

The following example includes verbose output. (Without the `-verbose` option, only the last line is shown.)

```
% ttCreateCerts -verbose -dir timesten_home/conf/wallets
Requested Certificates:
User Certificates:
Subject:        CN=server1,C=US
Trusted Certificates:
Subject:        CN=ecRoot,C=US
Requested Certificates:
User Certificates:
Subject:        CN=client1,C=US
Trusted Certificates:
Subject:        CN=ecRoot,C=US
ttCreateCerts : certificates created in timesten_home/conf/wallets
```

Here are the results. The contents of `clientWallet` and `serverWallet` are used.

```
% ls timesten_home/conf/wallets
client1.cert  clientWallet  cwallet.sso root.cert  rootWallet  server1.cert  serverWallet
% ls timesten_home/conf/wallets/clientWallet
cwallet.sso
% ls timesten_home/conf/wallets/serverWallet
cwallet.sso
```

The next example is a dry run. No certificates are created (despite the last line). This shows only the default values:

```
% ttCreateCerts -dir timesten_home/conf/wallets -dryrun
...
+ /usr/dev_infra/platform/bin/java -d64 -Djava.security.egd=file:///dev/./urandom -cp
/scratch/myinstance/install/lib/cryptoj.jar:/scratch/myinstance/install/lib/
oraclepki.jar oracle.security.pki.textui.OraclePKITextUI
wallet create -wallet rootWallet -auto_login_only -nologo
/scratch/myinstance/install/lib/cryptoj.jar:/scratch/myinstance/install/lib/
oraclepki.jar oracle.security.pki.textui.OraclePKITextUI
wallet add -wallet rootWallet -dn CN=ecRoot,C=US -sign_alg ecdsasha512 -self_signed -
asym_alg ECC -eccurve p521 -jsafe -validity 365 -auto_login_only -nologo
/scratch/myinstance/install/lib/cryptoj.jar:/scratch/myinstance/install/lib/
oraclepki.jar oracle.security.pki.textui.OraclePKITextUI
wallet export -wallet rootWallet -dn CN=ecRoot,C=US -cert root.cert -jsafe -nologo
...
ttCreateCerts : certificates created in timesten_home/conf/wallets
```

The default settings are `-sign_alg ecdsasha512`, `-asym_alg ECC`, `-eccurve p521`, and `-validity 365` (days).

Here is another example that sets signing algorithm ECC and the size of the elliptical curve:

```
% ttCreateCerts -dir timesten_home/conf/wallets -sign_alg ecdsasha256
-eccurve p256
ttCreateCerts : certificates created in timesten_home/conf/wallets
```

This example specifies that the certificates will expire one year from when they were created:

Copyright © 1996, 2026, Oracle and/or its affiliates.

```
% ttCreateCerts -dir timesten_home/conf/wallets -validity 365
ttCreateCerts : certificates created in timesten_home/conf/wallets
```

Or, equivalently:

```
% ttCreateCerts -dir timesten_home/conf/wallets -valid_from 01/28/2025
-valid_until 01/28/2026
ttCreateCerts : certificates created in timesten_home/conf/wallets
```

This example tries to create wallets in a location where wallets already exist:

```
% ttCreateCerts -dir timesten_home/conf/wallets
ttCreateCerts: wallets is not empty, use -force to overwrite
```

This example shows how to try again using the `-force` option:

```
% ttCreateCerts -dir timesten_home/conf/wallets -f
ttCreateCerts : certificates created in timesten_home/conf/wallets
```

ttCacheInfo

The `ttCacheInfo` utility returns change-log table information for all Oracle Database tables cached in a cache group with autorefresh and information about Oracle Database objects used to track DDL statements issued on cached Oracle Database tables. The output provided by this utility is primarily intended for use by TimesTen customer support.
The same output is obtained by running the `cacheInfo.sql` script installed with TimesTen. See Installed SQL*Plus Scripts in *Oracle TimesTen In-Memory Database Cache Guide*.

Required Privilege

This utility requires no privileges.

Syntax

```
ttCacheInfo  {-h | -help | -?}
ttCacheInfo  {-o OracleNetServiceName}
             {-d DSN}
             {-u CacheAdminUsername}

ttCacheInfo  [-p prompts_OraclePwd]
```

Options

If you set the `CacheAdminWallet` connection attribute to 0, the `ttCacheInfo` utility requires you to provide both the cache administration user name and password. Hence, only the instance administrator can successfully execute `ttCacheInfo` without specifying a password. See CacheAdminWallet.

If you set the `CacheAdminWallet` connection attribute to 1, the `ttCacheInfo` utility requires only the cache administration user name. It retrieves the registered password from the wallet that contains the credentials. Any user that is not the instance administrator must specify the `-p` option and enter the Oracle cache administration user password when prompted.

The `ttCacheInfo` utility supports the options:

Option	Description
`-d`	The DSN of the database whose information is being returned.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
-h -help -?	Prints a usage message and exits.
-o	The OracleNetServiceName on which to collect the information.
-p	If set, it indicates that ttCacheInfo should prompt for the Oracle cache administration user password.
-u	The cache administration user name.

Output

The following shows the ttCacheInfo output without the -p option

```
$ ttCacheInfo -o oracledb -u cacheadmin -d rep3a

***************** Database Information*********************
Database name: Database1
Unique database name: database1
Primary database name:
Database Role: PRIMARY
Database Open Mode: READ WRITE
Database Protection Mode: MAXIMUM PERFORMANCE
Database Protection Level: UNPROTECTED
Database Flashback On: NO
Database Current SCN: 1163173
************************************************************
**************Autorefresh Objects Information**************
Host name: host3
Timesten datastore name: /tmp/datastores/database1
Cache table name: customers
Has after insert trigger: YES
Change log table name: TT_07_74698_L
Number of rows in change log table: 2
Maximum logseq on the change log table: 0
Timesten has autorefreshed updates upto logseq: 0
Number of updates waiting to be autorefreshed: 0
Number of updates that has not been marked with a valid logseq:0
****************************************
****************No DDL Tracking objects are found****************
PL/SQL procedure successfully completed.
```

ttCWAdmin

Manages TimesTen active standby pairs that take advantage of the high availability framework of Oracle Clusterware. This utility starts administrative processes, generates scripts and

Copyright © 1996, 2026, Oracle and/or its affiliates.

performs other functions to administer active standby pairs and the corresponding Clusterware resources.

> **Note**
>
> This utility is not supported on Linux for ARM (aarch64) systems.

For information about using Oracle Clusterware, see Using Oracle Clusterware to Manage Active Standby Pairs in *Oracle TimesTen In-Memory Database Replication Guide*.

These options are supported with advanced high availability. The options are not supported for basic high availability:

- `ttCWAdmin -addMasterHosts`
- `ttCWAdmin -addSubscriberHosts`
- `ttCWAdmin -createVIPs`
- `ttCWAdmin -delMasterHosts`
- `ttCWAdmin -delSubscriberHosts`
- `ttCWAdmin -dropVIPs`

Required Privilege

On Windows, any user with Administrators privileges can run all commands in this utility.

On supported UNIX and Linux systems, the `root` user can run all commands in this utility. These commands must be run by the `root` user:

- `ttCWAdmin -addMasterHosts`
- `ttCWAdmin -addSubscriberHosts`
- `ttCWAdmin -createVIPs`
- `ttCWAdmin -delMasterHosts`
- `ttCWAdmin -delSubscriberHosts`
- `ttCWAdmin -ocrConfig`
- `ttCWAdmin -dropVIPs`

The administrator user can run 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.

Syntax

```
ttCWAdmin {-h | -help | -?}

ttCWAdmin {-V | -version}

ttCWAdmin -init [-hosts "host_name1, host_name2[, ...]"]

ttCWAdmin {-createVIPs | -dropVIPs | -create | -drop | -restore | -start |
           -stop | -status} [-ttclusterini fileName] [-dsn DSN]

ttCWAdmin - [-timeout seconds] -dsn DSN
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

```
ttCWAdmin -relocate -dsn DSN

ttCWAdmin -reauthenticate -dsn DSN

ttCWAdmin -ocrConfig

ttCWAdmin -beginAlterSchema -dsn DSN

ttCWAdmin -endAlterSchema -dsn DSN

ttCWAdmin -addMasterHosts [-hosts "host_name1, host_name2[, ...]"] -dsn DSN

ttCWAdmin -delMasterHosts [-hosts "host_name1, host_name2[, ...]"] -dsn DSN

ttCWAdmin -addSubscriberHosts [-hosts "host_name1, host_name2[, ...]"] -dsn DSN

ttCWAdmin -delSubscriberHosts [-hosts "host_name1, host_name2[, ...]"] -dsn DSN

ttCWAdmin -start [-noapp] -dsn DSN

ttCWAdmin -stop -dsn DSN

ttCWAdmin -startapps -dsn DSN

ttCWAdmin -stopapps -dsn DSN

ttCWAdmin -shutdown [-noderegister] [-hosts "host_name1, host_name2[, ...]"]
```

Options

`ttCWAdmin` has these options:

Option	Description
`-addMasterHosts`	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 run this command.
`-addSubscriberHosts`	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 run this command.
`-beginAlterSchema`	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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
`-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 administration user name. • Prompts for the TimesTen cache administration user password. • If cache groups are being used, prompts for the Oracle cache administration user password. This password is provided in the `OraclePWD` connection attribute when the cache administration user connects. This Oracle cache administration user sets 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 run 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 run 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

`-h`
`-help`
`-?`
	Prints a usage message and exits.

`-init`	Starts the TimesTen cluster agent.
`-noderegister`	Used with `ttCWAdmin -shutdown` 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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
`-ocrConfig`	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.
`-reauthenticate`	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. • 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 administration user name. • Prompts for the password of the TimesTen cache administration user. • If cache groups are being used, prompts for the password for the Oracle cache administration user. This password is provided in the `OraclePWD` connection attribute. This Oracle cache administration user is used to set the autorefresh states for cache groups. • Prompts for a random string used to encrypt the above information. For more details, see Changing User Names or Passwords When Using Oracle Clusterware in the *Oracle TimesTen In-Memory Database Replication Guide*.
`-relocate`	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.
`-restore`	Restores the active master database from the backup specified by `RepBackupDir`. Do not use this command when `AutoRecover` is enabled.
`-shutdown`	Stops the TimesTen daemon, cluster agent, and the replication agent (if the replication agent is still up) on the set of hosts either specifically mentioned with the optional `-hosts` argument or defined within the `ttcrsagent.options` file. Also, if this command does not include the `-noderegister` option, then the default behavior is to deregister from Clusterware all TimesTen processes that are registered as Clusterware resources for cluster agents and TimesTen daemon monitors.
`-start [-noapp]`	Starts the cluster active standby pair. This results in starting all of the agents on the active database, creation of the standby database and the subscriber databases (if they exist) through duplicate if necessary, and subsequent starting of all agents on those databases. If you specify `-noapp`, the applications are not started. You can use the `-startapps` option to start the applications later.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
`-startapps`	Starts the applications in the cluster.
`-stopapps`	Stops the applications in the cluster.
`-status`	Obtains the status of resources in the cluster.
`-stop`	Stops the replication agent and the cache agent and disconnects the application from both databases of an active standby pair.
`-`	Reverses the role of an active standby pair in a cluster. The standby database becomes the new active, while the existing active database becomes the standby database.
`-timeout` *seconds*	Specifies a timeout value for the `-` option. Specify an integer value greater than 0. The default is `900` seconds. If you enter an invalid value, TimesTen uses the default value of `900` seconds. If the timeout expires, TimesTen returns an error message and fails to verify the standby database.
`-dsn` *DSN*	Specifies the DSN for the active standby pair.
`-hosts "`*host_name1*`,` *host_name2*`[, ...]"`	Specifies the hosts on which to start or shut down the TimesTen cluster agent. If this option is not specified, the TimesTen cluster agent is started or stopped on all hosts.
`-ttclusterini` *filename*	Specifies the full path name of the `cluster.oracle.ini` file. The default location is in the daemon home directory. The default location is recommended.
`-V | -version`	Prints the release number of `ttCWAdmin` and exits.

Examples

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`.

Copyright © 1996, 2026, Oracle and/or its affiliates.

The TimesTen integration with Oracle Clusterware accomplishes these operations with the `ttCWAdmin` utility and the attributes in the `cluster.oracle.ini` file.

ttDaemonAdmin

Starts and stops the TimesTen main daemon and server.

Required Privilege

This utility requires the instance administrator privilege.

Syntax

```
ttDaemonAdmin {-h | -help | -?}
ttDaemonAdmin {-V | -version}
ttDaemonAdmin [-force] {-start | -stop | -restart}
ttDaemonAdmin [-startserver | -restartserver]
ttDaemonAdmin [-force] -stopserver
ttDaemonAdmin -verbose
```

Options

`ttDaemonAdmin` has the options:

Option	Description

`-h`
`-help`
`-?`
	Prints a usage message and exits.

`-force`	Starts or stops the TimesTen main daemon, even when warnings are returned or with `-stopserver` immediately stops the server processes.
`-restart`	Restarts the TimesTen main daemon.
`-restartserver`	Restarts the TimesTen server.
`-start`	Starts the TimesTen main daemon.
`-startserver`	Starts the TimesTen server daemon.
`-stop`	Stops the TimesTen main daemon.
`-stopserver`	Stops the TimesTen server daemon. Without the `-force` option, client/server connections to TimesTen databases are gracefully disconnected after completing any request they may be processing, and then the server exits. With the `-force` option, client/server connections to TimesTen databases are forcefully and immediately terminated, and then the server exits.
`-V | -version`	Prints the release number of `ttDaemonAdmin` and exits.

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 `-force` option with caution, as it may leave databases in a state where you must perform recovery procedures.

Copyright © 1996, 2026, Oracle and/or its affiliates.

- When a TimesTen instance is under the control of systemd, using `ttDaemonAdmin` to directly control the Timesten instance results in an error and notifies the user to use `systemctl` to start and stop the process.
- When you use this utility on Windows, you must be running with Windows Administrative privileges.When you stop the daemon (`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 (`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:

  ```
  ttCWAdmin -shutdown -hosts localhost
  ```

- When you use this utility to restart the server, the TimesTen daemon reads the `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 `timesten.conf` options:

  ```
  server_port server_shmipc server_shmsize noserverlog
  ```

See Also

For a description of all daemon options and instructions for changing the `timesten.conf` file, see TimesTen Instance Configuration File in this reference and Managing TimesTen Daemon Attributes in *Oracle TimesTen In-Memory Database Operations Guide*.

ttDaemonLog

Controls and displays daemon log messages.

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 daemon 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.

Syntax

```
ttDaemonLog [-h | -help | -?]
 ttDaemonLog [-V | -version]
 ttDaemonLog [-show type] [-b | -r | -s] [-f] [-maxLines]
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

```
[-file filename] [-facility facilityName]
[-loglevel level [DSN | [-connstr] connStr]]
[-[no]logcomponent component [DSN | [-connstr] connStr]]
[-logreset] [-msg messageString] [-setquiet | -setverbose]
```

> **Note**
>
> - 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:

Option	Description
`-b`	Prints all TimesTen-generated log entries.
`-f`	When the end of the log is reached, `ttDaemonLog` does not terminate but continues to run, 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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description

`-logcomponent` *component*
`-nologcomponent` *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 `CAC`
	`DAEMON`: For messages from the main daemon and subdaemons
	`DAEMONDBG`: For additional information from the main daemon and subdaemons
	`REPLICATION`: For messages from the replication agent, designated by `REP`
	`SERVER`: For messages from the server
	`TTSTATS`: For messages from `ttStats`

`-loglevel` *level*	Specifies a cutoff for the level of messages that are logged in the daemon 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.
`-n` *computer*	Displays the log from a different computer. Specify the Universal Naming Convention (UNC) name of the target computer. **Note:** This option applies only on Windows and only if you are using the Windows Event Log for TimesTen logging, which is not typical usage.
`-r`	Prints only the TimesTen replication agent log. (Same as `-show replication`.)
`-s`	Prints only the TimesTen Server log. (Same as `-show server`.)
`-setverbose` `-setquiet`	Enables (`-setverbose`) or disables (`-setquiet`) TimesTen verbose logging.
`-show` *type*	When you use `ttDaemonLog` to display log messages to the standard output, you can use the `-show` option with one of the following types to limit the displayed log messages to that type only: `all` (default): Shows all messages. `replication`: Shows only log messages from replication agents. (Same as `-r` option.) `cache`: Shows only log messages from cache agents. `server`: Shows only log messages from TimesTen Server. (Same as `-s` option.) **Note:** You cannot show a category whose logging has been disabled through `-[no]logcomponent`.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
-V | -version	Prints the release number of ttDaemonLog and exits.

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 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 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
```

On UNIX and Linux systems, to direct logging to the local7 facility:

```
% ttDaemonLog -facility local7
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

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 Error, Warning, and Informational Messages in *Oracle TimesTen In-Memory Database Operations Guide*.

ttDestroy

Destroys a database, including all checkpoint files, transaction logs, and daemon catalog entries (though not the DSNs). If the `CacheAdminWallet` connection attribute is set to 1, `ttDestroy` also deletes the associated Oracle Wallet containing Oracle cache administrator credentials.

Required Privilege

This utility requires the instance administrator privilege.

Syntax

```
ttDestroy {-h | -help | -?}
ttDestroy {-V | -version}
ttDestroy [[-wait] [-timeout secs]] [-force] {-connStr connection_string |
 DSN | dspath}
```

Options

`ttDestroy` has the options:

Option	Description
`-connStr` *`connection_string`*	An ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.
`DSN`	Specifies an ODBC data source name of the database to be destroyed.
`dspath`	The fully qualified name of the database to be destroyed. This is not the DSN associated with the connection but the fully qualified database path name associated with the database as specified in the `DataStore=` parameter of the database's ODBC definition. 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`.

`-h`
`-help`
`-?`
	Prints a usage message and exits.

`-force`	Destroy even if files are from an incompatible version or a different instance of TimesTen.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
`-timeout` *seconds*	Indicates the time in seconds that `ttDestroy` should wait. If no timeout value is supplied, TimesTen waits five seconds before retrying the destroy operation.
`-V | -version`	Prints the release number of `ttDestroy` and exits.
`-wait`	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.

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.
- `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, run 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.

ttExporter

On Linux x8664 systems, the `ttExporter` utility enables Prometheus to monitor TimesTen health and operations. Prometheus is an open source systems monitoring and alerting toolkit. It collects and stores metrics from a variety of sources. It has its own time-series database and time-series query language.
The TimesTen Exporter converts TimesTen metrics into the form used by Prometheus. This integration enables you to add TimesTen to the systems that you monitor with Prometheus.

For information on configuring Prometheus and details on the metrics monitored and how to view the metrics, see The TimesTen Prometheus Exporter in *Oracle TimesTen In-Memory Database Monitoring and Troubleshooting Guide*.

Required Privilege

The operating system user must match a user in each TimesTen database in the instance that Prometheus monitors. If you run `ttExporter` as the instance administrator user, no further privilege is required. If, however, you wish to run `ttExporter` as a different operating system user, then that user must exist in each TimesTen database in the instance monitored by Prometheus. In addition, the operating system user must be granted the `CREATE SESSION` privilege. For example, if you want the `osuser1` operating system user to run `ttExporter`, then you must do the following in each TimesTen database in the instance monitored by Prometheus:

```
Command> CREATE USER osuser1 IDENTIFIED EXTERNALLY;

User created.

Command> GRANT CREATE SESSION TO osuser1;
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

Syntax

```
ttExporter -h | -help

ttExporter -create-server-certificate [-rsa-key-size bits][-certificate-common-name
dnsip1] [-certificate-alt-names dnsip2]  [-certificate-directory mycertdir]

ttExporter -export-server-certificate serverfilename [-certificate-directory mycertdir]

ttExporter -export-client-certificate clientfilename -export-client-private-key keyfile
  [-certificate-directory mycertdir]

ttExporter [-insecure] [-limit-rate r] -port p [-pid-file pidfilename] [-d]
  [-certificate-directory mycertdir]
```

Options

`ttExporter` has these options:

Option	Description
`-h` `-help`	Prints a usage message and exits.
`-certificate-directory` *`mycertdir`*	Defines the directory where an Oracle Wallet containing certificate information used by the Exporter is stored. (*`mycertdir`* is the location of the directory in this example.) If you specify `-certificate-directory`, you must specify this option for all invocations of `ttExporter`. This includes when creating a server certificate, when exporting a server certificate, when creating and exporting both a client certificate and a client private key, and when starting the Exporter. If you do not specify a directory, the certificates are stored in an Oracle Wallet in the home directory (`$HOME`) of the user.
`-create-server-certificate [-rsa-key-size` *`bits`*`]`	Creates a new server certificate. The Exporter uses the server certificate to authenticate itself to clients and to authenticate client certificates. Creating a new server certificate invalidates any previously exported client certificates. The key size specified by `-rsa-key-size` can be a bit count value of `2048` or larger. If not specified, the default is `2048`. The `-rsa-key-size` option is not required. Specifying this option does not start the Exporter.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
[-certificate-common-name *dnsip1*] [-certificate-alt-names *dnsip2*]	Defines additional options for creating a server certificate. Must be used with the -create-server-certificate option. The -certificate-common-name *dnsip1* option lets you specify a *Common Name* (CN) for the certificate. It matches the DNS name or the IP address where the certificate is installed. This CN can contain only one name. Single-level wildcards are acceptable. Replace the *dnsip1* variable with this CN. The -certificate-alt-names *dnsip2* option lets you specify a *Subject Alternative Name* (SAN). This is a structured way to indicate all of the domain names and IP addresses that are secured by and included in the certificate. Replace the *dnsip2* variable with this SAN. The SAN includes the CN mentioned previously as well as any other DNS names or IP addresses that need access to the TimesTen Exporter. Single level wildcards are acceptable.
-d	Starts the Exporter in debug mode and writes log messages to standard output instead of syslog. (Log messages are written to syslog by default.)
-export-server-certificate *serverfilename*	Exports the server certificate in PEM format. This example exports the certificate to the *serverfilename* file. Specifying this option does not start the Exporter.
-export-client-certificate *clientfilename* -export-client-private-key *keyfile*	Creates and exports the client certificate to a file (*clientfilename*, in this example) and the client private key to a file (*keyfile*, in this example). You must specify these two options together. Specifying this option does not start the Exporter.
-insecure	Starts the exporter in insecure mode (no authentication), using the HTTP protocol. In this mode, the exporter does not authenticate itself using its server certificate and does not authenticate client certificates.
-limit-rate *r*	Sets the limit of HTTP (or HTTPS) GET requests per minute, where *r* equals the number of requests. The value of *r* can be any integer value from 1 to 15. If this option is not specified, the Exporter responds to at most 10 HTTP (or HTTPS) GET requests per minute.
-pid-file *pidfilename*	When starting the Exporter, you can optionally specify the -pid-file option. If specified, ttExporter writes its process ID into a file (*pidfilename*, in this example). You can then stop the Exporter by terminating the process ID stored in this file. This is an alternative to stopping the Exporter with a SIGINTR or SIGTERM signal.
-port *p*	Sets the listening port number (*p*, in this example) for the Exporter. You must set the port number whenever starting the Exporter.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Examples

To start the Exporter without authentication, use:

```
% ttExporter -insecure -port 12345
```

To use client certificate authentication:

Create a server certificate one time. Store the certificate in an Oracle Wallet located in the directory specified by `-certificate-directory`. The directory is `mycertdir` in this example:

```
% ttExporter -create-server-certificate -certificate-directory mycertdir
```

After creating the server certificate, export it in PEM format to a file. The `-certificate-directory` option must be specified (as it was specified when you created the server certificate):

```
% ttExporter -export-server-certificate mycertdir/server.crt -certificate-
directory mycertdir
```

After exporting the server certificate, create and export both a client certificate and a client private key. You must create and export a client certificate and a client private key for each Prometheus instance that scrapes metrics from the Exporter. The `-certificate-directory` option must be specified (as it was specified when you created the server certificate):

```
% ttExporter -export-client-certificate mycertdir/client.crt
    -export-client-private-key mycertdir/key.crt -certificate-directory
mycertdir
```

After creating and exporting the client certificate and the client private key, start the Exporter. Supply the `-pid-file` option if you want `ttExporter` to write its process ID into a file (`/tmp/ttexporter.pid`, in this example). The `-certificate-directory` option must be specified (as it was specified when you created the server certificate):

```
% ttExporter -port 12345 -pid-file /tmp/ttexporter.pid -certificate-directory
mycertdir
```

ttInstallationCheck

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

Copyright © 1996, 2026, Oracle and/or its affiliates.

Required Privilege

This utility requires the instance administrator privilege.

Syntax

```
ttInstallationCheck [-h | -help | -?]
ttInstallationCheck [-v | -verbose | -?]

ttInstallationCheck [-install_dir path] [-generate]
```

Options

`ttInstallationCheck` has the options:

Option	Description

`-h`
`-help`
`-?`
	Displays help information.

`-generate`	Generate and print the checksum for the installation, but do not verify.
`-install_dir path`	Specifies the directory in which TimesTen is installed.
`-v | -verbose`	Displays extra installation information.

ttInstallDSN

The `ttInstallDSN` utility is a Windows-only utility. It 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. For more information, see Create a User or System odbc.ini File in *Oracle TimesTen In-Memory Database Operations Guide*.

Required Privilege

This utility requires the instance administrator privilege.

Syntax

```
ttInstallDSN [-h | -help | -?]

ttInstallDSN [-f file] [Client_DSN_Name | -a | -l] [-force]
```

Options

`ttInstallDSN` has the options:

Option	Description

`-h`
`-help`
`-?`
	Displays help information.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
`-f file [Client_DSN_Name | -a | -l]`	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: • The name of the DSN you want to install from the input file. • The `-a` option to install all DSNs from the input file. • The `-l` option to simply list all the DSNs in the input file. None is installed. If you do not specify `-f`, the default file is `sys.odbc.ini` in the current directory. 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.
`-force`	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.

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

The `ttInstanceCreate` utility creates a new TimesTen instance.

Required Privilege

Run this utility as the instance administrator.

Syntax

```
% ttInstanceCreate -h
ttInstanceCreate - Creates a new instance
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

```
Usage:

  ttInstanceCreate [common_options] [server_options] [classic_options]

    Creates a classic instance

  ttInstanceCreate [common_options] -clientonly

    Creates a Client-Only instance

common_options:

  -name <instance_name> Specifies the target instance name
  -location <path>      Specifies the target instance directory
  -force                Removes any existing instance in <location> without confirmation
  -record <file>        Creates instance and records prompt responses to the given file
  -batch <file>         Reads answers for prompted questions from the given file
  -strict               Prevents instance creation on unsupported platforms
  -tnsadmin <path>      Specifies the path for the TNS_ADMIN environment variable
  -verbose              Displays extra information
  -help|h               Displays this help message and exits
  -clientonly           Creates a client instance
  -ssl_fips_140         Enables SSL (TLS) FIPS encryption for both client and server
  -db_fips_140          Enables FIPS symmetric encryption (TDE) for both client and
server

server_options:

  -nostart              Do not start the daemon (even in interactive mode)
  -daemonport <port>    The port on which the daemon of the created instance will listen
  -csport <port>        The port on which the C/S server of the created instance will
listen
  -systemd              Enable systemd
  -serverencryption <requirement>   Whether C/S traffic is encrypted.
  -asym_alg <algorithm> Asymmetric algorithm for certificate generation: ECC, RSA
  -serverciphersuites <suites>      Cipher suites used for TLS encryption. One of, or a
comma-separated list of:
                        TLS_AES_128_CCM_SHA256  (FIPS=yes, ECC=yes, RSA=yes)
                        TLS_AES_128_GCM_SHA256  (FIPS=yes, ECC=yes, RSA=yes)
                        TLS_AES_256_GCM_SHA384  (FIPS=yes, ECC=yes, RSA=yes)
                        TLS_CHACHA20_POLY1305_SHA256    (FIPS=no , ECC=yes, RSA=yes)
                        TLS_DHE_RSA_WITH_AES_128_GCM_SHA256     (FIPS=yes, ECC=no ,
RSA=yes)
                        TLS_DHE_RSA_WITH_AES_256_GCM_SHA384     (FIPS=yes, ECC=no ,
RSA=yes)
                        TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (FIPS=yes, ECC=yes,
RSA=no )
                        TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (FIPS=yes, ECC=yes,
RSA=no )
                        TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256   (FIPS=yes, ECC=no ,
RSA=yes)
                        TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384   (FIPS=yes, ECC=no ,
RSA=yes)
classic_options:

  -start               Starts the daemon after instance creation (default when running
this tool in interactive mode)
```

Options

`ttInstanceCreate` has the following options:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
Common options:	

`-h`
`-help`
`-?`
	Displays help information.

`-name` *`name`*	Specifies the name of the instance to be created.
`-location` *`path`*	Specifies the directory path where the instance is 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 `-name` option. This new directory is the new instance home.
`-force`	Specifies that an instance directory specified with the `-force` option is to be overwritten if it already exists. The directory is overwritten only if: • The specified instance directory is empty, or • The specified instance directory contains a `conf/timesten.conf` file.
`-record` *`filename`*	Records responses to installation questions into the file specified by *`filename`*. The file then can be specified as the parameter to the `-batch` option.
`-batch [`*`filename`*`]`	Specifies the file to be used to provide input to the `ttInstanceCreate` utility. If not specified, no input file is used.
`-strict`	Ensures that the platform running the command is supported and prevents the instance creation if it is not.
`-tnsadmin` *`path`*	For cache operations, this option configures the location to be used for the `TNS_ADMIN` setting.
`-verbose`	Displays additional information during the operation of the utility.
`-clientonly`	Specifies that the instance is client-only.
`-ssl_fips_140`	Enables SSL (TLS) FIPS encryption for both client and server.
`-db_fips_140`	Enables FIPS symmetric encryption (TDE) for both client and server.
Server options:	
`-nostart`	Do not start the daemon.
`-daemonport` *`port`*	The port number on which the TimesTen daemon process (`timestend`) for this instance listens. This port must not already be in use by any other application or instance on the system.
`-csport` *`port`*	The value to be used for the TimesTen client/server port number for this instance. If not specified, the default is *`daemonport`* `+ 1`.
`-systemd`	Specifies that the instance's main daemon will be managed by systemd.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
`-serverEncryption` *`requirement`*	Determines whether encryption is accepted, rejected, requested, or required for a client/server connection. • `accepted`: Enable an encrypted session if required or requested by the client; use an unencrypted session otherwise. This is the default. • `rejected`: Demand an unencrypted session. (If the server does not support encryption, TimesTen behaves as if this is the setting on the server.) The connection is rejected if the client requires encryption. • `requested`: Request an encrypted session if the client allows it (if the client has any setting other than rejected); use an unencrypted session otherwise. • `required`: Demand an encrypted session. Reject the connection if the client rejects encryption. With a setting other than rejected, the `ttInstanceCreate` utility generates certificates for TLS if there are compatible settings for server encryption between the server and client (as long as there are also compatible settings for cipher suites). See Transport Layer Security for TimesTen Client/Server in *Oracle TimesTen In-Memory Database Security Guide*. The `-serverEncryption` and `-serverCipherSuites` values are set at the instance creation level in the `timesten.conf` file and serve as the default values for any database created on that instance. However, you can override them by setting the corresponding connection attribute in the database definition. **Note:** If you did not create certificates when you created the instance, you need to run the ttCreateCerts utility manually to use TLS on a particular database.
`-asym_alg` *`ECC|RSA`*	Specifies the asymmetric cryptographic algorithm to use when generating a key pair for TLS. The valid values are ECC or RSA. The `ttCreateInstance` utility passes the `-asym_alg` option to the `ttCreateCerts` utility. See ttCreateCerts. The `-asym_alg` option is optional. If you do specify it, ensure that the algorithm you choose matches the cipher suites you choose. For example, if you specify RSA, ensure your list of cipher suites include one or more RSA-specific cipher suites. If you do not specify `-asym_alg`, TimesTen does the following: • Checks to see which of the algorithms (ECC or RSA) is common to the given cipher suites, and uses that one. • If both ECC and RSA are commonly used, TimesTen uses ECC. • If there is no common algorithm, TimesTen generates an error.
`-serverCipherSuites` `{`*`suites`*`}`	Lists the cipher suite or cipher suites that can be used for Transport Layer Security, depending also on the client setting. There is no default setting. Specify one or more TLS version 1.3 or 1.2 cipher suites in a comma-separated list from strongest to weakest. To use TLS, the server and client settings must include at least one common suite. See Cipher Suites in TimesTen in *Oracle TimesTen In-Memory Database Security Guide* to learn more about the supported cipher suites and how TimesTen negotiates cipher suite selection. For more information on TLS for encrypted communication between clients and server, see Transport Layer Security for TimesTen Client/Server in *Oracle TimesTen In-Memory Database Security Guide*.
Classic options:	
`-start`	Specifies that the instance should be started after it is created.

Copyright © 1996, 2026, Oracle and/or its affiliates.

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 the information needed to create the instance.

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.

Examples

This example shows how a new TimesTen client only instance named `testclient` in the `/scratch/mylocation` location is created.

```
% ttInstanceCreate -name testclient -location /scratch/myinstance -clientonly
Creating instance in /scratch/myinstance/testclient ...
The 26.1 Release Notes are located here :
  '/scratch/newinstall/tt26.1.1.1.0/README.html'

Instance created successfully.
```

In this example, the utility creates an instance named `test112` in the `/scratch/mylocation` location with the specified cipher suite `TLS_AES_128_CCM_SHA256`. It generates the ECC key for the cipher suite `TLS_AES_128_CCM_SHA256`.

```
% ttInstanceCreate -name test112 -location /scratch/mylocation -serverEncryption
accepted -serverciphersuites TLS_AES_128_CCM_SHA256
Creating instance in /scratch/mylocation/test112 ...
INFO: Creating certificates, this may take some time ...
ttCreateCerts : certificates created in /scratch/mylocation/test112/conf

NOTE: The TimesTen daemon startup/shutdown scripts have not been installed.

The startup script is located here :
        '/scratch/mylocation/test112/startup/tt_test112'

Run the 'setuproot' script :
        /scratch/mylocation/test112/bin/setuproot -install
This will move the TimesTen startup script into its appropriate location.

The 26.1 Release Notes are located here :
  '/scratch/mylocation/newinstall/tt26.1.1.1.0/README.html'

Instance created successfully.
```

In this example, `ttInstanceCreate` generates the RSA key for encryption at the instance level.

```
% ttInstanceCreate -name test121 -location /scratch/mylocation -serverEncryption
accepted -serverciphersuites TLS_AES_128_CCM_SHA256 -asym_alg RSA
Creating instance in /scratch/mylocation/test121 ...
INFO: Creating certificates, this may take some time ...
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

```
ttCreateCerts : certificates created in /scratch/mylocation/test121/conf

NOTE: The TimesTen daemon startup/shutdown scripts have not been installed.

The startup script is located here :
        '/scratch/mylocation/test121/startup/tt_test121'

Run the 'setuproot' script :
        /scratch/mylocation/test121/bin/setuproot -install
This will move the TimesTen startup script into its appropriate location.

The 26.1 Release Notes are located here :
  '/scratch/mylocation/newinstall/tt26.1.1.1.0/README.html'

Instance created successfully.
```

This example shows that the asymmetric algorithm ECC is not compatible with the TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 cipher suite.

```
% ttInstanceCreate -name test12 -location /scratch/mylocation -serverEncryption accepted
-serverciphersuites TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 -asym_alg ECC
ERROR: Requested asymmetric algorithm 'ECC' not compatible with all the given cipher
suites.
Failed to create instance.
```

This example shows when a compatible cipher suite that resolves to RSA is passed.

```
% ttInstanceCreate -name test12 -location /scratch/mylocation -serverEncryption accepted
-serverciphersuites TLS_DHE_RSA_WITH_AES_128_GCM_SHA256,TLS_AES_256_GCM_SHA384
[...]
Will use asym_alg 'RSA'
```

ttInstanceDestroy

Use the ttInstanceDestroy utility to destroy an existing instance.

The instance to be destroyed is chosen based on the current setting of the TIMESTEN_HOME environment variable.

Required Privilege

This utility requires the instance administrator privilege.

Syntax

```
ttInstanceDestroy {-h | -help } [-verbose | -v]

ttInstanceDestroy [-force]
```

Options

ttInstanceDestroy has the options:

Option	Description
-h -help -?	Displays help information.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
`-force`	If specified, you are not asked to confirm operations. If `-force` is not specified: • You are reminded that if you installed the startup scripts for this instance as `root`, you must uninstall them as root with the `setuproot -uninstall` command. • You are asked for confirmation to destroy the instance. • You are asked for confirmation to remove the `info` and `conf` directories.
`-verbose`	Displays additional TimesTen installation information.

ttInstanceModify

Use the `ttInstanceModify` utility to modify certain attributes of an instance.

The attributes that you can modify are:

- The installation associated with this instance
- The daemon and server port numbers
- The `TNS_ADMIN` path for the instance
- The configuration of TimesTen Replication with Oracle Clusterware for this instance
- TLS settings, if any
- The system start up scripts

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:

1. Shuts down the TimesTen daemon for the instance.
2. Edits the `timesten.conf` file in the *`timesten_home`*`/conf` directory.
3. Starts the TimesTen main daemon for the instance.

Required Privilege

Run this utility as the instance administrator.

Syntax

```
% ttInstanceModify -h
ttInstanceModify - Modifies an existing instance

Usage:
  ttInstanceModify { -help | -h }
  ttInstanceModify [-install <path>] [-tnsadmin <path>] [-verbose | -v]
      [-port <number>] [-serverport <number>] [-crs]
      [-serverencryption <requirement>] [-serverciphersuites <suites>] [-serverwallet
<path>]

Options:
  -help, -h              Displays this help message
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

```
-verbose, -v           Displays extra information
-install               Changes the installation the instance points to
-tnsadmin <path>       Changes the path for the TNS_ADMIN environment variable
-port <number>         Specifies a new daemon port number
-serverport <number>   Specifies a new server port number
-[no]systemd           Enables/disables systemd to manage TimesTen Daemon
-serverencryption <requirement>  Whether client/server traffic is encrypted.
                       One of: accepted rejected requested required
-serverciphersuites <suites>     Cipher suites used for TLS encryption.
                       One of, or a comma-separated list of:
                         TLS_AES_128_CCM_SHA256
                         TLS_AES_128_GCM_SHA256
                         TLS_AES_256_GCM_SHA384
                         TLS_CHACHA20_POLY1305_SHA256
                         TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
                         TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
                         TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
                         TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
                         TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
                         TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
-serverwallet <path>   Absolute path to the wallet directory
-crs                   Creates or modifies Clusterware configuration
```

Options

`ttInstanceModify` has the following options:

Option	Description

`-h`
`-help`
`-?`
	Displays help information.

`-verbose`	Displays additional information during the operation of the utility.
`-install` *`path`*	Changes the installation that the instance uses. You can use the `-install` option to associate the instance with a different TimesTen installation. Typically, this is used to upgrade the instance to a new maintenance or patch release. This option is not used to upgrade to a new major release.
`-tnsadmin` *`path`*	Updates the instance's `TNS_ADMIN` setting (for cache).
`-port` *`number`*	Specifies a new daemon port number.
`-serverport` *`number`*	Specifies a new server port number.
`-systemd | -nosystemd`	You can use this option to change whether systemd is used to manage the instance's main daemon or not. This option cannot be used with Clusterware.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
`-serverEncryption` *`requirement`*	Determines whether encryption is accepted, rejected, requested, or required for a client/server connection. • `accepted`: Enable an encrypted session if required or requested by the client; use an unencrypted session otherwise. This is the default. • `rejected`: Demand an unencrypted session. (If the server does not support encryption, TimesTen behaves as if this is the setting on the server.) The connection is rejected if the client requires encryption. • `requested`: Request an encrypted session if the client allows it (if the client has any setting other than rejected); use an unencrypted session otherwise. • `required`: Demand an encrypted session. Reject the connection if the client rejects encryption. The `-serverEncryption` and `-serverCipherSuites` values are set at the instance creation level in the `timesten.conf` file and serve as the default values for any database created on that instance. However, you can override them by setting the corresponding connection attribute in the database definition. See Transport Layer Security for TimesTen Client/Server in *Oracle TimesTen In-Memory Database Security Guide* regarding usage of the `-serverEncryption` and `-serverCipherSuites` options.
`-serverCipherSuites` `{`*`suites`*`}`	Lists the cipher suite or cipher suites that can be used for Transport Layer Security, depending also on the client setting. There is no default setting. Specify one or more TLS version 1.3 or 1.2 cipher suites in a comma-separated list from strongest to weakest. To use TLS, the server and client settings must include at least one common suite. See Cipher Suites in TimesTen in *Oracle TimesTen In-Memory Database Security Guide* to learn more about the supported cipher suites and how TimesTen negotiates cipher suite selection. For more information on TLS for encrypted communication between clients and server, see Transport Layer Security for TimesTen Client/Server in *Oracle TimesTen In-Memory Database Security Guide*.
`-serverWallet` *`new_wallet_dir`*	If you move or copy the server wallet to a different location, you must specify this new location in the `timesten.conf` file. You can do this using `-serverWallet`, which determines if the new server wallet directory is valid, and if so, modifies the entry in the `timesten.conf` file. See Task 3: Configure the Server for TLS in the *Oracle TimesTen In-Memory Database Security Guide* for more information.
`-crs`	Creates or modifies the instance's Oracle Clusterware configuration.

Examples

This example modifies the encryption settings of an existing TimesTen instance by updating the cipher suite to TLS 1.2 `TLS_DHE_RSA_WITH_AES_128_GCM_SHA256` and configures the instance to accept the encrypted connections. It specifies a new server wallet location and warns that you may need to manually re-create the certificates to match the updated settings.

```
% ttInstanceModify -serverEncryption accepted -serverciphersuites
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 -serverWallet /scratch/myinstance/conf/test
INFO: Changing cipher suites. Make sure to manually re-create certificates if needed.
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

This example modifies the encryption settings of an existing TimesTen instance by updating the cipher suite to TLS 1.3 `TLS_AES_128_CCM_SHA256` and configures the instance to accept the encrypted connections.

```
% ttInstanceModify -serverEncryption accepted -serverciphersuites TLS_AES_128_CCM_SHA256
-serverWallet /scratch/myinstance/conf/test
INFO: Changing cipher suites. Make sure to manually re-create certificates if needed.
```

In this example, an attempt is made to alter the TLS 1.2 cipher suite with a TLS 1.3 `TLS_AES_256_GCM_SHA384` cipher suite that does not require re-creating the certificates.

```
% ttInstanceModify -serverciphersuites TLS_AES_256_GCM_SHA384
INFO: Changing cipher suites. Make sure to manually re-create certificates if needed.
```

ttIsql

Use `ttIsql` to run SQL statements and call TimesTen built-in procedures from this utility. Additionally, you can run SQL interactively from the command line and 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 a detailed description on running SQL from `ttIsql`, use the `-helpfull` option.

For more details on the `ttIsql` utility, see *Oracle TimesTen In-Memory Database ttIsql User's and Reference Guide*.

Required Privilege

This utility requires no privileges.

Syntax

```
ttIsql {-h | -help | -? | -helpcmds | - helpfull}
ttIsql {-V | -version}
ttIsql [-f inputFile] [-v verbosity] [-e commands | sql_statement]
[-interactive] [-N ncharEncoding] [-wait] [-restrict] {-connStr connStr | DSN}
```

Options

`ttIsql` has the options:

Option	Description
`-connStr` *`connStr`*	An ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.
`DSN`	Specifies an ODBC data source name of the database to be connected.
`-e` *`commands`*	Specifies a semicolon separated list of `ttIsql` commands to run on startup.
`-f` *`filename`*	Read SQL statements from *`filename`*.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description

`-h`
`-help`
`-?`
	Prints a usage message and exits.

`-helpcmds`	Prints a short list of the interactive commands.
`-helpfull`	Prints a full description of the interactive commands.
`-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.
`-restrict`	The following commands are blocked, and an error message is displayed: • `cachesqlget` • `connect` • `edit` • `host` • `putenv` • `savehistory` • `set signalbehavior 1` • `set signalbehavior waitfordebugger` • `spool` See Use ttIsql with Restrict Mode in the *Oracle TimesTen In-Memory Database Security Guide*.
`-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.

For more details, see Get Started with ttIsql in the *Oracle TimesTen In-Memory Database ttIsql User's Guide and Reference*.

Copyright © 1996, 2026, Oracle and/or its affiliates.

ttMigrate

The `ttMigrate` utility saves and restores TimesTen objects and migrates databases between different TimesTen releases.
You can perform these operations:

- Saves an object from a TimesTen database into the `ttMigrate` data file.
- Restores an object from the `ttMigrate` data file into a TimesTen database.
- Prints details about the contents of a `ttMigrate` data file.

Saved objects include:

- Tables
- Cache group definitions
- Views and materialized views
- Sequences
- Replication schemes
- Users and user information, including database privileges for each user.

You can use the `ttMigrate` utility to migrate databases from one major release of TimesTen Classic (such as Release 18.1) to another major release of TimesTen (for example, Release 26.1).

Since database checkpoint and transaction log files are not compatible between major releases, you need to use the `ttMigrate` utility to upgrade major release versions of TimesTen.

> **Note**
>
> *Users* and *user privileges* did not exist in TimesTen 7.0 and previous releases. Hence, these objects will not be available for restoration.

For more information about TimesTen databases migration, see Moving to a Different Major Release of TimesTen Classic in *Oracle TimesTen In-Memory Database Installation, Migration, and Upgrade Guide*.

The data files produced by this utility are platform-dependent. For example, a `ttMigrate` data file created on Linux can only be restored on Linux and not any other platform, for example AIX.

By default, the `ttMigrate` utility restores the database using one thread. During restoration, you can specify the `-numThreads` option to restore the data files using multiple threads, thus potentially improving performance.

The `ttMigrate` utility is supported for TimesTen server DSNs. For TimesTen client DSNs use the utility `ttMigrateCS` (client/server version).

Copyright © 1996, 2026, Oracle and/or its affiliates.

> **Note**
>
> Although cross-release compatibility over client/server protocol is supported in TimesTen, the tool `ttMigrateCS` is not backward and forward release compatible; hence it can be used only for the same version client/server connections.

Required Privilege

This utility requires various privileges depending on the options specified. The instance administrator, or `ADMIN`, has access to all the options.

Only the instance administrator can use the `-r` option. 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 cache operations, then `CACHE_MANAGER` is also required.

Only the instance administrator can use the `-c` option to capture *an entire database*. 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.

Syntax

```
ttMigrate {-h | -help | -?}
              {-V | -version}
```

To create or append a data file, use:

```
ttMigrate {-a | -c} [-v verbosity] [-nf] [-nr] [-fixNaN] [-saveAsCharset charset]
[-relaxedUpgrade | -exactUpgrade]
[-activeDML | -noActiveDML] [-restrict]
{DSN | -connStr connection_string} data file [[objectOwner.]objectName...]
```

To restore a database from a data file created by this utility, use:

```
ttMigrate -r  [-C ckptFreq] [-v level] [-nf] [-nr] [-fixNaN] [-numThreads n]
[-updateStats | -estimateStats percent] [-relaxedUpgrade | -exactUpgrade]
[-inline rule] [-noCharsetConversion] [-cacheUid uid [-cachePwd pwd]]
 [-autorefreshPaused] [-restorePublicPrivs] [-localhost host]
[-resizeHashIndexes]
{DSN | -connstr connection_string} dataFile [objectOwner.objectName...]
```

To list or display the contents of a data file created by this utility, use:

```
ttMigrate {-l | -L | -d | -D} dataFile [[objectOwner.]objectName...]
```

Options

The append (`-a`) or create (`-c`) modes, the list (`-l`/`-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:.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description

`-h`
`-help`
`-?`
	Prints a usage message and exits.

`-V | -version`	Prints the release number of `ttMigrate` and exits.
`-a`	Selects append mode: Appends data to a pre-existing data file, that was originally created using `ttMigrate -c`. See Create Mode (-c) and Append Mode (-a) for more details.
`-c`	Create mode: Creates an original data file. See Create Mode (-c) and Append Mode (-a) for more details.
`-v` *verbosity*	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 `-v` option in list, long-list, describe and long-describe modes.
`-nf`	Specifies that `ttMigrate` should not save or restore foreign key information when saving or restoring ordinary (non-cached) tables.
`-nr`	Specifies that `ttMigrate` should not save or restore table rows when saving or restoring ordinary (non-cached) tables.
`-fixNaN`	Converts all `NaN`, `Inf` and `-Inf` values found in objects to 0.0. This is useful for migrating data into releases of TimesTen that do not support the `NaN`, `Inf` and `-Inf` values.
`-saveAsCharset` *charset*	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.
`-relaxedUpgrade`	Save or restore the tables in a way that is compatible with a replication scheme that uses `TABLE DEFINITION CHECKING RELAXED`. `ttMigrate` ignores this option when the `-a` option is given. This option should not be used in combination with a replication scheme that uses `TABLE DEFINITION CHECKING EXACT`, or else replication may no longer work. The default is `-exactUpgrade`.
`-exactUpgrade`	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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
`-activeDML |-noActiveDML`	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.
`-restrict`	When the `-c` option is used with `-restrict`, it prevents `ttMigrate` from overwriting an existing file. See Use ttMigrate with Restrict Mode in the *Oracle TimesTen In-Memory Database Security Guide*.
DSN	Specifies an ODBC data source name of the database to be migrated.
`-connStr` *connection_string*	An ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.
owner	The owner of an object.
name	The name of the database object(s) to be saved or restored.
`-r`	Selects restore mode. Restores a database from a data file created by this utility. See Restore Mode (-r)for more details.
`-numThreads` *n*	Specifies the number of threads to use while restoring a database files. If unspecified, `ttMigrate` uses one thread to restore objects from the data file. Valid values are 1 through 32.
`-updateStats`	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 `-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.
`-estimateStats` *percent*	Specifies that `ttMigrate` should estimate statistics on restored tables and materialized views for the specified percentage of rows. Supported 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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
`-inline` *`rule`*	Indicates the rule to be used for converting variable-length columns to `INLINE` in restore mode. The value for rule is one of: `preserve` - `ttMigrate` preserves the original `INLINE` attribute of each column. This is the default, and it is required if `-exactUpgrade` is used. `dsDefault` - `ttMigrate` uses the database's default rule for setting the `INLINE` attribute of restored columns. *`maxlen`* - `ttMigrate` restores as `INLINE` all variable-length columns with length <= *`maxlen`* and restores as `NOT INLINE` all variable-length columns with length greater than `maxlen`. If *`maxlen`* is 0 then all columns are restored as `NOT INLINE`. `INLINE` variable-length columns cannot successfully be replicated to `NOT INLINE` columns.
`-cacheUid`	The cache administration user ID to use when restoring asynchronous writethrough cache groups and cache groups with the `AUTOREFRESH` attribute.
`-cachePwd`	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.
`-l`	Selects list mode. Lists the names of database objects in the specified data file. See List Mode (-l) and Long-list Mode (-L) for more details.
`-L`	Selects long-list mode. Lists the names of database objects in the specified data file and other details about the database objects. See List Mode (-l) and Long-List Mode (-L) for more details.
`-d`	Selects describe mode. Displays a short description of the objects in the data file. See Describe Mode (-d) for more details.
`-D`	Selects long-describe mode. Displays a full description of the objects in the data file. See Long-Describe Mode (-D) for more details.
`dataFile`	The path name of the data file to which objects are to be saved or from which objects are to be restored.

The following `ttMigrate` options are available in restore mode (`-r`) only:

Option	Description
`-autorefreshPaused`	Restores cache groups with `AUTOREFRESH` attribute with autorefresh state paused. Otherwise the state is set to `OFF`.
`-C` *`ckptFreq`*	Specifies that `ttMigrate` should checkpoint the database after restoring every *`ckptFreq`* megabytes of data. A value of zero (the default) specifies that `ttMigrate` should never checkpoint the database.
`-convertCGTypes`	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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
`-localhost` *`hostName`*	Explicitly identifies the name or IP address of the local host when restoring replicated tables.
`-noCharsetConversion`	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 `WE8ISO8895P1`, 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.
`-resizeHashIndexes`	Resizes user hash indexes during restore to be optimal size based on number of table rows.
`-restorePublicPrivs`	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`.

Modes

Create Mode (-c) and Append Mode (-a)

In create mode, the `ttMigrate` utility saves objects from a TimesTen database into a new binary data file. If the data file does not exist, the `ttMigrate` utility creates it. Otherwise, `ttMigrate` overwrites the existing file, destroying its contents.

The data file format used by the `ttMigrate` utility 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, the `ttMigrate` utility appends objects from a TimesTen database to an existing data file. If the data file does not exist, the `ttMigrate` utility creates it.

For each ordinary (non-cached) table, the `ttMigrate` utility saves the following:

- 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, the `ttMigrate` utility saves the following:

- 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. No row data is exported.

Copyright © 1996, 2026, Oracle and/or its affiliates.

- 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.

After the ttMigrate utility is 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).

For each view, the ttMigrate utility saves the following:

- All the same information as a typical table.
- The query defining the view.

For each sequence, the ttMigrate utility saves the following:

- The complete definition of the sequence.
- The sequence's current value.

For each user (except the instance administrator), the ttMigrate utility saves the following:

- User name.
- The user's encrypted password.
- Privileges that have been granted to the user.

For PUBLIC, the ttMigrate utility saves all privileges that have been granted to PUBLIC after database creation.

If there are any replication schemes defined, the ttMigrate utility 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, the ttMigrate utility 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). The ttMigrate utility 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 the ttMigrate utility issues an error.

Use the -v option to control the information that ttMigrate prints while the save is in progress.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Restore Mode (-r)

In restore mode, the `ttMigrate` utility restores all database objects from a data file into a TimesTen database.

For each ordinary (non-cached) table, the `ttMigrate` utility 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, the `ttMigrate` utility restores:

- The cache group definition, using the original cache group owner and name. No data is imported.
- 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.

Notes

- 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.
- All autorefresh cache groups in the restored database have the `AUTOREFRESH` state set to `OFF`, no matter how it was set on the source database. If preferred, you can set the `AUTOREFRESH` state of restored autorefresh cache groups to `PAUSED` instead of `OFF` using the `-autorefreshPaused` option.

The `-exactUpgrade` option is the default for both `-c` and `-r` options.

By default, the `ttMigrate` utility 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 the `ttMigrate` utility issues an error.

Use the `-v` option to control the information that the `ttMigrate` utility 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, `-inlinedsDefault` and `-inline`*`maxlen`*, `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 `-inline`*`maxlen`* mode restores variable length

Copyright © 1996, 2026, Oracle and/or its affiliates.

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, the ttMigrate utility lists the names of database objects in the specified data file, including cached tables and the replication scheme TTREP tables.

In long-list mode, the ttMigrate utility 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, the ttMigrate utility 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, the ttMigrate utility gives a short description for database objects in the specified file.

For each table, the ttMigrate utility 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, the ttMigrate utility also lists the query defining the view.

For cache groups, the ttMigrate utility 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, the ttMigrate utility lists the replication scheme name and all the TTREP replication scheme tables in the same manner as user tables.

By default, the ttMigrate utility 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, the ttMigrate utility 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, the ttMigrate utility 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, the ttMigrate utility lists all the values used to define the sequence and its current value.

For replication schemes, the ttMigrate utility lists all the TTREP replication scheme tables in the same manner as user tables.

Copyright © 1996, 2026, Oracle and/or its affiliates.

By default, the `ttMigrate` utility 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.

Pre-7.0 TimesTen Type	Oracle Type	Converted Type
TINYINT	NUMBER(p,s) when s > 0	NUMBER(p,s)
TINYINT	NUMBER(p,s) when s <= 0	TT_TINYINT
SMALLINT	NUMBER(p,s) when s > 0	NUMBER(p,s) TT_SMALLINT
SMALLINT	NUMBER(p,s) when s <= 0	TT_SMALLINT
INTEGER	NUMBER(p,s) when s > 0	NUMBER(p,s)
INTEGER	NUMBER(p,s) when s <= 0	TT_INTEGER
BIGINT	NUMBER(p,s) when s > 0	NUMBER(p,s)
BIGINT	NUMBER(p,s) when s <= 0	TT_BIGINT
NUMERIC(p,s)DECIMAL(p,s)	NUMBER	NUMBER
NUMERIC(p,s)DECIMAL(p,s)	NUMBER(x,y)	NUMBER(x,y)
NUMERIC(p,s)DECIMAL(p,s)	FLOAT(x)	NUMBER(p,s)
REAL	Any	BINARY_FLOAT
DOUBLE	Any	BINARY_DOUBLE
FLOAT(x) x <=24	Any	BINARY_FLOAT
FLOAT(x) x >= 24	Any	BINARY_DOUBLE
CHAR(x)	Any	ORA_CHAR(x)
VARCHAR(x)	Any	ORAVARCHAR2(x)
BINARY(x)	Any	TT_BINARY(x)
VARBINARY(x)	Any	TT_VARBINARY(x)
DATE	DATE	ORA_DATE
TIMESTAMP	DATE	ORA_DATE
TIME	DATE	ORA_DATE
Any1	TIMESTAMP(m)	ORA_TIMESTAMP(m)

> **Note**
>
> Any means the type value does not affect the converted result type.

Copyright © 1996, 2026, Oracle and/or its affiliates.

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 In-Memory Database Cache 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.

```
% ttMigrate -c SalesDS sales.ttm
```

This command appends all database objects in the `SalesDS` database owned by user `MARY` to `sales.ttm`:

```
% ttMigrate -a SalesDS sales.ttm MARY.%
```

This command restores all database objects from `sales.ttm` into the `SalesDS` database:

```
% ttMigrate -r SalesDS sales.ttm
```

This command restores `MARY.PENDING` and `MARY.COMPLETED` from `sales.ttm` into `SalesDS` (objects are case-insensitive):

```
% ttMigrate -r SalesDS sales.ttm MARY.PENDINGMARY.COMPLETED
```

This command lists all objects saved in `sales.ttm`:

```
% 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.

Consider the following restrictions, limitations, and suggestions before using the `ttMigrate` utility.

Cache groups: In restore mode, the presence of foreign key dependencies between tables may require the `ttMigrate` utility 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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

The connection attribute `PassThrough` with a nonzero value is not supported with this utility and returns an error.

Character sets: By default, the `ttMigrate` utility 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 the `ttMigrate` utility 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.

Replication: Before starting to migrate an entire set of replicated databases, 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 vies during migration.

Other considerations: You cannot use the `ttMigrate` utility to:

- Migrate databases between hardware platforms.
- Restore data saved with ttBackup or use ttBackup to restore data saved with `ttMigrate`.
- Migrate from one database release to a different `ttMigrate` release. The release of the `ttMigrate` utility must match the release of the database to which you are connecting.

It is recommended that you do not run DDL SQL commands via `ttIsql` or programmatically while running the `ttMigrate -r` operation to avoid lock contention issues for your application.

See Also

ttBackup
ttBulkCp
ttRestore

Copyright © 1996, 2026, Oracle and/or its affiliates.

ttRepAdmin

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.

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 -detail {-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 | -noSetMasterRepStart] [-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]
      [-open | -close]
      {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

Copyright © 1996, 2026, Oracle and/or its affiliates.

- 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}
```

Option	Description

`-h`
`-help`
`-?`
	Display help information.

`-V | -version`	Display TimesTen version information.

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` has the options:

Option	Description
DSN	Data source name of a master or subscriber database.
`-connStr` *connection_string*	Connection string of a master or subscriber database, an ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.
`-self`	Specified database.
`-list`	Lists database name, host, port number, and bookmark position.
`-scheme` [*owner*.]*schemeName*	Name of replication scheme when there is more than one scheme.

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]
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

```
[-list] [-scheme [owner.]schemeName]
{DSN | -connStr connection_string}
```

Options

`ttRepAdmin -receiver` has the options:

Option	Description
DSN	Data source name of the master database.
`-connStr` *connection_string*	Connection string of the master database, an ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.
`-receiver`	Subscriber databases receiving updates from the master. Use `-name` and `-host` to specify a specific subscriber database.
`-name` *receiverName*	A specific subscriber (receiving) database. The *receiverName* is the last component in the database path name.
`-host` *receiverHostName*	Host name or TCP/IP address of the subscriber host.
`-state start`	Sets the state of replication for the subscriber.
`-state stop`	`start` (default) - Starts replication to the subscriber.
`-state pause`	`stop` - Stops replication to the subscriber, discarding updates.
	`pause` - Pauses the replication agent, preserving updates.
	See Set the Replication State of Subscribers in *Oracle TimesTen In-Memory Database Replication Guide* for more information.
`-reset`	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.
`-list`	Lists information about a replication definition.
`-scheme` [*owner.*]*schemeName*]	Specifies the replication scheme name when there is more than one scheme.

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
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

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 run `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 | -noSetMasterRepStart] [-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]
      [-open | -close]
      {destDSN | -connStr connection_string}
```

Options

`ttRepAdmin -duplicate` has the options:

Option	Description
`-close`	Closes a database to user connections. When a database is closed to user connections, new connection attempts will fail, but existing connections are unaffected.
`-bandwidthmax` *`maxKbytesPerSec`*	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`.
`-compression 0 | 1`	Enables or disables compression during the duplicate operation. The default is `0` (disabled).

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
`-connStr` *connection_string*	Specifies the connection string of the destination database, an ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.
`-delXla`	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.
destDSN	Indicates the data source name of the destination database.
`-drop {[owner.]`*table* `... | [`*owner*`.]`*sequence* `|ALL`	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.
`-duplicate`	Creates a duplicate of the specified database using replication to transmit the database contents across the network. See Duplicating a Database in *Oracle TimesTen In-Memory Database Replication Guide*.
`-from` *srcDataStoreName*	Used with `-duplicate` to specify the name of the sender (or master) database. The *srcDataStoreName* is the last component in the database path name.
`-host` *srcDataStoreHost*	Defines the host name or `TCP/IP` address of the sender (or master) database.
`-initCacheDr [-cacheUid` cacheUid `-cachePwd` *cachePwd*`]`	Initializes disaster recovery. Must provide the cache admin user id and password. • *cacheUid* is the cache administration user ID. • *cachePwd* is the password for the cache administrator user. If no password is provided, `ttRepAdmin` prompts for a password.
`( -keepCG [-cacheUid` *cacheUid* `-cachePwd` *cachePwd*`] ([-recoveringNode | -deferCacheUpdate])) | -noKeepCG`	`-keepCG` and `-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`. • *cacheUid* is the cache administration user ID. • *cachePwd* is the password for the cache administrator user. If no password is provided, `ttRepAdmin` prompts for a password. If you cannot connect to the Oracle database or the Oracle database is down, then specify the `-recoveringNode` option when the `-duplicate` is being used to recover a failed node for a replication scheme that includes all AWT or incremental autorefresh cache groups. Otherwise, specify the `-deferCacheUpdate` option. These options defer changes to metadata on the Oracle database (that is used to manage AWT or incremental autorefresh cache groups) until after the duplicate operation completes, the cache and replication agents are started, and these agents can connect to the Oracle database. See Duplicating a Database in *Oracle TimesTen In-Memory Database Replication Guide* for more information.
`-localhost` *hostName*	Use with `-duplicate` and `-setMasterRepStart` to explicitly identify the name or IP address of the local host.
`-localIP` *localIPAddress*	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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
`-noDRTruncate`	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.
`-nThreads` *n*	Used with the `-initCacheDr` option, `-nThreads` indicates the number of threads used to truncate the Oracle database tables and push the data in the cache into Oracle during the initialization process.
`-open`	Opens a database to user connections. A database is open to user connections by default upon creation.
`-PWD` *pwd*	The password of the internal user specified in the `-UID` option.
`-PWDCrypt` *encryptedPwd*	The encrypted password of the user specified in the `-UID` option.
`-ramLoad`	Keeps the database in memory upon completion of the duplicate operation. This option avoids the /reload database cycle to improve the performance of the duplicate operation when copying large databases. After the duplicate option, RAM Policy for the database is set to `manual`. Use the ttAdmin utility to make further changes to the RAM policy.
`-remoteDaemonPort` *portNo*	The port number of the remote main daemon. The port number supplied as an argument to this option is used unless the value is zero. In that case the default behavior to determine the port number is used. The `-remoteDaemonPort` option cannot be used to duplicate databases that have stores which use automatic port configuration.
`-remoteIP` *remoteIPAddress*	Specifies the alias or IP (`IPv4` or `IPv6`) address of the remote or destination network interface to be used. If not specified, `ttRepAdmin` chooses any compatible interface.
`-setMasterRepStart`	This is on by default. When `-duplicate` option is set, the replication state for the newly created database is set to the `start` state just before the database is copied across the network. This ensures that all changes made to the source database after the duplicate operation starts are replicated to the newly duplicated target database. Any unnecessary transaction log files for the database are removed. Set the `-noSetMasterRepStart` option if you do not want this behavior.
`-noSetMasterRepStart`	Used with the `-duplicate` option. The replication state for the target database is not modified during the duplicate operation. Thus, if the state of the target database is set to `stopped`, or `failed`, then any changes made to the source database after the duplicate operation starts are not replicated to the newly duplicated target database.
`-truncate` [*owner.*]*table* `...|` `ALL`	Truncates any tables 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.
`-UID` *userid*	The user ID of a user having the `ADMIN` privilege on the source database must be supplied. This must be an internal user.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
`-verbosity {0 | 1 | 2}`	Provide details of the communication steps within the duplicate process and reports progress information about the duplicate transfer. `0` (default) - No diagnostics are returned. `1` - Reports details of the duplicate parameters to `stdout`. `2` - Reports details of the duplicate parameters and details of the duplicate transfer operation to `stdout`.

Examples

Duplicating a Database

On the source database, create a user and grant the `ADMIN` privilege to the user:

```
CREATE USER sampleuser IDENTIFIED BY sampleuser;
User created.

GRANT admin TO sampleuser;
```

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 sampleuser -PWD sampleuser
           -connStr "dsn=ds2;UID=sampleuser;PWD=sampleuser"
```

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 sampleuser -pwd sampleuser
        -keepCG -cacheuid orauser -cacheuid 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="`.

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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

```
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:

Option	Description
DSN	Indicates the data source name of the master database.
`-connStr` *connection_string*	Specifies the connection string of the master database, an ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.
`-wait`	Waits for replication to become current before continuing.
`-name` *receiverName*	Identifies the database. The database name is the last component in the database path name.
`-host` *receiverHostName*	Defines the host name or TCP/IP address of the subscriber host.
`-timeout` *seconds*	Specifies timeout value in seconds. `ttRepAdmin` returns within this amount of time, even if all updates to subscribers have not been completed.

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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

```
% 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 -detail {-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:

Option	Description
DSN	Indicates the data source name of the master database.
`-awtmoninfo`	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: • 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.
`-connStr` *connection_string*	Specifies the connection string of the master database, an ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.
`-detail`	Indicates whether the replication agent transmitters and receivers are using TLS (indicated as SSL)
`-log`	Prints out number and size of transaction log files retained by replication to transmit updates to other databases.
`-showconfig`	Lists the entire replication configuration. See Show the Configuration of Replicated Databases in *Oracle TimesTen In-Memory Database Replication Guide Oracle TimesTen In-Memory Database Replication Guide* for more information.
`-showstatus`	Reports the current status of the specified replicated database. See Use ttRepAdmin to Show Replication Status in *Oracle TimesTen In-Memory Database Replication Guide* for more information.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
`-bookmark`	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.

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.

See Use ttRepAdmin to Show Replication Status in *Oracle TimesTen In-Memory Database Replication Guide* for more information.

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
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

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.

For example, consider how to use the `ttRepAdmin -showstatus` utility to display status for a unidirectional replication scheme from the `rep1` database to the `rep2` database.

The first `ttRepAdmin -showstatus` output shows the status of the `rep1` database and its `TRANSMITTER` thread. The second output shows the status of the `rep2` database and its `RECEIVER` thread.

```
% ttRepAdmin -showstatus rep1

DSN                      : rep1
Process ID               : 1980
Replication Agent Policy : MANUAL
Host                     : MYHOST
RepListener Port         : 1113 (AUTO)
Last write LSN           : 0.1487928
Last LSN forced to disk  : 0.1487928
Replication hold LSN     : 0.1486640

Replication Peers:
  Name                   : rep2
  Host                   : MYHOST
  Port                   : 1154 (AUTO)
  Replication State      : STARTED
  Communication Protocol : 12

TRANSMITTER thread(s):
  For                     : rep2
    Start/Restart count   : 2
    Send LSN              : 0.1485960
    Transactions sent     : 3
    Total packets sent    : 10
    Tick packets sent     : 3
    MIN sent packet size  : 48
    MAX sent packet size  : 460
    AVG sent packet size  : 167
    Last packet sent at   : 17:41:05
    Total Packets received: 9
    MIN rcvd packet size  : 48
    MAX rcvd packet size  : 68
    AVG rcvd packet size  : 59
    Last packet rcvd'd at : 17:41:05
    Earlier errors (max 5):
    TT16060 in transmitter.c (line 3590) at 17:40:41 on 08-25-2004
    TT16122 in transmitter.c (line 2424) at 17:40:41 on 08-25-2004
```

The replication status for the `rep2` database should look similar to the following:

```
> ttRepAdmin -showstatus rep2

DSN                      : rep2
Process ID               : 2192
Replication Agent Policy : MANUAL
Host                     : MYHOST
RepListener Port         : 1154 (AUTO)
Last write LSN           : 0.416464
Last LSN forced to disk  : 0.416464
Replication hold LSN     : -1.-1
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

```
Replication Peers:
  Name              : rep1
  Host              : MYHOST
  Port              : 0 (AUTO)
  Replication State : STARTED
  Communication Protocol : 12

RECEIVER thread(s):
  For                   : rep1
  Start/Restart count   : 1
  Transactions received : 0
  Total packets sent    : 20
  Tick packets sent     : 0
  MIN sent packet size  : 48
  MAX sent packet size  : 68
  AVG sent packet size  : 66
  Last packet sent at   : 17:49:51
  Total Packets received: 20
  MIN rcvd packet size  : 48
  MAX rcvd packet size  : 125
  AVG rcvd packet size  : 52
  Last packet rcvd'd at : 17:49:51

% 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:

Copyright © 1996, 2026, Oracle and/or its affiliates.

```
% 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

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 Back Up, 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.

Syntax

```
ttRestore {-h | -help | -?}
ttRestore {-V | -version}
ttRestore [-fname filePrefix] [-noconn] -dir directory
       [-open | -close] {DSN | -connStr connection_string}
ttRestore -i [-noconn] [-open | -close] {DSN | -connStr connection_String}
```

Options

`ttRestore` has the options:

Option	Description
`-connStr` *`connection_string`*	An ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.
`-close`	Closes a database to user connections. When a database is closed to user connections, new connection attempts will fail, but existing connections are unaffected.
`DSN`	Specifies an ODBC data source name of the database to be administered.
`-dir` *`directory`*	Specifies the directory where the backup files are stored.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
`-fname` *`filePrefix`*	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.

`-h`
`-help`
`-?`
	Prints a usage message and exits.

`-i`	Read standard input for the backup data. You cannot use the `-dir` or `-fname` options with `-i`.
`-noconn`	To ensure that the restore was successful, `ttRestore` connects to the database as a last step. This option disables that last connect. *We recommend that you specify this option for best performance*. If this option is not specified, the database is loaded into memory and from memory.
`-open`	Opens a database to user connections. A database is open to user connections by default upon creation.
`-V | -version`	Prints the release number of `ttRestore` and exits.

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 21.1.1.1.0 and release 21.1.1.2.0 or 21.1.2.1.0. You cannot back up and restore files between release 11.2.2.8.0 and release 18.1.1.1.0, or between release 21.1.2.1.0 and release 21.2.1.1.0. You can use the `ttBulkCp` or `ttMigrateCS` (UNIX and Linux systems only) utility to migrate databases across major releases or operating systems.

You can 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 TimesTen Database with Cache Groups in the *Oracle TimesTen In-Memory Database Cache Guide* for details.

Copyright © 1996, 2026, Oracle and/or its affiliates.

See Also

ttBackup
ttBulkCp
ttMigrate

ttSchema

The `ttSchema` utility prints out the schema or selected objects of a database.
This utility can list the following schema objects that are found in SQL `CREATE` statements:

- Tables
- Indexes
- Cache group definitions
- Sequences
- Views
- Column definitions, including partition information
- PL/SQL program units
- Users and user information

> **Note**
>
> `ttSchema` does not export passwords.

You can control the level of detail in the listing and the objects listed through options. The output is determined by the privileges of the utility user and represents a point-in-time snapshot of the state of a database rather than a history of how it arrived 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 only the privileges needed to perform `DESCRIBE` operations on database objects.

This utility prints information only about objects owned by the utility user and those for which the user has `SELECT` privileges. If the utility user has the `ADMIN` privilege, `ttSchema` prints information about all objects.

Syntax

```
ttSchema {-h | -help | -?}
ttSchema {-V | -version}
ttSchema [-l] [-c] [-fixedTypes] [-st | -systemTables]
         [ -list {all | tables | views | sequences |
         cachegroups | repschemes | plsql | synonyms | userinfo} [,...] ]
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

```
[-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 -help -?	Prints a usage message and exits.
-l	One per-line listing of objects in the database.
-list {all | tables | views | sequences | cachegroups | repschemes | plsql | synonyms | userinfo}[,...]	This argument is a comma-delimited (no space after comma) list of object types that the user wants to include in the output. Default is -list all. This information is accessible only to users with ADMIN privilege. No user information is displayed if a user without ADMIN privilege runs -list all. If a user without ADMIN privilege runs -list userinfo, an error message stating that this information is unavailable to unprivileged users is returned.
[*owner*.]*object_name*	Limits the scope of the output to a specified pattern of database object(s). If the user running the utility has the ADMIN privilege, ttSchema prints information about all objects; otherwise, ttSchema prints only the information of the objects owned by the user. When the list of objects at the end of the command line is empty, ttSchema prints out all the objects of each type in the database, restricted only by the user's privileges.
-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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
-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	By default, the ttSchema utility only prints non-system objects. When used by a user with ADMIN privilege, -st prints out all the system-created objects, including system tables. If -st is omitted, system-created objects are omitted. When used by a user without ADMIN privilege, -st prints out no results unless the user specifies a system-created object, for example: ttSchema -st -dsn <DSN> SYS.%
-V | -version	Prints the release number of ttSchema and exits.

Impact on User Passwords

Suppose internal users are stored in the database and you use ttSchema to export the data and metadata out of your database. In that case the password associated with each user is not exported. Instead, ttSchema writes a CREATE USER statement for each internal user in the database and assigns this user a random password. The ACCOUNT LOCK and PASSWORD EXPIRE clauses of the CREATE USER statement are also used by ttSchema to lock each user's account and expire each user's password. If the caller of ttSchema imports this CREATE USER SQL into a new database or an existing database then the user's account is locked and the password is expired. The instance administrator must unlock the user's account and provide a new password. Additionally, if the user attempts to connect to the database with the original password or with the password created by ttSchema's CREATE USER statement, the connection fails with an account-locked error.

In this example, the instance administrator creates two users and uses ttSchema to export the data and metadata from the database into the schema.sql file.

```
Command> CREATE USER sampleuser identified by sampleuser;

User created.

Command> CREATE USER sampleuser2 identified by sampleuser2;

User created.

Command> GRANT CONNECT TO sampleuser,sampleuser2;
Command> exit
Disconnecting...
Done.
% ttSchema -DSN mydatabase > schema.sql
% cat schema.sql
-- Database is in Oracle type mode
create user SAMPLEUSER identified by 'DISABLED:m}Lc#MhP_Sh<X~M[p~z1' profile
"DEFAULT" password expire account lock;

grant CREATE SESSION to SAMPLEUSER;

create user SAMPLEUSER2 identified by 'DISABLED:fA~d2[XM/kjNp#?MYJ4E' profile
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

```
"DEFAULT" password expire account lock;

grant CREATE SESSION to SAMPLEUSER2;
```

The example drops the original users and then runs the `schema.sql` file in `ttIsql`. The connection fails when the user `sampleuser` attempts to connect to the database. The user's account is locked.

```
% ttIsql mydatabase

Copyright (c) 1996, 2021, Oracle and/or its affiliates. All rights reserved.
Type ? or "help" for help, type "exit" to quit ttIsql.



connect "DSN=access1";
Connection successful: DSN=access1;UID=instanceadmin;DataStore=/scratch/
sampleuser/mydatabase;
DatabaseCharacterSet=AL32UTF8;ConnectionCharacterSet=AL32UTF8;LogBufMB=1024;Pe
rmSize=500;TempSize=300;
(Default setting AutoCommit=1)
Command> DROP USER sampleuser;
User dropped.
Command> DROP USER sampleuser2;
User dropped.
Command> @schema
-- Database is in Oracle type mode
create user SAMPLEUSER identified by 'DISABLED:m}Lc#MhP_Sh<X~M[p~z1' profile
"DEFAULT" password expire account lock;

User created.

grant CREATE SESSION to SAMPLEUSER;
create user SAMPLEUSER2 identified by 'DISABLED:fA~d2[XM/kjNp#?MYJ4E' profile
"DEFAULT" password expire account lock;

User created.

grant CREATE SESSION to SAMPLEUSER2;

Command> connect adding "UID=sampleuser;PWD=sampleuser";
15179: the account is locked
The command failed.
none: Command> connect adding
"UID=sampleuser;PWD=DISABLED:m}Lc#MhP_Sh<X~M[p~z1";
15179: the account is locked
The command failed.
```

The instance administrator uses the `ALTER USER` statement to unlock the `sampleuser` account. The user `sampleuser` can then connect to the database to run `theschema.sql` file.

```
none: Command> use mydatabase
mydatabase: Command> ALTER USER sampleuser identified by sampleuser account
unlock;
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

```
User altered.

mydatabase: Command> connect adding "UID=sampleuser;PWD=sampleuser";
Connection successful: DSN=access1;UID=sampleuser;DataStore=/scratch/
sampleuser/mydatabase;
DatabaseCharacterSet=AL32UTF8;ConnectionCharacterSet=AL32UTF8;LogBufMB=1024;Pe
rmSize=500;TempSize=300;
(Default setting AutoCommit=1)
```

Examples

This example creates `sampleuser3` objects.

```
CREATE TABLE sampleuser3.customer (
cust_num            INTEGER NOT NULL PRIMARY KEY,
  region            CHAR(2) NOT NULL,
  name              VARCHAR2(80),
  address           VARCHAR2(255) NOT NULL);

CREATE SEQUENCE sampleuser3.custid MINVALUE 1 MAXVALUE 1000000;

CREATE TABLE sampleuser3.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 sampleuser3.customer (cust_num));

CREATE MATERIALIZED VIEW sampleuser3.order_summary AS
  SELECT cust.name, ord.ord_num, count(*) ord_count
  FROM sampleuser3.orders ord, sampleuser3.customer cust
  WHERE ord.cust_num = cust.cust_num
    GROUP BY cust.name, ord.ord_num;
```

This example returns the schema for the `orderdsn` database. The user `sampleuser` has the `ADMIN` privilege.

```
% ttSchema "DSN=orderdsn;UID=sampleuser;PWD=sampleuser";
-- Database is in Oracle type mode
create table SAMPLEUSER3.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 SAMPLEUSER3.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 SAMPLEUSER3.CUSTOMER (CUST_NUM));

create sequence SAMPLEUSER3.CUSTID
    increment by 1
    minvalue 1
    maxvalue 1000000
    start with 1
    cache 20;
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

```
create materialized view SAMPLEUSER3.ORDER_SUMMARY as
    SELECT CUST.NAME "NAME", ORD.ORD_NUM "ORD_NUM", COUNT(*) "ORD_COUNT"
    FROM SAMPLEUSER3.ORDERS ORD, SAMPLEUSER3.CUSTOMER CUST WHERE ORD.CUST_NUM =
    CUST.CUST_NUM GROUP BY CUST.NAME, ORD.ORD_NUM ;
```

Listing Specific Objects

This example returns only the materialized views and sequences for the `orderdsn` database.

```
% ttSchema -list views,sequences orderdsn
-- Database is in Oracle type mode
create sequence SAMPLEUSER3.CUSTID
    increment by 1
    minvalue 1
    maxvalue 1000000
    start with 1
    cache 20;

create materialized view SAMPLEUSER3.ORDER_SUMMARY as
    SELECT CUST.NAME "NAME", ORD.ORD_NUM "ORD_NUM", COUNT(*) "ORD_COUNT"
    FROM SAMPLEUSER3.ORDERS ORD, SAMPLEUSER3.CUSTOMER CUST WHERE ORD.CUST_NUM =
    CUST.CUST_NUM GROUP BY CUST.NAME, ORD.ORD_NUM ;
```

Specifying an Object

The following example returns the schema information for the `orders` table in the `orderdsn` database.

```
% ttSchema orderdsn sampleuser3.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 SAMPLEUSER3.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 SAMPLEUSER3.CUSTOMER (CUST_NUM));
```

Specifying Fixed Data Types

The following example returns the schema information for the `orderdsn` database using fixed data type names.

```
% ttSchema -fixedTypes orderdsn
-- Database is in Oracle type mode
create table SAMPLEUSER3.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 SAMPLEUSER3.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 SAMPLEUSER3.CUSTOMER (CUST_NUM));
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

```
create sequence SAMPLEUSER3.CUSTID
    increment by 1
    minvalue 1
    maxvalue 1000000
    start with 1
    cache 20;

create materialized view SAMPLEUSER3.ORDER_SUMMARY as
    SELECT CUST.NAME "NAME", ORD.ORD_NUM "ORD_NUM",
    COUNT(*) "ORD_COUNT" FROM SAMPLEUSER3.ORDERS ORD, SAMPLEUSER3.CUSTOMER CUST
    WHERE ORD.CUST_NUM = CUST.CUST_NUM
    GROUP BY CUST.NAME, ORD.ORD_NUM ;
```

Notes

- The generated SQL does not produce a history of transformations through `ALTER` statements nor preserve table partitions. However, 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`. Partition `1`, as printed by `ttSchema`, is secondary partition `1`, not the initial partition. For more details on partitions, see ALTER TABLE in *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.
- You should not run DDL SQL commands while running `ttSchema` to avoid lock contention issues for your application.

ttShmSize

The `ttShmSize` utility returns the required total shared memory size needed for both the database and the PL/SQL shared memory segments based on the connection string attributes specified by you.

Required Privilege

This utility requires no privileges.

Syntax

```
ttShmSize {-h | -help | -?}
ttShmSize {-V | -version}
ttShmSize [-v] {-connstr connStr
```

Options

`ttShmSize` has the options:

Option	Description
`-connStr` *`connStr`*	An ODBC connection string containing the name of the database, PermSize, TempSize, Connections and LogBufMB.
`-h -help -?`	Prints a usage message and exits.
`-V | -version`	Prints the release number of `ttShmSize` and exits.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
`-v`	Prints information on the parameters (and their values) used to calculate the required shared memory size.

> **Note**
>
> The name of the database used in the `-connStr` option must be available in either the system or user `.odbc.ini` file.

ttSize

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.

Syntax

```
ttSize {-h | -help | -?}
ttSize {-V | -version}
ttSize -tbl [owner.][tableName] [-rows rows] [- frac fraction]
     {-connStr connStr | DSN}
```

Options

`ttSize` has the options:

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
`-connStr` *connStr*	An ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.
DSN	Specifies the name of a data source to which `ttSize` should connect to retrieve table information.
`-frac` *frac*	Specifies the estimated average fraction of out-of-line VARCHAR or VARBINARY column sizes that will be used. If this option is omitted and the table contains out-of-line variable sized columns, a table scan is done to determine the average sizes. If the table is empty, the fraction is estimated to be 0.5 (50%) filled.

`-h`
`-help`
`-?`
	Prints a usage message and exits.

`-tbl [`*owner.*`][`*tableName*`]`	Specifies the name of the table whose definition should be used for size estimation. If the owner is omitted, the login name of the user is tried. If that is not found, the user SYS is used. `tableName` is an optional parameter. The output of `ttSize` is not very useful if you omit the `tableName` parameter.
`-rows` *rows*	Specifies the expected number of rows in the table. Space required to store a TimesTen table includes space for the actual data, plus overhead for bookkeeping, dynamic memory allocation and indexes. TimesTen may consume additional space due to memory fragmentation, temporary space allocated during query implementation and space to hold compiled SQL statements. If this option is omitted, the `ttSize` utility uses the number of rows in the existing table to estimate the table space, or uses 1 row if the table is empty.
`-V | -version`	Prints the release number of `ttSize` and exits.

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 run`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, run:

```
% 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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

- This utility is supported only for TimesTen Data Manager DSNs and not for TimesTen Client DSNs.

ttStats

The `ttStats` utility monitors database metrics (statistics, states, and other information) or takes and compares snapshots of metrics for SQL statements.

The `ttStatsConfig` built-in procedure configures the collection of metrics for SQL statements. See ttStatsConfig.

In TimesTen, the `ttStats` utility 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 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 TimesTen Client DSNs, use the `ttStatsCS` version of the utility.

> **Note**
>
> Although cross-release compatibility over client/server protocol is supported in TimesTen, the tool `ttStatsCS` is not backward and forward release compatible; hence it can be used only for the same version client/server connections.

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

Copyright © 1996, 2026, Oracle and/or its affiliates.

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.

Action	TimesTen Classic
Create a snapshot	Run the `ttStats -snapshot` utility to create a snapshot. The `-description` option for `-snapshot` is optional
Drop a snapshot	Run the `ttStats -drop -begin_snap` *snapid1* `-end_snap` *snapid2* utility to drop snapshots.
View information about a snapshot	Run the `ttStats -report` utility without any arguments. The `ttStats` utility prints the snapshot ID, date, time, capture level, and the description of all snapshots.
Generate a report between two snapshots	Run the `ttStats -report -snap1` *snapid1* `-snap2` *snapid2* utility. The utility generates a report between two snapshot IDs.

Do not use this utility if you are connecting to TimesTen through a driver manager.

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}
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

```
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:

Option	Description
`-h`	Prints the list of options and exits.
`-help`	**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.
`-V`	Prints the TimesTen release number and exits.
`-version`	
`-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`	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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
`-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.
`-report`	Generate a report from two specified snapshots, in HTML format by default. Use snapshots taken at the same capture level. **Notes**: • If you do not specify any snapshot IDs, a list of available snapshots (with date, time, capture level, and any notes) is displayed and you are prompted to enter each of the desired IDs. If you specify only one snapshot ID, you are told that you must enter two—reenter the command, specifying two snapshots. • This option is implemented by a call to the `GENERATE_REPORT_HTML` or (if the `-text` option is used) the `GENERATE_REPORT_TEXT` procedure of the `TT_STATS` PL/SQL package.
`-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.
`-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.
`-html | -text`	For report mode, specify HTML or plain text output format. **Note**: It is not necessary to specify `-html`. If you specify no format, the report is in HTML format by default.
`-drop`	Delete snapshots in the range specified by `-begin_snap` and `-end_snap` (inclusive) from the system. This option is useful to keep the snapshot storage under the limit of 255 snapshots in the database.
`-begin_snap` *`snapid1`*	For the `-drop` option, this specifies the snapshot ID at the beginning of the range of snapshots to delete.
`-end_snap` *`snapid2`*	For the `-drop` option, this specifies the snapshot ID at the end of the range of snapshots to delete. If `-end_snap` is not specified, then only the snapshot specified by `-begin_snap` is deleted.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
`-connStr` *`connection_string`* or *`DSN`*	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 database1

Connected to TimesTen Version 26.1.1.1.0 TimesTen Cache version 26.1.1.1.0

Waiting for 10 seconds for the next snapshot
Description                          Current  Rate/Sec  Notes
date.2021-Feb-20 16:49:25      -869676175380467200           1  sample #, not rate
connections.count                         12
db.size.temp_high_water_mark.kb         7153       7
lock.locks_granted.immediate             832       1
log.log_bytes_per_transaction              0
loghold.bookmark.log_force_lsn    0/12027904
loghold.bookmark.log_write_lsn    0/12050944
loghold.checkpoint_hold_lsn       0/12025856                 database1.ds0
loghold.checkpoint_hold_lsn       0/12023808                 database1.ds1
stmt.executes.count                       44       1
stmt.executes.selects                     32       1
```

> **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=database1;PLSQL_MEMORY_ADDRESS=20000000"
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

Snapshot Example

The following examples take two snapshots at the default typical level:

```
% ttStats -snapshot database1

Connected to TimesTen Version 26.01.0001.0001 TimesTen Cache version
26.1.1.1.0.
Snapshot 1 at TYPICAL level was successfully captured.

% ttStats -snapshot database1

Connected to TimesTen Version 26.1.1.1.0 TimesTen Cache version 26.1.1.1.0
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 database1

Connected to TimesTen Version 26.01.0001.0001 TimesTen Cache version 26.1.1.1.0.
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.

> **Note**
>
> Examples are not shown for SWT cache group metrics, local cache group metrics, dynamic global cache group 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
> ```

Figure 4-1 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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

- `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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Figure 4-1 ttStats report: summary

Summary

Info	Snap Id	Snap Time
Begin Snap:	3	2013-03-18 15:02:11.000000
End Snap:	4	2013-03-18 15:05:52.000000
Elapsed Time:		221 secs

Memory Usage and Connections

Metrics	Begin Value	End Value
connections.established.client_server	0	0
connections.established.count	20	20
connections.established.direct	20	20
connections.established.first.count	2	2
connections.established.threshold_exceeded	0	0
db.size.perm_allocated.kb	1048576	1048576
db.size.perm_high_water_mark.kb	183548	183746
db.size.perm_in_use.kb	183513	183711
db.size.temp_allocated.kb	524288	524288
db.size.temp_high_water_mark.kb	113468	115330
db.size.temp_in_use.kb	112616	112933

Load Profile

Metrics	Per Second	Per Transaction
Ckpt Size	1819680.4	62
Sorts	.2	0
Log Bytes	13030771.9	445
Log Buf Waits	0	0
Log Reads for Commit	0	0
Log Reads	0	0
Log Writes	41.9	0
Deletes	0	0
Inserts	6.3	0
Select	.9	0
Updates	0	0
Hard Parses	.2	0
Total Parses	.5	0
Durable Commit	.1	0
Non-durable Commit	29309.9	1
Rollback	0	0

Rows Per Read	0	Rows Per Write	0
Temp Indexes Created	0	Fast Path Log Buffer %	0

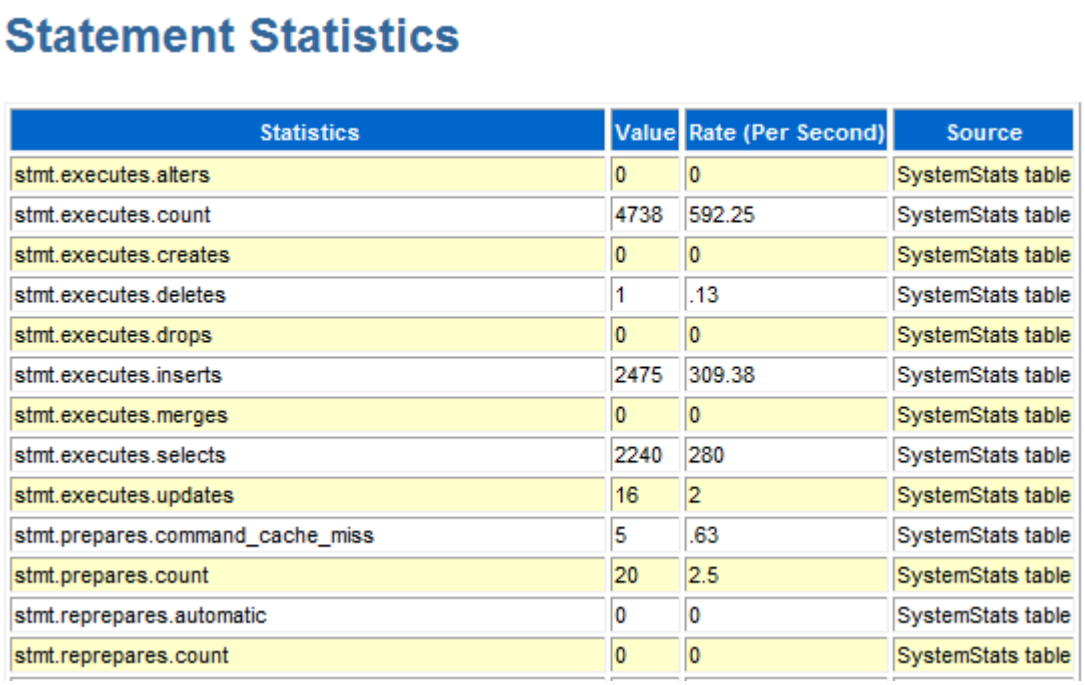
Instance Efficiency Percentage (Target 100%)

Command Cache Hit %	97.75	Non-Parse/Execs %	93.07
Lock Hit %	100	Log Buffer No Wait %	100

Copyright © 1996, 2026, Oracle and/or its affiliates.

Figure 4-2 shows statement metrics from a report. Both external metrics (`stmt.executes`, `stmt.prepares`, 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 4-2 ttStats report: statement statistics

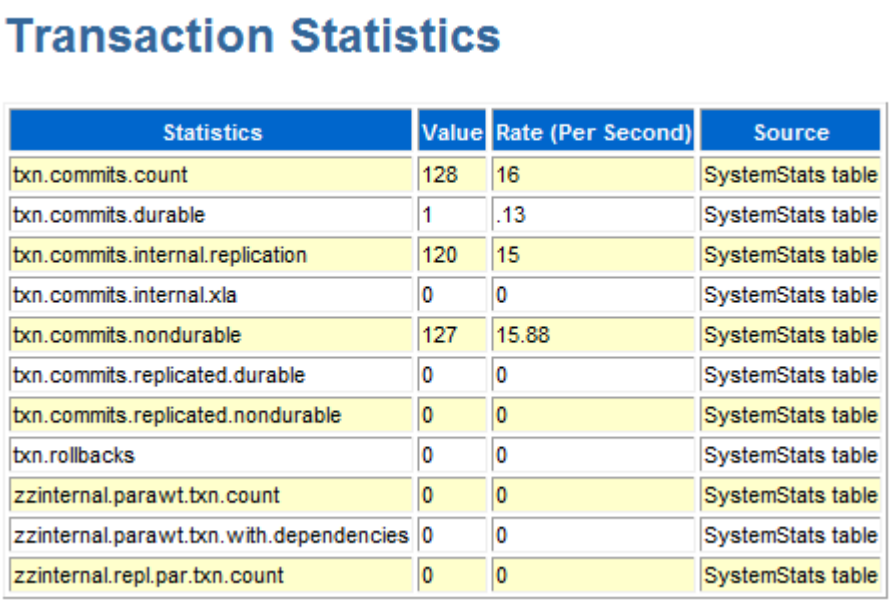


Statement Statistics

Statistics	Value	Rate (Per Second)	Source
stmt.executes.alters	0	0	SystemStats table
stmt.executes.count	4738	592.25	SystemStats table
stmt.executes.creates	0	0	SystemStats table
stmt.executes.deletes	1	.13	SystemStats table
stmt.executes.drops	0	0	SystemStats table
stmt.executes.inserts	2475	309.38	SystemStats table
stmt.executes.merges	0	0	SystemStats table
stmt.executes.selects	2240	280	SystemStats table
stmt.executes.updates	16	2	SystemStats table
stmt.prepares.command_cache_miss	5	.63	SystemStats table
stmt.prepares.count	20	2.5	SystemStats table
stmt.reprepares.automatic	0	0	SystemStats table
stmt.reprepares.count	0	0	SystemStats table

Figure 4-3 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 4-3 ttStats report: transaction statistics



Transaction Statistics

Statistics	Value	Rate (Per Second)	Source
txn.commits.count	128	16	SystemStats table
txn.commits.durable	1	.13	SystemStats table
txn.commits.internal.replication	120	15	SystemStats table
txn.commits.internal.xla	0	0	SystemStats table
txn.commits.nondurable	127	15.88	SystemStats table
txn.commits.replicated.durable	0	0	SystemStats table
txn.commits.replicated.nondurable	0	0	SystemStats table
txn.rollbacks	0	0	SystemStats table
zzinternal.parawt.txn.count	0	0	SystemStats table
zzinternal.parawt.txn.with.dependencies	0	0	SystemStats table
zzinternal.repl.par.txn.count	0	0	SystemStats table

Copyright © 1996, 2026, Oracle and/or its affiliates.

Figure 4-4 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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Figure 4-4 ttStats report: SQL execution statistics

SQL Statistics

SQL Sort by Execution Count

- **Only top 30 SQL Commands are displayed**

S.No	Executions	% Total	Cmd ID	Cmd Text	Source
1	278	24.98	10765485704	INSERT INTO SNAPSHOT_VALUE_CONFIG VALUES(:B3 , :B	ttSQLCmdCacheInfo
2	232	20.84	10764919976	INSERT INTO SNAPSHOT_VALUE_SQL VALUES(:B10 , :B1	ttSQLCmdCacheInfo
3	232	20.84	10764924680	call ttSqlExecutionTimeHistogram(:1)	ttSQLCmdCacheInfo
4	48	4.31	10751839192	SELECT TRIM(TB1.STAT_NAME), TB1.STAT_VALUE,TB2.STA	ttSQLCmdCacheInfo
5	20	1.8	10769328064	INSERT INTO SNAPSHOT_VALUE_PLSQL VALUES(:B3 , :B1	ttSQLCmdCacheInfo
6	16	1.44	10738871104	INSERT INTO SNAPSHOT_VALUE_CKPTHIST VALUES(:B8 ,	ttSQLCmdCacheInfo
7	10	.9	10738891504	call ttConfiguration('PLSQL')	ttSQLCmdCacheInfo
8	8	.72	10751740304	SELECT * FROM SNAPSHOT_INFO	ttSQLCmdCacheInfo
9	8	.72	10751754992	SELECT ABS(SYSDATE + (T2.TS - T1.TS) * 86400 - SYS	ttSQLCmdCacheInfo
10	8	.72	10751765552	SELECT T2.STAT_NAME, RPAD(TO_CHAR(T2.STAT_VALUE),1	ttSQLCmdCacheInfo
11	8	.72	10751782144	SELECT TO_NUMBER(T1.STAT_VALUE), TO_NUMBER(T2.STAT	ttSQLCmdCacheInfo
12	8	.72	10751792368	SELECT TRIM(TB1.STAT_NAME), TB1.STAT_VALUE, TB2.ST	ttSQLCmdCacheInfo
13	8	.72	10751795720	SELECT CASE WHEN T2.STAT_VALUE < T1.STAT_VALUE THE	ttSQLCmdCacheInfo
14	8	.72	10751799080	SELECT TRIM(T2.STAT_NAME), CASE WHEN T2.STAT_VALUE	ttSQLCmdCacheInfo
15	8	.72	10755119504	select count(*) from snapshot_value_cggroup	ttSQLCmdCacheInfo
16	8	.72	10755126624	select count(*) from snapshot_value_cggroup	ttSQLCmdCacheInfo
17	8	.72	10755166408	SELECT REPL_LOG_BEHIND, REPL_LOG_SEND_LFN, REPL_LO	ttSQLCmdCacheInfo
18	8	.72	10755168488	SELECT T1.STAT_NAME, CONCAT(CONCAT(TO_CHAR(T1.LFN)	ttSQLCmdCacheInfo
19	8	.72	10755170968	SELECT STARTTIME, ENDTIME, TRIM(CKPT_TYPE), TRIM(C	ttSQLCmdCacheInfo
20	8	.72	10755321784	select count(*) from snapshot_value_cggroup	ttSQLCmdCacheInfo
21	8	.72	10755324344	select count(*) from snapshot_value_cggroup wher	ttSQLCmdCacheInfo
22	8	.72	10764836336	SELECT REPL_LOG_BEHIND, REPL_LOG_SEND_LFN, REPL_LO	ttSQLCmdCacheInfo
23	8	.72	10764905560	select count(*) from snapshot_value_cggroup	ttSQLCmdCacheInfo
24	8	.72	10764959080	SELECT T1.STAT_NAME, COALESCE(T1.STAT_VALUE, '-'),	ttSQLCmdCacheInfo
25	8	.72	10764967448	SELECT TRIM(TB1.STAT_NAME), TB1.STAT_VALUE, TB2.ST	ttSQLCmdCacheInfo
26	4	.36	10765518624	SELECT DESC_ID FROM SNAPSHOT_DESCRIPTION WHERE TRI	ttSQLCmdCacheInfo
27	4	.36	10765523848	INSERT INTO SNAPSHOT_VALUE_LOGHOLD VALUES(:B5 , :B	ttSQLCmdCacheInfo
28	2	.18	10755307960	select LOG_BYTES_TO_LOG_BUFFER, LOG_FS_READS, LOG_	ttSQLCmdCacheInfo
29	2	.18	10764797296	SELECT PARA_VALUE*1024*1024 FROM TT_STATS_PARAM WH	ttSQLCmdCacheInfo
30	2	.18	10764802112	call ttConfiguration('TTGridEnable')	ttSQLCmdCacheInfo

Figure 4-5 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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Figure 4-5 ttStats report: SQL preparation statistics

SQL Sort by Preparation Count

- **Only top 30 SQL Commands are displayed**

S.No	Preparations	% Total	Cmd ID	Cmd Text	Source
1	278	24.39	10765485704	INSERT INTO SNAPSHOT_VALUE_CONFIG VALUES(:B3 , :B	ttSQLCmdCacheInfo
2	232	20.35	10764919976	INSERT INTO SNAPSHOT_VALUE_SQL VALUES(:B10 , :B1	ttSQLCmdCacheInfo
3	232	20.35	10764924680	call ttSqlExecutionTimeHistogram(:1)	ttSQLCmdCacheInfo
4	48	4.21	10751839192	SELECT TRIM(TB1.STAT_NAME), TB1.STAT_VALUE,TB2.STA	ttSQLCmdCacheInfo
5	20	1.75	10769328064	INSERT INTO SNAPSHOT_VALUE_PLSQL VALUES(:B3 , :B1	ttSQLCmdCacheInfo
6	16	1.4	10738871104	INSERT INTO SNAPSHOT_VALUE_CKPTHIST VALUES(:B8 ,	ttSQLCmdCacheInfo
7	10	.88	10738891504	call ttConfiguration('PLSQL')	ttSQLCmdCacheInfo
8	8	.7	10751740304	SELECT * FROM SNAPSHOT_INFO	ttSQLCmdCacheInfo
9	8	.7	10751754992	SELECT ABS(SYSDATE + (T2.TS - T1.TS) * 86400 - SYS	ttSQLCmdCacheInfo
10	8	.7	10751765552	SELECT T2.STAT_NAME, RPAD(TO_CHAR(T2.STAT_VALUE),1	ttSQLCmdCacheInfo
11	8	.7	10751782144	SELECT TO_NUMBER(T1.STAT_VALUE), TO_NUMBER(T2.STAT	ttSQLCmdCacheInfo
12	8	.7	10751792368	SELECT TRIM(TB1.STAT_NAME), TB1.STAT_VALUE, TB2.ST	ttSQLCmdCacheInfo
13	8	.7	10751795720	SELECT CASE WHEN T2.STAT_VALUE < T1.STAT_VALUE THE	ttSQLCmdCacheInfo
14	8	.7	10751799080	SELECT TRIM(T2.STAT_NAME), CASE WHEN T2.STAT_VALUE	ttSQLCmdCacheInfo
15	8	.7	10751851056	begin tt_stats.generate_report_html(:id1, :id2, :o	ttSQLCmdCacheInfo
16	8	.7	10755119504	select count(*) from snapshot_value_cggroup	ttSQLCmdCacheInfo
17	8	.7	10755126624	select count(*) from snapshot_value_cggroup	ttSQLCmdCacheInfo
18	8	.7	10755166408	SELECT REPL_LOG_BEHIND, REPL_LOG_SEND_LFN, REPL_LO	ttSQLCmdCacheInfo
19	8	.7	10755168488	SELECT T1.STAT_NAME, CONCAT(CONCAT(TO_CHAR(T1.LFN)	ttSQLCmdCacheInfo
20	8	.7	10755170968	SELECT STARTTIME, ENDTIME, TRIM(CKPT_TYPE), TRIM(C	ttSQLCmdCacheInfo
21	8	.7	10755321784	select count(*) from snapshot_value_cggroup	ttSQLCmdCacheInfo
22	8	.7	10755324344	select count(*) from snapshot_value_cggroup wher	ttSQLCmdCacheInfo
23	8	.7	10764836336	SELECT REPL_LOG_BEHIND, REPL_LOG_SEND_LFN, REPL_LO	ttSQLCmdCacheInfo
24	8	.7	10764905560	select count(*) from snapshot_value_cggroup	ttSQLCmdCacheInfo
25	8	.7	10764959080	SELECT T1.STAT_NAME, COALESCE(T1.STAT_VALUE, '-'),	ttSQLCmdCacheInfo
26	8	.7	10764967448	SELECT TRIM(TB1.STAT_NAME), TB1.STAT_VALUE, TB2.ST	ttSQLCmdCacheInfo
27	4	.35	10738881336	insert /*+ TT_CommitDMLOnSuccess(0) */ into vpn_us	ttSQLCmdCacheInfo
28	4	.35	10764986832	delete /*+ TT_CommitDMLOnSuccess(0) */ from vpn_us	ttSQLCmdCacheInfo
29	4	.35	10765367560	COMMIT	ttSQLCmdCacheInfo
30	4	.35	10765518624	SELECT DESC_ID FROM SNAPSHOT_DESCRIPTION WHERE TRI	ttSQLCmdCacheInfo

Figure 4-6 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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Figure 4-6 ttStats report: SQL command texts

Top SQL Command Texts

SQL ID	SQL Text
228847168	select owner#,name,namespace,obj#,type#,ctime,mtime,stime,status,flags from sys.obj$ where owner#=:1 and name=:2 and namespace=:3
230064776	COMMIT
230873952	INSERT INTO SNAPSHOT_DESCRIPTION VALUES(:B2 , :B1)
230938584	SELECT DESC_ID FROM SNAPSHOT_DESCRIPTION WHERE TRIM(DESC_NAME) = TRIM(:B1)
230939608	SELECT MAX(DESC_ID) FROM SNAPSHOT_DESCRIPTION
157941728	select owner,name from sys.syn$ where obj#=:1
157944352	select owner#,name,namespace,p_timestamp,p_obj#,nvl(property,0),type# from sys.dependency$ d, sys.obj$ o where d_obj#=:1 and p_obj#=obj#(+) order by order#
157952248	select order#,columns,types from sys.access$ where d_obj#=:1
157955200	call tt_stats.capture_snapshot
157956224	select piece#,length,piece from sys.idl_sb4$ where obj#=:1 and part=:2 and version=:3 order by piece#
157960344	select piece#,length,piece from sys.idl_ub1$ where obj#=:1 and part=:2 and version=:3 order by piece#
157964328	select piece#,length,piece from sys.idl_char$ where obj#=:1 and part=:2 and version=:3 order by piece#
157968312	select piece#,length,piece from sys.idl_ub2$ where obj#=:1 and part=:2 and version=:3 order by piece#
230203424	SELECT MAX(REPLPEER_ID) FROM SNAPSHOT_REPL_PEER
230207168	INSERT INTO SNAPSHOT_REPL_PEER VALUES(:B2 , :B1)
230333008	INSERT INTO SNAPSHOT_VALUE_PARAWT VALUES(:B4 , :B1 , :B2 , :B3)
230336568	call ttLatchStatsGet('show')

Figure 4-7 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 4-7 ttStats report: PL/SQL memory statistics

PL/SQL Memory Statistics

Statistics	Begin Value	End Value	Rate (Per Sec)	Source
CurrentConnectionMemory	1	1	0	ttPLSQLMemoryStats
DeferredCleanups	0	0	0	ttPLSQLMemoryStats
GetHitRatio	.64	.65	-	ttPLSQLMemoryStats
GetHits	196	206	1.25	ttPLSQLMemoryStats
Gets	305	316	1.38	ttPLSQLMemoryStats
Invalidations	0	0	0	ttPLSQLMemoryStats
PinHitRatio	.53	.54	-	ttPLSQLMemoryStats
PinHits	304	314	1.25	ttPLSQLMemoryStats
Pins	569	581	1.5	ttPLSQLMemoryStats
Reloads	25	26	.13	ttPLSQLMemoryStats

Figure 4-8 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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Figure 4-8 ttStats report: replication statistics

Replication Statistics

Repl_Log_Send_LSN		Repl_Log_Behind		Repl_RPS		Repl_TPS		Repl_Latency		Repl_Peer
Begin Value	End Value	Begin Value	End Value	Value	Rate (Per Sec)	Value	Rate (Per Sec)	Begin Value	End Value	
0/17687464	0/18578344	0	0	-1	-1	-1	-1	-1	-1	REPLDB2:0
0/17687464	0/17687464	0	0	-1	-1	-1	-1	-1	-1	REPLDB2:1
0/17687464	0/17687464	0	0	-1	-1	-1	-1	-1	-1	REPLDB2:2
0/17687464	0/17687464	0	0	-1	-1	-1	-1	-1	-1	REPLDB2:3
0/17687464	0/18578344	0	0	-1	-1	-1	-1	-1	-1	SUB1:0
0/17654696	0/18578344	0	0	-1	-1	-1	-1	-1	-1	SUB1:1
0/17654696	0/18578344	0	0	-1	-1	-1	-1	-1	-1	SUB1:2
0/17654696	0/18578344	0	0	-1	-1	-1	-1	-1	-1	SUB1:3
0/17687464	0/18578344	0	0	-1	-1	-1	-1	-1	-1	SUB2:0
0/17671080	0/18578344	0	0	-1	-1	-1	-1	-1	-1	SUB2:1
0/17671080	0/18578344	0	0	-1	-1	-1	-1	-1	-1	SUB2:2
0/17671080	0/18578344	0	0	-1	-1	-1	-1	-1	-1	SUB2:3

Figure 4-9 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.

Figure 4-9 ttStats report: parallel replication/AWT statistics

Parallel Replication/AWT Statistics

Statistics	MIN Value	MIN Rate	MAX Value	MAX Rate	AVG Value	AVG Rate	SUM Value	SUM Rate	Repl_Peer	Source
zzinternal.log.trackswitchin.busy	0	0	0	0	0	0	0	0	REPLDB1	v$repstats view
zzinternal.log.trackswitchin.busy	0	0	0	0	0	0	0	0	REPLDB2	v$repstats view
zzinternal.log.trackswitchin.busy	0	0	0	0	0	0	0	0	SUB1	v$repstats view
zzinternal.log.trackswitchin.busy	0	0	0	0	0	0	0	0	SUB2	v$repstats view
zzinternal.log.trackswitchin.timeout	0	0	1	.13	.25	.03	1	.13	REPLDB1	v$repstats view
zzinternal.log.trackswitchin.timeout	0	0	0	0	0	0	0	0	REPLDB2	v$repstats view
zzinternal.log.trackswitchin.timeout	0	0	0	0	0	0	0	0	SUB1	v$repstats view
zzinternal.log.trackswitchin.timeout	0	0	0	0	0	0	0	0	SUB2	v$repstats view
zzinternal.log.trackswitchout.busy	0	0	0	0	0	0	0	0	REPLDB1	v$repstats view
zzinternal.log.trackswitchout.busy	0	0	0	0	0	0	0	0	REPLDB2	v$repstats view
zzinternal.log.trackswitchout.busy	0	0	0	0	0	0	0	0	SUB1	v$repstats view
zzinternal.log.trackswitchout.busy	0	0	0	0	0	0	0	0	SUB2	v$repstats view
zzinternal.log.trackswitchout.timeout	0	0	1	.13	.25	.03	1	.13	REPLDB1	v$repstats view
zzinternal.log.trackswitchout.timeout	0	0	0	0	0	0	0	0	REPLDB2	v$repstats view
zzinternal.log.trackswitchout.timeout	0	0	0	0	0	0	0	0	SUB1	v$repstats view
zzinternal.log.trackswitchout.timeout	0	0	0	0	0	0	0	0	SUB2	v$repstats view

Figure 4-10 shows log metrics 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

Copyright © 1996, 2026, Oracle and/or its affiliates.

insufficient, writing to disk is taking too long, or the log flusher is CPU-bound. (Also see Managing Transaction Log Buffers and Files and Configure Log Buffer and Log File Size Parameters in *Oracle TimesTen In-Memory Database Operations Guide*.)

Figure 4-10 ttStats report: log statistics

Log Statistics

- **Numbers in log.file.earliest and log.file.latest represent values in begin snapshot and end snapshot**

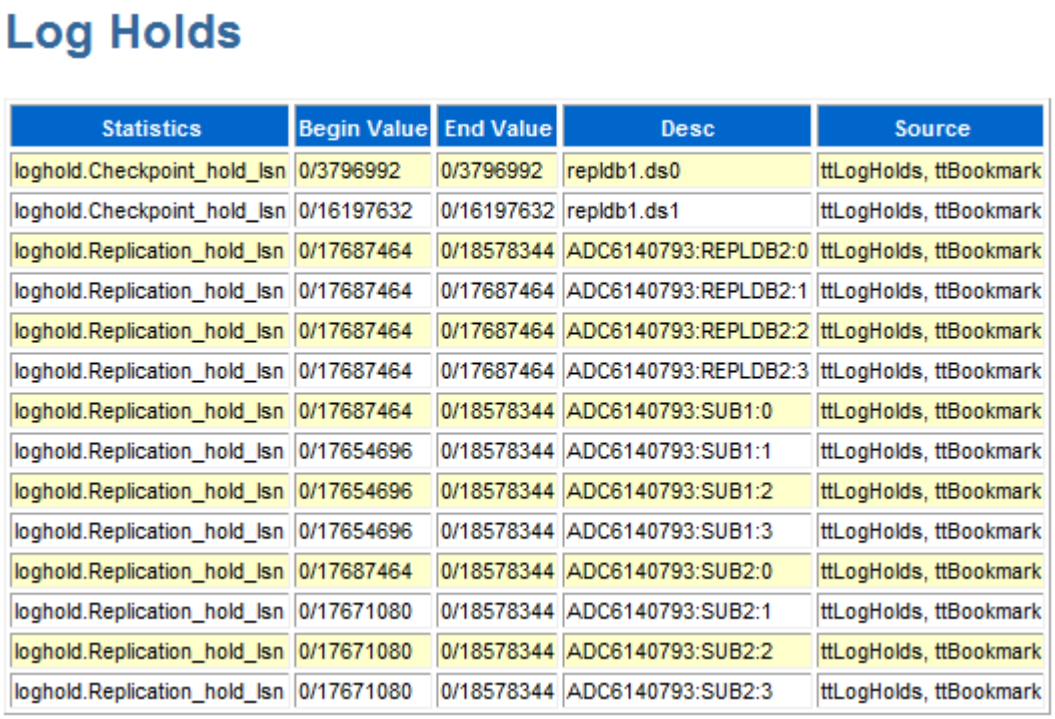
Statistics	Value	Rate (Per Second)	Source
log.buffer.bytes_inserted	780960	97620	SystemStats table
log.buffer.insertions	6474	809.25	SystemStats table
log.buffer.waits	0	0	SystemStats table
log.commit.bytes.read	460168	57521	SystemStats table
log.commit.file.reads	0	0	SystemStats table
log.file.earliest	0	0	SystemStats table
log.file.latest	0	0	SystemStats table
log.file.reads	0	0	SystemStats table
log.file.writes	17	2.13	SystemStats table
log.files.generated	0	0	SystemStats table
log.forces	11	1.38	SystemStats table
log.last_log_lfn	0	0	SystemStats table
log.log_bytes_per_transaction	87	10.88	SystemStats table
log.recovery.bytes.read	0	0	SystemStats table
zzinternal.log.buffer.bytes_inserted.fast_path	0	0	SystemStats table
zzinternal.log.buffer.insertions.fast_path	0	0	SystemStats table
zzinternal.log.strand_switches.insertion_latch_held	0	0	SystemStats table
zzinternal.log.strand_switches.strand_full	0	0	SystemStats table
zzinternal.repl.transmitter.log.wait_sleeps	0	0	SystemStats table
zzinternal.repl.transmitter.log.waits	0	0	SystemStats table

Figure 4-11 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.)

Copyright © 1996, 2026, Oracle and/or its affiliates.

Figure 4-11 ttStats report: log holds

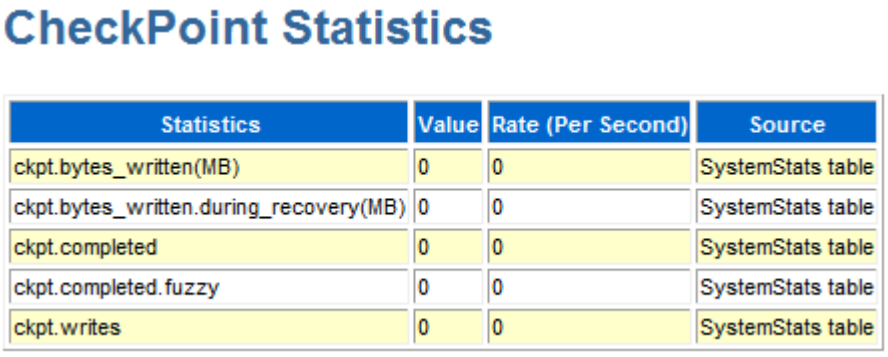


Log Holds

Statistics	Begin Value	End Value	Desc	Source
loghold.Checkpoint_hold_lsn	0/3796992	0/3796992	repldb1.ds0	ttLogHolds, ttBookmark
loghold.Checkpoint_hold_lsn	0/16197632	0/16197632	repldb1.ds1	ttLogHolds, ttBookmark
loghold.Replication_hold_lsn	0/17687464	0/18578344	ADC6140793:REPLDB2:0	ttLogHolds, ttBookmark
loghold.Replication_hold_lsn	0/17687464	0/17687464	ADC6140793:REPLDB2:1	ttLogHolds, ttBookmark
loghold.Replication_hold_lsn	0/17687464	0/17687464	ADC6140793:REPLDB2:2	ttLogHolds, ttBookmark
loghold.Replication_hold_lsn	0/17687464	0/17687464	ADC6140793:REPLDB2:3	ttLogHolds, ttBookmark
loghold.Replication_hold_lsn	0/17687464	0/18578344	ADC6140793:SUB1:0	ttLogHolds, ttBookmark
loghold.Replication_hold_lsn	0/17654696	0/18578344	ADC6140793:SUB1:1	ttLogHolds, ttBookmark
loghold.Replication_hold_lsn	0/17654696	0/18578344	ADC6140793:SUB1:2	ttLogHolds, ttBookmark
loghold.Replication_hold_lsn	0/17654696	0/18578344	ADC6140793:SUB1:3	ttLogHolds, ttBookmark
loghold.Replication_hold_lsn	0/17687464	0/18578344	ADC6140793:SUB2:0	ttLogHolds, ttBookmark
loghold.Replication_hold_lsn	0/17671080	0/18578344	ADC6140793:SUB2:1	ttLogHolds, ttBookmark
loghold.Replication_hold_lsn	0/17671080	0/18578344	ADC6140793:SUB2:2	ttLogHolds, ttBookmark
loghold.Replication_hold_lsn	0/17671080	0/18578344	ADC6140793:SUB2:3	ttLogHolds, ttBookmark

Figure 4-12 shows checkpoint metrics from a report.

Figure 4-12 ttStats report: checkpoint statistics



CheckPoint Statistics

Statistics	Value	Rate (Per Second)	Source
ckpt.bytes_written(MB)	0	0	SystemStats table
ckpt.bytes_written.during_recovery(MB)	0	0	SystemStats table
ckpt.completed	0	0	SystemStats table
ckpt.completed.fuzzy	0	0	SystemStats table
ckpt.writes	0	0	SystemStats table

Figure 4-13 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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Figure 4-13 ttStats report: AWT cache group statistics

Cache Group Statistics

AWT Cache Group Statistics

Statistics	Value	Rate(Per Sec)	Rate(Per AWT Txn)	Source
cg.awt.calls_to_oracle	330	15.71	.01	sys.SystemStats table
cg.awt.commits_on_oracle	330	15.71	.01	sys.SystemStats table
cg.awt.plsql_mode.batches	330	15.71	.01	sys.SystemStats table
cg.awt.plsql_mode.bytes	4587698	218461.81	98.6	sys.SystemStats table
cg.awt.plsql_mode.deletes.rows	0	0	0	sys.SystemStats table
cg.awt.plsql_mode.inserts.rows	0	0	0	sys.SystemStats table
cg.awt.plsql_mode.updates.rows	46530	2215.71	1	sys.SystemStats table
cg.awt.retries_on_oracle	0	0	0	sys.SystemStats table
cg.awt.rollbacks_on_oracle	0	0	0	sys.SystemStats table
cg.awt.sql_mode.batches	0	0	0	sys.SystemStats table
cg.awt.sql_mode.bytes	0	0	0	sys.SystemStats table
cg.awt.sql_mode.deletes.batches	0	0	0	sys.SystemStats table
cg.awt.sql_mode.deletes.rows	0	0	0	sys.SystemStats table
cg.awt.sql_mode.inserts.batches	0	0	0	sys.SystemStats table
cg.awt.sql_mode.inserts.rows	0	0	0	sys.SystemStats table
cg.awt.sql_mode.updates.batches	0	0	0	sys.SystemStats table
cg.awt.sql_mode.updates.rows	0	0	0	sys.SystemStats table
cg.awt.tt_txns	46530	2215.71	1	sys.SystemStats table
zzinternal.cg.awt.plsql_mode.exec_time	0	0	0	sys.SystemStats table
zzinternal.cg.awt.rxbatches	0	0	0	sys.SystemStats table
zzinternal.cg.awt.rxskips	0	0	0	sys.SystemStats table
zzinternal.cg.awt.sql_mode.exec_time	0	0	0	sys.SystemStats table
zzinternal.cg.awt.tt_proc_time	0	0	0	sys.SystemStats table

Figure 4-14 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).

Copyright © 1996, 2026, Oracle and/or its affiliates.

Figure 4-14 ttStats report: auto-refresh cache group statistics

Auto-refresh Cache Group Stats

Statistics	Value	Rate(Per Sec)	Rate(Per Cycle)	Source
cg.autorefresh.cycles.completed	5	.24	1	sys.SystemStats table
cg.autorefresh.cycles.failed	0	0	0	sys.SystemStats table
cg.autorefresh.deletes.rows	0	0	0	sys.SystemStats table
cg.autorefresh.full_refreshes	5	.24	1	sys.SystemStats table
cg.autorefresh.inserts.rows	0	0	0	sys.SystemStats table
cg.autorefresh.updates.rows	0	0	0	sys.SystemStats table
zzinternal.cg.autorefresh.bookmark_cycles	0	0	0	sys.SystemStats table
zzinternal.cg.autorefresh.bookmark_updates	0	0	0	sys.SystemStats table
zzinternal.cg.autorefresh.garbage_collector_cycles	0	0	0	sys.SystemStats table
zzinternal.cg.autorefresh.log_table.rows.garbage_collected	0	0	0	sys.SystemStats table
zzinternal.cg.autorefresh.log_table.rows.marked	0	0	0	sys.SystemStats table
zzinternal.cg.autorefresh.marker_cycles	0	0	0	sys.SystemStats table

Figure 4-15 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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Figure 4-15 ttStats report: database activity statistics

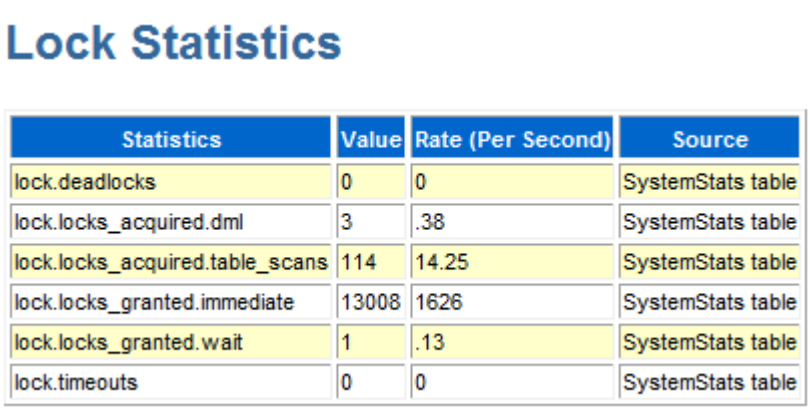
DB Activity Statistics

Statistics	Value	Rate (Per Second)	Source
db.cache_hits	0	0	SystemStats table
db.index.hash.deletes	0	0	SystemStats table
db.index.hash.inserts	0	0	SystemStats table
db.index.hash.inserts.recovery_rebuild	0	0	SystemStats table
db.index.hash.rows_fetched.count	0	0	SystemStats table
db.index.hash.rows_fetched.repl	0	0	SystemStats table
db.index.hash.scans.count	0	0	SystemStats table
db.index.hash.scans.repl	0	0	SystemStats table
db.index.range.deletes	0	0	SystemStats table
db.index.range.inserts.count	0	0	SystemStats table
db.index.range.inserts.recovery_rebuild	0	0	SystemStats table
db.index.range.rows_fetched.count	0	0	SystemStats table
db.index.range.rows_fetched.repl	0	0	SystemStats table
db.index.range.scans.count	0	0	SystemStats table
db.index.range.scans.repl	0	0	SystemStats table
db.index.range.updates	0	0	SystemStats table
db.index.rebuilds	0	0	SystemStats table
db.index.temporary.created	0	0	SystemStats table
db.index.temporary.rows_fetched.count	0	0	SystemStats table
db.index.temporary.rows_fetched.repl	0	0	SystemStats table
db.index.temporary.scans.count	0	0	SystemStats table
db.index.temporary.scans.repl	0	0	SystemStats table
db.joins.merge	0	0	SystemStats table
db.joins.nested_loop	28	3.5	SystemStats table
db.passthroughs	0	0	SystemStats table
db.size.perm_allocated.kb	0	0	SystemStats table
db.size.perm_high_water_mark.kb	595	74.38	SystemStats table
db.size.perm_in_use.kb	595	74.38	SystemStats table
db.size.temp_allocated.kb	0	0	SystemStats table
db.size.temp_high_water_mark.kb	30	3.75	SystemStats table
db.size.temp_in_use.kb	886	110.75	SystemStats table
db.sorts	4	.5	SystemStats table
db.table.full_scans	17	2.13	SystemStats table

Figure 4-16 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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Figure 4-16 ttStats report: lock statistics

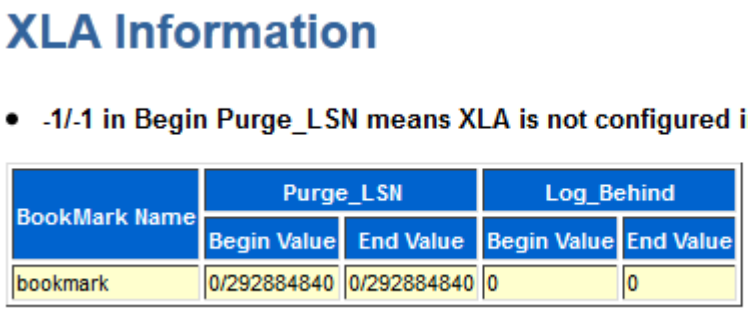


Lock Statistics

Statistics	Value	Rate (Per Second)	Source
lock.deadlocks	0	0	SystemStats table
lock.locks_acquired.dml	3	.38	SystemStats table
lock.locks_acquired.table_scans	114	14.25	SystemStats table
lock.locks_granted.immediate	13008	1626	SystemStats table
lock.locks_granted.wait	1	.13	SystemStats table
lock.timeouts	0	0	SystemStats table

Figure 4-17 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 4-17 ttStats report: XLA information



XLA Information

- **-1/-1 in Begin Purge_LSN means XLA is not configured in begin_snapshot.**

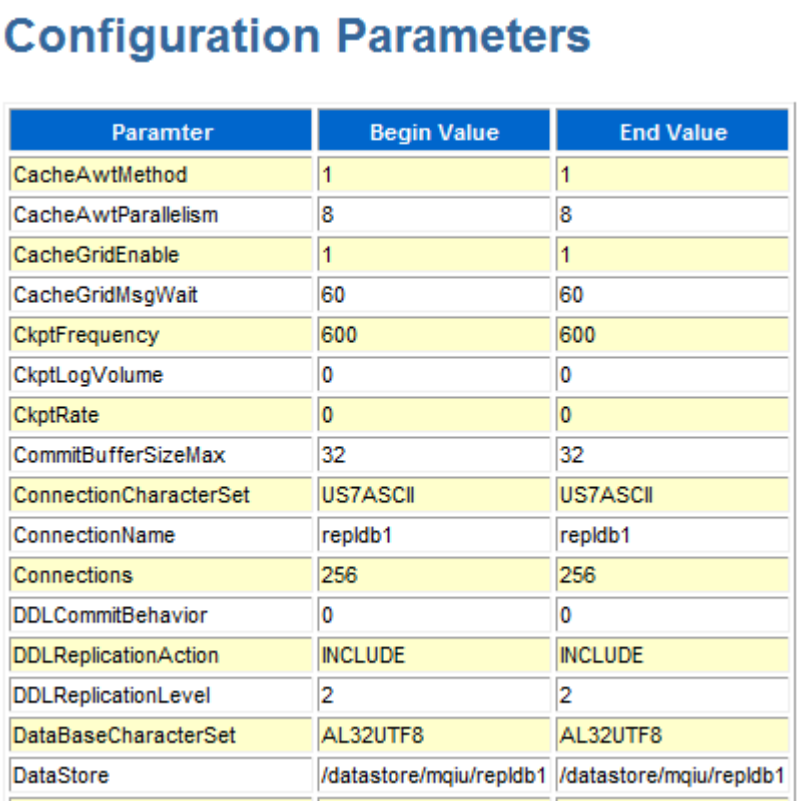
BookMark Name	Purge_LSN		Log_Behind	
	Begin Value	End Value	Begin Value	End Value
bookmark	0/292884840	0/292884840	0	0

Figure 4-18 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 Connection Attributes.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Figure 4-18 ttStats report: configuration parameters



Configuration Parameters

Paramter	Begin Value	End Value
CacheAwtMethod	1	1
CacheAwtParallelism	8	8
CacheGridEnable	1	1
CacheGridMsgWait	60	60
CkptFrequency	600	600
CkptLogVolume	0	0
CkptRate	0	0
CommitBufferSizeMax	32	32
ConnectionCharacterSet	US7ASCII	US7ASCII
ConnectionName	repldb1	repldb1
Connections	256	256
DDLCommitBehavior	0	0
DDLReplicationAction	INCLUDE	INCLUDE
DDLReplicationLevel	2	2
DataBaseCharacterSet	AL32UTF8	AL32UTF8
DataStore	/datastore/mqiu/repldb1	/datastore/mqiu/repldb1

See Also

ttStatsConfig
ttStatsConfigGet

ttStatus

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.
- The status of the start up scripts and which command needs to be used to start the daemon.
- 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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

- 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 | -?}
  or: ttStatus {-V | -version}
  or: ttStatus [-v] [-r secs] [-[no]pretty]
  or: ttStatus [-r secs] [-[no]pretty] {DSN | -connstr connStr | dspath}
```

Options

`ttStatus` has the options:

Option	Description

`-h`
`-help`
`-?`
	Prints usage information and exits.

`-V | -version`	Prints the release number of `ttStatus` and exits.
`-connStr` *`connection_string`*	An ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.
`DSN`	An ODBC data source name of the database for which to get status.
`-v`	Prints detailed information that is useful for TimesTen customer support.
`-r` *`secs`*	Enables `ttStatus` to continue running. Updates status report every *`secs`* seconds.
`-[no]pretty`	The `-pretty` option (default) is for "pretty" formatting, which uses the values of the ConnectionName connection attribute. The `-nopretty` option is to not use pretty formatting.

Sample Output

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
------------------------------------------------------------------------
Open for user connections
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
Type            PID     Context     Connection Name             ConnID
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

```
Process         5196    0x01066a58  cachedb1_18110                   1
Subdaemon       3912    0x00b2c398  Manager                       2047
Subdaemon       3912    0x00b7e4a0  Rollback                      2046
Subdaemon       3912    0x015d25e8  Flusher                       2045
Subdaemon       3912    0x015e46b0  Monitor                       2044
Subdaemon       3912    0x016767f8  Deadlock Detector             2043
Subdaemon       3912    0x016888c0  Checkpoint                    2041
Subdaemon       3912    0x0d350578  Aging                         2042
Subdaemon       3912    0x0d362640  Log Marker                    2040
Subdaemon       3912    0x0d4347c8  AsyncMV                       2039
Subdaemon       3912    0x0d446890  HistGC                        2038
Subdaemon       3912    0x0d458958  IndexGC                       2037
Replication policy  : Manual
Cache Agent policy  : Manual
PL/SQL enabled.
------------------------------------------------------------------------
Accessible by group . . .
End of report
```

Notes

- While primarily intended for use by TimesTen customer support, this information may be useful to system administrators and developers.
- The `ttStatus` utility only reports the RAM policy if it is not `inUse`.

See Also

ttAdmin

ttTail

Fetches TimesTen internal trace information from a database and displays it to `stdout`. By default, TimesTen generates no tracing information. See ttTraceMon 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:

Option	Description
`-connStr` *`connection_string`*	An ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.
`DSN`	Indicates the ODBC data source name of the database from which to get a trace.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
`-f`	When the end of the trace is reached, `ttTail` does not terminate but continues to run, 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.
`-h-help` `-?`	Prints a usage message and exits.
`-V | -version`	Prints the release number of `ttTail` and exits.

Examples

```
% ttTail database1
```

> **Note**
>
> While primarily intended for use by TimesTen customer support, this information may be useful to system administrators and developers.

ttTraceMon

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 run interactively (multiple subcommands can be entered at a prompt) or not interactively (one subcommand can be specified on the `ttTraceMon` command line).

When run 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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Syntax

```
ttTraceMon {-h | -help | -?}
ttTraceMon {-V | -version}
ttTraceMon [-e subcommand] {-connStr connection_string | DSN}
```

Options

`ttTraceMon` has the options:

Option	Description
`-connStr` *`connection_string`*	An ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.
`DSN`	Indicates the ODBC data source name of the database from which to get trace information.
`-e` *`subcommand`*	Causes the subcommand to be run against the specified database. If the subcommand consists of more than one word, enclose it in double quotes. For example: `ttTraceMon -e "show err" database1` Once the subcommand is complete, `ttTraceMon` exits. If `-e` is not specified, `ttTraceMon` starts in interactive mode, reading commands from `stdin` and displaying results to `stdout`.

`-h`
`-help`
`-?`
	Prints a usage message and exits.

`-V | -version`	Prints the release number of `ttTraceMon` and exits.

Subcommands

`ttTraceMon` can be called with the following subcommands:

Command	Description
`components`	List the names and internal identifiers of 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*.
`connection {all | self |` *`connectionNum`*`} [on |off]`	Turn tracing on/off for specified connection. At database creation, tracing is "on" for all connections. The value for *`connectionNum`* is the connection slot number or the first number in the transaction ID.
`dump`	Prints all trace records currently buffered. Requires `SELECT` privileges or database object ownership.
`dump` *`comp`*	Prints all trace records for component *`comp`*.Requires `SELECT` privileges or database object ownership.
`flush`	Discards all buffered trace records.
`flush` *`comp`*	Discards all buffered trace records for component *`comp`*.
`help`	Prints a summary of the trace commands.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Command	Description
`level comp n`	Sets the trace level for component *comp* to *n*.Requires ADMIN privileges or database object ownership.
`outfile file`	Prints trace output to the specified file. The file may be any of 0, stdout, stderr, 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 ttTail. If no *file* is specified, prints the current outfile setting.
`show`	Shows all the trace levels in force.
`show comp`	Shows the trace level for component *comp*.
`tracefiles n`	Sets the maximum number of output files.
`tracefilesize n[M|G]`	Sets the file size limit for output files. If M is specified indicates the file size in the indicated number of megabytes If G is specified indicates the file size in the indicated number of gigabytes.

> **Note**
>
> 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.

ttUser

The ttUser utility helps you secure passwords.

With the ttUser utility you can either:

- Use an Oracle Wallet to securely store user IDs and passwords.
- Hash a password and use the obtained value for the PWDCrypt connection attribute.

Store Your Credentials in an Oracle Wallet

The most secure method to provide credentials when connecting to a database is to store a user's password in an Oracle Wallet.

With ttUser you can perform the following wallet-related tasks:

- Add user IDs and cache administration user IDs with associated passwords to a user-managed Oracle Wallet.
- Provide the name of the directory in which you want the wallet to be placed.
- Remove user IDs and cache administration user IDs with associated passwords from the wallet.

The ttUser utility creates or modifies a wallet as follows:

Copyright © 1996, 2026, Oracle and/or its affiliates.

- TimesTen places the wallet in a subdirectory created by `ttUser`. You provide the path to place such a subdirectory. If the subdirectory does not already exist, then `ttUser` creates it. Do not create this subdirectory yourself.
- If your wallet does not already exist, `ttUser` creates the wallet in the specified location. The credentials are added to the Oracle Wallet in this directory location.
- If your wallet does exist but the user does not exist in the wallet, the `ttUser` utility adds the user and password to the wallet.
- If a user has already been added to the wallet, you can provide a new password to overwrite the existing one.

You can store multiple different users with associated passwords in the same wallet for a particular DSN. However, you need to use separate wallets when you have the same user with different passwords in different DSNs.

Let us assume that the user Terry needs wallets to store their credentials for two different DSNs.

1. Terry creates a directory to contain the wallet for ds1: `/terry/wallets/ds1wallet`.
2. Terry creates a directory to contain the wallet for the ds2: `/terry/wallets/ds2wallet`.
3. Using `ttUser -setPwd`, Terry provides the paths for each directory structure in which they want their wallets placed and their passwords stored. A subdirectory to contain each wallet is created by `ttUser` in each case.

Hash Your Password with ttUser

If you specify the `-pwdCrypt` option, the `ttUser` utility prompts you for a password and returns a hashed 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 | -V}

ttUser -pwdCrypt
ttUser -setPwd -uid {UID} -wallet {walletPath}
       -removePwd -uid {UID} -wallet {walletPath}
       -setOraclePwd -uid {UID} -wallet {walletPath}
       -removeOraclePwd -uid {UID} -wallet {walletPath}
```

Options

The `ttUser` utility supports the options:

Option	Description

`-h`
`-help`
`-?`
	Prints a usage message and exits.

`-pwdCrypt`	Generates an hashed password value for the `PWDCrypt` connection attribute.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
`-removeOraclePwd`	Removes the Oracle cache administration user and password from a wallet.
`-removePwd`	Removes the TimesTen user and password from a wallet.
`-setOraclePwd`	Sets the Oracle cache administration user and password in a wallet.
`-setPwd`	Sets the TimesTen user and password in a wallet.
`-uid {`*`username`* `|` *`cacheadmin`*`}`	The user name for the credentials (required for all wallet options).
`-V | -version`	Prints the release number of `ttUser` and exits.
`-wallet {`*`walletPath`*`}`	Absolute path to a wallet file to be created or updated.

Notes

- You are responsible for securing and managing your wallets.
- If you save a password and there is already an entry in the wallet for that user ID and password type, the existing password is overwritten without warning.
- You are responsible for creating wallets and making them accessible from hosts used to access TimesTen.
- TimesTen recommends having one wallet per combination of TimesTen DSN and Oracle NetServiceName.

Examples

This example sets a password for a TimesTen user. The `ttUser` utility prompts you to enter the password.

```
% ttUser -setPwd -uid terry -wallet /home/terry/wallets/mywallet
Enter password:
```

After you have created the `/home/cacheadmin/wallets` directory to contain all of your wallets, this example sets the password for an Oracle cache administration user.

```
% ttUser -setOraclePwd -uid cacheadmin -wallet /home/cacheadmin/wallets
Enter password:
```

To remove an Oracle password, use `ttUser -removeOraclePwd`.

```
% ttUser -removeOraclePwd -uid cacheadmin -wallet /home/cacheadmin/wallets
```

This example shows how the Oracle cache administration user connects to TimesTen and Oracle through `ttIsql` without providing `Pwd` and `OraclePwd` in the connection string or DSN definition. Instead, the Oracle cache administration user indicates the values for `UID` and `PwdWallet` to specify the wallet from which to retrieve credentials. See `PwdWallet`.

```
Command> connect “dsn=mydb;uid=cacheadmin;oracleNetServiceName=myorcl;PwdWallet=/home/
terry/wallets/mywallet”;
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

ttVersion

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 options:

Option	Description
`-m`	Generates computer-readable enhanced output. If not specified and no attribute is specified, abbreviated information is output.
`attribute`	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.

Attributes

`ttVersion` has these attributes:

Attribute	Description
`patched`	Lists `yes` or `no`, indicating whether the release has been patched.
`config_found`	Lists `yes` or `no`, indicating whether the configuration file is found.
`product`	Lists the name of the product.
`major1`	The first part of the five-part release number (26 for release 26.1.1.1.0), indicating the last two digits of the year of the major release. A change in `major1` indicates major infrastructure and functionality changes.
`major2`	The second part of the five-part release number (1 for release 26.1.1.1.0). A change in only `major2` indicates a version with new functionality and possibly some infrastructure changes. Releases with the same `major1.major2` are patch-compatible. Data can be unloaded from a database from one release and loaded into a database from the other without the migration process.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Attribute	Description
patchset	The third part of the five-part release number (1 for release 26.1.1.1.0). A change in only patchset indicates a release that contains bug fixes and possibly some feature enhancements.
patch	The fourth part of the five-part release number (9 for release 26.1.1.1.0). A change in only patch indicates a release with only critical bug fixes.
reserved	The fifth part of the five-part release number (0 for release 26.1.1.1.0). Reserved for future use.
major3	No longer needed, but maintained for backward compatibility. Same as patchset.
portpatch	No longer needed, but maintained for backward compatibility. Same as reserved.
version	All five parts of the release number, separated by periods (such as 26.1.1.1.0).
shortversion	The first two parts of the five-part release number, without periods (261 for release 26.1.1.1.0).
numversion	All five parts of the release number, without periods, in the format %02d%02d%04d%02d%02d, compatible with ODBC version number format (26010001100 for release 26.1.1.1.0).
bits	Lists 64 to indicate the 64-bit-level of the operating system for which this release is intended.
os	The operating system for which this release is intended
buildstamp	A number indicating the specific build.
buildtime	The UTC time the release was built, for example: 2013-03-19T17:21:59Z
clientonly	Lists yes or no to indicate if the release is a client-only release
instance	The name of the instance, for example: tt2611.
effective_port	The number of the port on which the main daemon listens.
orig_port	The original number of the port on which the main daemon listened.
instance_admin	The user name of the instance administrator.
effective_insthome	The path that indicates the location of the instance.
orig_insthome	The path that indicates the location of the instance.
effective_daemonhome	The path to the home of the daemon for the specific instance.
effective_daemonhome_long	On Windows, the path to the home of the daemon for the specific instance, including a bit extension on the instance name.
orig_daemonhome	The path to the original home of the daemon.
plsql	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.
group_name	The name of the instance group.

Output

The following is the ttVersion output without the -m option:

Copyright © 1996, 2026, Oracle and/or its affiliates.

```
ttVersion
TimesTen Release 26.1.1.1.0 (64 bit Linux/x86_64) (myinstance:6630) 2026-01-13T15:50:07Z
  Instance admin: john
  Instance home directory: /<home_dir>
  Group owner: dba
  Daemon home directory: /<home_dir>/info
  PL/SQL enabled.
```

And with the `-m` option:

```
patched=yes
config_found=yes
product=TimesTen
major1=26
major2=1
patchset=1
patch=1
reserved=0
major3=1
portpatch=0
version=26.1.1.1.0
shortversion=261
numversion=260100010100
bits=64
os=Linux/x86_64
buildtstamp=1768319407
buildtime=2026-01-13T15:50:07Z
xe=0
clientonly=no
instance=myinstance
effective_port=6630
orig_port=6630
instance_admin=john
effective_insthome=/<home_dir>
orig_insthome=/<home_dir>
effective_daemonhome=/<home_dir>/info
orig_daemonhome=/<home_dir>/info
plsql=1
systemd=0
group_name=dba
label=TIMESTEN_MAIN_LINUX.X64_260112.2000
rdbms_label=RDBMS_23.8.0.25.04DBRU_LINUX.X64_RELEASE
mlr=38603953
cjson=1.7.18
```

ttXactAdmin

The `ttXactAdmin` utility lists ownership, status, log and lock information for each outstanding transaction. It 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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

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 connection_string}

ttXactAdmin -connections [-pid pid] [-interval seconds]
[-count iterations] {DSN | -connstr connection_string}

ttXactAdmin -xactIdRollback xid {DSN | -connstr connection_string}

ttXactAdmin {-HCommit xid | -HAbort xid | -HForget xid} {DSN | -connstr
connection_string}
```

Options

`ttXactAdmin` has the options:

Option	Description
`-connections`	Shows all current connections to the database. When run with the `-connections` option, `ttXactAdmin` 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 `ADMIN` privileges.
`-connStr` *`connection_string`*	An ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.
`-count` *`iterations`*	Generate the report iterations times. If no`-interval` option is specified, an interval of 1 second is used.
`DSN`	Indicates the ODBC data source name of the database to be administered. This option requires `ADMIN` privileges.

`-h`
`-help`
`-?`
	Prints a usage message and exits.

`-HAbort` *`xid`*	Heuristically terminates an XA transaction branch in TimesTen. The specified transaction ID must be the local TimesTen TransID. This option requires `ADMIN` privileges or ownership of the specified transactions.
`-HCommit` *`xid`*	Heuristically commit an XA transaction branch in TimesTen. The specified transaction ID must be the local TimesTen TransID. This option requires `ADMIN` privileges or ownership of the specified transactions.
`-HForget` *`xid`*	Heuristically forget an XA transaction branch in TimesTen. The specified transaction ID must be the local TimesTen TransID. This option requires `ADMIN` privileges or ownership of the specified transactions.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
`-interval` *seconds*	Repeat the generation of the report, pausing the indicated number of seconds between each generation. If no `-count` option is specified, repeat forever.
`-latch`	This option is to be used by TimesTen Customer Support only. Shows only the latch information for the database specified.
`-ml` *maxLocks*	Maximum number of locks per transaction. Default is 6000.
`-mt` *maxTrans*	Specifies the maximum number of transactions to be displayed. The default is all outstanding transactions.
`-pid` *pid*	Displays only transactions started by the process with the specified `pid`. On Linux, it is the pid of the thread that opens the connection. This option requires `ADMIN` privileges or ownership of the specified transactions.
`-tbl` [*owner*.]*tableName*	Displays lock information for the specified table. This option requires `ADMIN` privileges or ownership of the specified table.
`-V | -version`	Prints the release number of `ttXactAdmin` and exits.
`-v` *verbosity*	Specifies the verbosity level. One of: `0` - Does not display the names of the tables for row locks. In this case, `ttXactAdmin` runs faster. 1 (default) - Displays the names of the tables for row locks.
`-xact` *xid*	Displays information for the specified transaction, including its log hold LSN.In the output, the field "`Last ID`" is a set of two sequence numbers. If the sequence numbers did not change in an interval, then no log record was written by the transaction during that interval. This option requires `ADMIN` privileges or ownership of the specified transactions.
`-xactIdRollback` *xid*	Enables you to roll back a transaction. This may be particularly useful for long running transactions. The parameter *xid* represents the transaction ID. This stops any currently running operations on behalf of that transaction and then rollback the transaction in TimesTen. If there is currently a checkpoint in process when the rollback is requested, TimesTen terminates the checkpoint operation. This command does not stop cache operations on the Oracle database. Operations include passthrough statements, flushing, manual loading, manual refreshing, synchronous writethrough, propagating, and dynamic loading. This option requires `ADMIN` privileges or ownership of the specified transactions.

Output

`ttXactAdmin` produces the following output:

Column	Description
Program File Name	The executable file name of the process that owns the transaction.
PID	The process ID of the application that owns the transaction. On Linux, the PID of the thread that opens the connection.
Context	The internal identifier that distinguishes between multiple connections to the database made by a single multithreaded process.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Column	Description
XactID	The unique identifier for the transaction used internally by TimesTen. The identifier has two parts. The first part is a relatively small value (less than 2048), used to identify the connection of the program running 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).
State	Current state of the transaction, one of: `Active` - Active transaction. `Aborting` - A transaction is in the process of terminating. See Notes for more information. `Committing` - Committing transaction, locks are being released. `Ckpointing` - A transaction doing checkpoint. `Rep-Wait-Return` - Replicated transaction waiting Return Receipt/ Commit. `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.
Resource	The type of the lock being requested: `Row` - Row-level lock. `HashedKey` - A lock held on a key value of a hash index; acquired when an operation requires a hash index to be updated. `Table` - Table-level lock. `EndScan` - End of table or range scan lock. `Database` - Database-level lock. `Command` - Command lock. `Prepare` - Lock acquired while preparing commands. `GrpComm` - Group commit lock. `ReplHold` - Lock for replication hold. `XlaHold` - Lock for XLA hold.
ResourceId	A unique identifier of each unique resource. The identifier is displayed in hexadecimal format with a few exception. `Table` and `CompCmd` are shown as decimal values. Row locks are shown in the `ROWID` character format.

Copyright © 1996, 2026, Oracle and/or its affiliates.

Column	Description
Mode	A value used to determine the level of concurrency that the lock provides: S - Shared lock in serializable isolation. Sn - Shared lock in non-serializable isolation. U - Update lock in serializable isolation. Un - Update lock in non-serializable isolation. En - End-of-scan lock for non-serializable isolation. IRC - Intention shared lock in non-serializable isolation. IS - Intention shared lock in serializable isolation. IU - Intention update lock in serializable isolation. IUn - Intention update lock in non-serializable isolation. IX - Intention exclusive lock in serializable isolation. IXn - Intention exclusive lock non-serializable isolation. SIX - Shared lock with intent to set an exclusive lock in serializable isolation. SIXn - Shared lock with intent to set an exclusive lock non-serializable isolation. X - Exclusive lock. Xn - Exclusive lock in non-serializable isolation. W - Update, insert or delete table lock. XNi - Next lock for inserting into tables or non-unique index. NS - Table lock in read-committed isolation that conflicts with all table locks in serializable isolation. Lock "0" means the blocker is still in the waiting list.
HMode	The mode in which the competing transaction is holding the lock which the waiting transaction is requesting. See "Mode" in this table for concurrency level descriptions.
RMode	Shows the mode in which the waiting transaction has requested to hold the lock. See "Mode" in this table for concurrency level descriptions.
HolderTransId	The identifier of the transaction with which the waiting transaction is in contention.
Name	The name of the table that the lock is being held on or within.

Examples

The following command displays all locks in the database:

```
% ttXactAdmin -connStr "DSN=demodata"
2022-03-20 13:02:54.760
/timesten/jsmith/demo/demodata
TimesTen Release 26.1.1.1.0
ElementID 1

Program File Name: _ttIsql
XactID            PID      Context            State       Loghold Last ID
1.1.195           115640   0x859570           Active      391.15355904 [1666839:6]

  Resource  ResourceID           Mode  SqlCmdID             Name
  Database  0x01312d0001312d00   IX    0
  Table     2367528              IXn   275642480            JSMITH.T
  Row       AAAVVUAAADXAQAANje   Xn    275642480            JSMITH.T

  Begin Time: 13:01:43.108
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

```
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 a 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

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 lfn.lfo [-l2 lfn.lfo]] [-r recType] [...] [-tr dir]
[-lb] [-headers recs] [-logdir dir]
{-connStr connection_string | DSN | dspath}

ttXactLog [-v verbosity] -logAnalyze
[-subscriber subscriberName -host hostname]]
[-xid xid] {-connStr connection_string | DSN | dspath}
```


Copyright © 1996, 2026, Oracle and/or its affiliates.

```
ttXactLog -moveBookmark id [-o option] [-l1 lfn.lfo]
```

Options

ttXactLog has the options:

Option	Description
-b *blkID*	Restricts log records to those accessing this block, plus any transaction records.
-connStr *connection_string*	An ODBC connection string that specifies a database location, driver, and optionally other connection attribute settings.
DSN	The ODBC source name of the database for which to display the transaction log.
dspath	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 DataStore=*parameter* of the database's ODBC definition. For example, for a database consisting of files/home/payroll/2011.ds0, /home/payroll/2011.ds1 and several transaction log files /home/payroll/2011.log*n*,*dspath* is /home/payroll/2011.
-h -help -?	Prints a usage message and exits.
-headers *records*	Prints one header for every *records* records. A value of 0 disables headers.
-host *hostName*	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.
-lb	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.
lfn.lfo	Transaction log file number (*lfn*) and transaction log file offset (*lfo*) for a log record.
-l1	Considers this log record only (unless an -l2 argument is present).
-l2	Considers records between -l1 and -l2, inclusive.
-logAnalyze	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.
-logdir *dir*	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.
-m *maxChars*	Maximum number of characters printed for binary items (for -v 3) only (defaults to 4000).

Copyright © 1996, 2026, Oracle and/or its affiliates.

Option	Description
`-r recType`	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.
`-s`	Prints summary information. Requires `SELECT` privileges or database object ownership.
`-subscriber subscriberName`	Specifies the name of the subscriber. To qualify the name of the subscriber, use `-host hostname`.
`-t`	Only reads transaction log file tail (from start of last checkpoint transaction log file or, if no checkpoint, the most recent transaction log file).
`-tr dir`	Truncates all log records in the directory at the `LWN` boundary. The original transaction log files are moved to the directory `dir`.
`-V | -version`	Prints the release number of `ttXactLog` and exits.
`-v verbosity`	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).
`-x xid`	Specifies the transaction ID.
`-moveBookmark id`	Moves the specified ID bookmark to the nearest logMarker in the log.
`-o option`	Sets the defined XLA bookmark • `0` - set the XLA bookmark to the next log marker in the log (default). • `1` - set the XLA bookmark to the latest logmarker in the dbhdr. • `2` - set the XLA bookmark to the nearest logmarker of a specified lsn value.

Examples

```
% ttXactLog -v 3 -m 100 /users/pat/TimesTen/Daily/F112697SS
```

See Also

Analyze Outstanding Transactions in the Replication Log in the *Oracle TimesTen In-Memory Database Replication Guide* .

Copyright © 1996, 2026, Oracle and/or its affiliates.

5

System Limits

Operating system limits and defaults.

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*.

Description	Value
Maximum number of subscriber databases in a replication scheme that is not an active standby pair.	255
Maximum number of propagators in a replication scheme. Each propagator can have the maximum number of subscribers.	255
Maximum number of subscriber databases in an active standby pair.	254
Minimum database size (bytes). Size includes both the permanent and temporary space required to perform operations on the database.	32 MB
Maximum length for a fixed-length column (bytes).	8,300
Maximum number of columns in a table.	1,000
Maximum number of columns in an `ORDER BY` clause.	1,000
Maximum number of columns in a `GROUP BY` clause.	1,000
Maximum cumulative length of a row's fixed-length columns (bytes).	32,768
Maximum length for a varying-length column (bytes).	2^{22} = 4,194,304
Maximum length for a replicated column.	4 MB 16 MB for columns with `BLOB` type
Maximum number of concurrent connections to a database (including system connections).	32,047
Maximum number of concurrent application connections to a database (may be limited by semaphore configuration or `Connections` DSN attribute or both).	32,000 2,000 (default)
Maximum number of connections (system and application) across all databases in an instance.	32,048
Maximum number of concurrent client connections to a TimesTen instance. **Note:** Some instances may support a slightly smaller maximum number of connections depending on such things as whether the database is shared or replicated and operating system limits.	32,048

Copyright © 1996, 2026, Oracle and/or its affiliates.

Description	Value
Maximum length of database names.	32
Maximum length of the path name for a database in an asynchronous writethrough cache group	248
Maximum number of projected expressions in a SELECT statement.	32,767
Maximum length of string specifying a join order.	1,024
Maximum number of columns in an index (or primary key).	32
Maximum length of basic names.	30
Maximum length of displayed predicate string in the SYS.PLAN table.	1,024
Maximum length of SQL statement, including the NULL terminator.	409,600
Maximum number of table references in an SQL query.	24
Maximum number of indexes on a table.	500
Maximum number of partitions in a table.	999
Maximum number of prepared PL/SQL statements per connection.	5000
Maximum number of recently-used PL/SQL blocks that can be cached per session.	5000
Maximum number of concurrent segment client/server connections per TimesTen instance	Unlimited (up to the limits of the operating system)
Maximum number of concurrent shared memory segment client/server connections per TimesTen instance.	Unlimited (up to the limits of the operating system)
Maximum size of IPC shared memory segment for client/server connections	4 GB
Maximum number of allocated statement handles per shared memory segment client/server connection.	512
Maximum depth of nesting subqueries.	Equal to the maximum number of table references in a SQL query.
Maximum error message length for applications that specify an error message length (for example, through a call to SQLError).	512
Maximum number of replicated XLA bookmarks.	64

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 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.

Copyright © 1996, 2026, Oracle and/or its affiliates.

- 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.

Copyright © 1996, 2026, Oracle and/or its affiliates.