

*Oracle TimesTen
In-Memory Database
Documentation Addendum
Release 7.0.4*

B12345



Copyright ©1996, 2008, Oracle. All rights reserved.

ALL SOFTWARE AND DOCUMENTATION (WHETHER IN HARD COPY OR ELECTRONIC FORM) ENCLOSED AND ON THE COMPACT DISC(S) ARE SUBJECT TO THE LICENSE AGREEMENT.

The documentation stored on the compact disc(s) may be printed by licensee for licensee's internal use only. Except for the foregoing, no part of this documentation (whether in hard copy or electronic form) may be reproduced or transmitted in any form by any means, electronic or mechanical, including photocopying, recording, or any information storage and retrieval system, without the prior written permission of TimesTen Inc.

Oracle, JD Edwards, PeopleSoft, Retek, TimesTen, the TimesTen icon, MicroLogging and Direct Data Access are trademarks or registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

February 2008

Printed in the United States of America

Contents

About this Addendum

Conventions used in this guide	1
Technical Support	2

1 Platform Support and Installation Prerequisites

Platform support.	3
JDK Support.	3
Cache Connect to Oracle	5
Installation prerequisites on HP-UX platforms	5
Veritas file system.	5
Shared memory daemon option	6

2 Disaster Recovery Using Active Standby Pair Replication With AWT Cache Groups

Overview	8
Requirements for using a disaster recovery subscriber with an active standby pair.	9
Rolling out a disaster recovery subscriber	10
Switching over to the disaster recovery site	12
Creating a new active standby pair after switching over to the disaster recovery site	12
Switching over to a single data store	13
Returning to the original configuration at the primary site	14

3 Query Threshold Warning

Overview	16
SNMP trap	17
Warning message	17
Setting the query threshold.	18
QueryThreshold general connection attribute	18
Setting the query threshold with ttIsql	19
Setting the query threshold with an ODBC option	19
Setting the query threshold in JDBC	19
Setting the query threshold with TTCclasses.	20
Retrieving the query threshold	21
Retrieving the QueryThreshold attribute with ttIsql	21
Retrieving the TT_QUERY_THRESHOLD ODBC option.	21
Retrieving the query threshold with JDBC	21

Retrieving the query threshold with TTClasses21
Setting the query threshold for replication22
ttRepQueryThresholdSet22
ttAdmin -repQueryThresholdSet23
Retrieving the query threshold for replication24
ttRepQueryThresholdGet24
ttAdmin -repQueryThresholdGet24

4 Monitoring AWT Cache Group Performance

Overview.27
Enabling AWT monitoring27
ttCacheAwtMonitorConfig28
Displaying AWT monitoring results30
ttRepAdmin -showstatus -awtmoninfo30

5 Cleaning Up Autorefresh Objects on Oracle

Overview.34
Using the autorefresh cleanup script35

6 Detection of Dual Active Masters in an Active Standby Pair

7 Transaction Log API (XLA)

8 New Error Messages and SNMP Traps

Error messages41
SNMP traps.45

Index

About this Addendum

This documentation addendum contains information about features added in Release 7.0.4 of the Oracle TimesTen In-Memory Database. For more information about the Oracle TimesTen In-Memory Database, please see the documentation set for Release 7.0.3.

Conventions used in this guide

TimesTen supports multiple platforms. Unless otherwise indicated, the information in this guide applies to all supported platforms. The term Windows refers to Windows 2000, Windows XP and Windows Server 2003. The term UNIX refers to Solaris, Linux, HP-UX, Tru64 and AIX.

TimesTen documentation uses these typographical conventions:

If you see...	It means...
<code>code font</code>	Code examples, filenames, and pathnames. For example, the <code>.odbc.ini</code> or <code>ttconnect.ini</code> file.
<i>italic code font</i>	A variable in a code example that you must replace. For example: <code>Driver=install_dir/lib/libtten.sl</code> Replace <code>install_dir</code> with the path of your TimesTen installation directory.

TimesTen documentation uses these conventions in command line examples and descriptions:

If you see...	It means...
<code>fixed width italics</code>	Variable; must be replaced with an appropriate value. In some cases, such as for parameter values in built-in procedures, you may need to single quote (' ') the value.
[]	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 arguments that you may use more than one argument on a single command line.

...	An ellipsis (...) after an argument indicates that you may use more than one argument on a single command line.
%	The percent sign indicates the UNIX shell prompt.
#	The number (or pound) sign indicates the UNIX root prompt.

TimesTen documentation uses these variables to identify path, file and user names:

If you see...	It means...
<i>install_dir</i>	The path that represents the directory where the current release of TimesTen is installed.
<i>TTinstance</i>	The instance name for your specific installation of TimesTen. Each installation of TimesTen must be identified at install time with a unique alphanumeric instance name. This name appears in the install path. The instance name “giraffe” is used in examples in this guide.
<i>bits</i> or <i>bb</i>	Two digits, either 32 or 64, that represent either the 32-bit or 64-bit operating system.
<i>release</i> or <i>rr</i>	Two digits that represent the first two digits of the current TimesTen release number, with or without a dot. For example, 70 or 7.0 represents TimesTen Release 7.0.
<i>jdk_version</i>	Two digits that represent the version number of the major JDK release. Specifically, 14 represent JDK 1.4; 5 represents JDK 5.
<i>timesten</i>	A sample name for the TimesTen instance administrator. You can use any legal user name as the TimesTen administrator. On Windows, the TimesTen instance administrator must be a member of the Administrators group. Each TimesTen instance can have a unique instance administrator name.
<i>DSN</i>	The data source name.

Technical Support

For information about obtaining technical support for TimesTen products, go to the following Web address:

<http://www.oracle.com/support/contact.html>

Platform Support and Installation Prerequisites

This chapter describes changes and additions to platform support and installation prerequisites for Oracle TimesTen In-Memory Database Release 7.0.4.0.0. We recommend that you first read the *Oracle TimesTen In-Memory Database Installation Guide* if you have not previously installed TimesTen or if you have questions regarding installing TimesTen on your particular platform or for your configuration. Specifically, this chapter describes changes and additions for:

- [JDK Support](#)
- [Cache Connect to Oracle](#)
- [Installation prerequisites on HP-UX platforms](#)

Platform support

JDK Support

New in 7.0.4, JDK 6 is supported with the Oracle TimesTen In-Memory Database. For clarity, this section contains the entire JDK support table.

Note: This table supersedes the table that you find in the *Oracle TimesTen In-Memory Database Installation Guide*.

TimesTen supports the following JDKs on the specified platforms:

Environment	JDK 1.4	JDK 5.0	JDK 6.0
Asianux 2.0 for Intel IA-32	Yes (Sun and BEA JRockit)	Yes (Sun and BEA JRockit)	Yes (Sun and BEA JRockit)
Asianux 2.0 for EM64T and AMD Opteron CPUs, 32-bit and 64-bit		Yes (Sun and BEA JRockit)	Yes (Sun and BEA JRockit)
HP-UX 11i, 11i v2 and 11iv3 for PA-RISC, 32-bit and 64-bit	Yes (HP 1.4.2)	Yes (HP)	

Environment	JDK 1.4	JDK 5.0	JDK 6.0
HP-UX 11i v2 and 11iv3 for Itanium2, 32-bit and 64-bit	Yes (HP 1.4.2)	Yes (HP)	
HP-UX Tru64 UNIX 5.1B for AlphaChip EV68 CPUs	Yes (HP)		
IBM AIX 5L 5.2 and 5.3 for POWER CPUs, 32-bit and 64-bit	Yes (IBM 1.4.2 or greater)	Yes (IBM)	
Microsoft Windows 2000, Windows XP and Windows Server 2003 for Intel IA-32 and EM64T and AMD Opteron CPUs, 32-bit and 64-bit	Yes (Sun 32-bit)	Yes (Sun and BEA JRockit)	Yes (Sun and BEA JRockit)
MontaVista Linux Carrier Grade Edition Release 4.0 for Intel IA-32 and EM64T CPUs, 32-bit and 64-bit	Yes (Sun 32-bit)	Yes (Sun and BEA JRockit)	Yes (Sun and BEA JRockit)
Oracle Enterprise Linux 4 and 5 for Intel IA-32 CPUs	Yes (Sun)	Yes (Sun and BEA JRockit)	Yes (Sun and BEA JRockit)
Oracle Enterprise Linux 4 and 5 for EM64T and AMD Opteron CPUs, 32-bit and 64-bit		Yes (Sun and BEA JRockit)	Yes (Sun and BEA JRockit)
Red Hat Enterprise Linux 3, 4 and 5 for Intel IA-32	Yes (Sun)	Yes (Sun and BEA JRockit)	Yes (Sun and BEA JRockit)
Red Hat Enterprise Linux 3, 4 and 5 for Intel Itanium2 CPUs	Yes (Sun)	Yes (Sun and BEA JRockit)	
Red Hat Enterprise Linux 3, 4 and 5 for EM64T and AMD Opteron CPUs, 32-bit and 64-bit		Yes (Sun and BEA JRockit)	Yes (Sun and BEA JRockit)
Solaris 8, 9 and 10 for UltraSparc CPUs, 32-bit and 64-bit	Yes (Sun)	Yes (Sun and BEA JRockit 64-bit)	Yes (Sun)
Solaris 10 for AMD Opteron CPUs, 32-bit and 64-bit		Yes (Sun)	Yes (Sun)
SuSE LINUX Enterprise Server 9 and 10 for Intel IA-32 and EM64T and AMD Opteron CPUs	Yes (Sun 32-bit)	Yes (Sun and BEA JRockit)	Yes (Sun and BEA JRockit)

Environment	JDK 1.4	JDK 5.0	JDK 6.0
SuSE LINUX Enterprise Server 10 for Itanium2 CPUs	Yes (Sun)	Yes (BEA JRockit)	

Cache Connect to Oracle

TimesTen Cache Connect to Oracle allows you to cache Oracle Database data in TimesTen. The following Oracle client and server releases are supported with this option for TimesTen 7.0.4:

- Oracle Database 9i Release 2 (Oracle 9.2.0.8.0 or above)
- Oracle Database 10g Release 1 (Oracle 10.1.0.5.0 or above)
- Oracle Database 10g Release 2 (Oracle 10.2.0.1.0 or above)
- Oracle Database 11g Release 1 on these platforms:
 - Microsoft Windows (32-bit)
 - Linux (32-bit and 64-bit)

To enable TimesTen Cache Connect to Oracle on MontaVista CGE 4.0, install the 10.2.0.3 or 11.1.0.6.0 Linux version of the Oracle Instant Client.

Installation prerequisites on HP-UX platforms

Veritas file system

Before installing TimesTen, make sure the appropriate requirements are met for your operating system.

Filesystem options

On the Veritas file system set the options `discovered_direct_iosz` and `max_direct_iosz` to 3MB.

The absence of these direct I/O settings could result in poor file system performance for TimesTen operations.

To set these options, log in as root and use:

```
# /usr/sbin/vxtunefs -o discovered_direct_iosz=3145728
# /usr/sbin/vxtunefs -o max_direct_iosz=3145728
```

Note: Using `vxtunefs` online option requires Advanced VxFS.

For more on running TimesTen on a Veritas file system, see “Installation prerequisites” in the *Oracle TimesTen In-Memory Database Installation Guide*.

Shared memory daemon option

This release includes a new daemon option that allows you to specify the locality of the shared memory segment where TimesTen data stores will exist on HP-UX ccNUMA systems. ccNUMA systems have non-uniform memory latency depending on the data location. Accessing data in a remote cell takes longer than accessing data in a local cell. To ensure the best results for TimesTen operations, set the `IPC_MEM_LOCAL` and confine the TimesTen processes to the local cell.

To set the locality hint for the shared memory segment, on a separate line in the `ttendaemon.options` file, add:

```
-shmLocalityHint locality_hint
```

Legal values for `locality_hint` are:

- `IPC_MEM_LOCAL`
- `IPC_MEM_INTERLEAVED`
- `IPC_MEM_FIRST_TOUCH`
- `IPC_MEM_STRIPED`

Only one value string can be specified at a time. If specified, TimesTen attempts to create the shared memory segment for all data stores in the instance with the appropriate locality hint.

Note: This option only takes effect if the instance administrator has permission to access the memory resource.

The semantics of the hints are described in the man page for `shmget()`. The default behavior is to create the segment without the hint. Expect the default behavior if the daemon option is not specified or if it is specified incorrectly. To see whether a segment has been created with the hint, use the HP-UX `pstat()` facility. See the HP-UX man page for `pstat()`.

For an overview of TimesTen daemon options, see “Managing TimesTen daemon options,” in the *Oracle TimesTen In-Memory Database Operations Guide*.

Disaster Recovery Using Active Standby Pair Replication With AWT Cache Groups

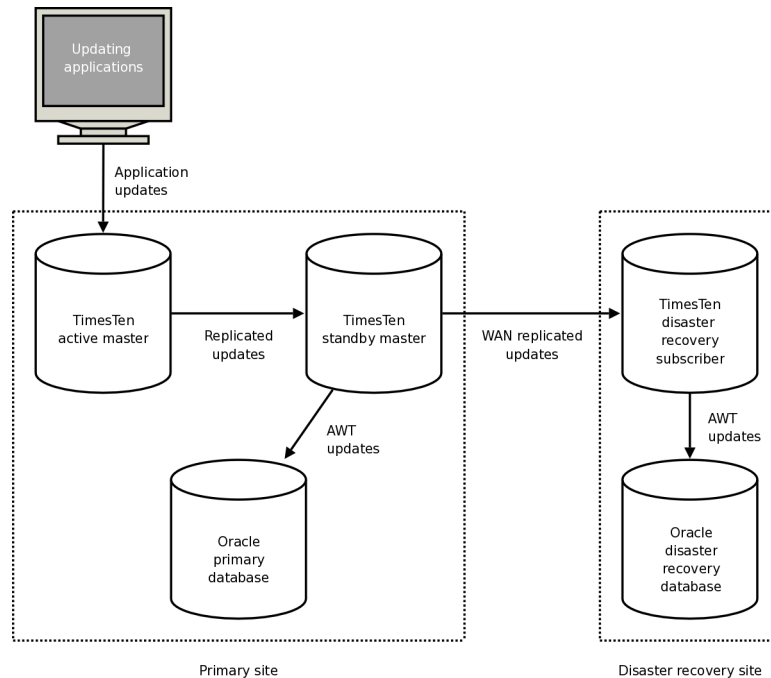
TimesTen 7.0.4.0.0 adds support for a special subscriber in an active standby pair replication scheme that is able to propagate updates to an Oracle database using asynchronous writethrough (AWT) cache groups. This chapter describes how to deploy and use this subscriber at a remote disaster recovery site in the following sections:

- [Overview](#)
- [Rolling out a disaster recovery subscriber](#)
- [Switching over to the disaster recovery site](#)
- [Returning to the original configuration at the primary site](#)

Overview

TimesTen active standby pair replication provides high availability by allowing for fast switching between data stores within a data center. This includes the ability to automatically change which data store propagates changes to an Oracle database using AWT cache groups (see “ASYNCHRONOUS WRITETHROUGH cache groups in an active standby pair” in the *TimesTen to TimesTen Replication Guide* for more information). However, for additional high availability across data centers, you may require the ability to recover from a failure of an entire site, which can include a failure of both TimesTen master data stores in the active standby pair as well as the Oracle database used for the cache groups.

Figure 2.1 Active Standby Pair with a Disaster Recovery Subscriber



You can recover from a complete failure of a site by creating a special disaster recovery read-only subscriber as part of the active standby pair replication configuration. This special subscriber, located at a remote disaster recovery site, can propagate updates to a second Oracle database, also located at the disaster recovery site. The disaster recovery subscriber can take over as the active master in a new active standby pair at the disaster recovery site if the primary site suffers a complete failure. Any applications may then connect to the disaster recovery site and continue operating, with minimal interruption of service.

Requirements for using a disaster recovery subscriber with an active standby pair

To use a disaster recovery subscriber, you must:

- Be using an active standby pair configuration with AWT cache groups at the primary site, as described in “Setting up an active standby pair” in the *TimesTen to TimesTen Replication Guide*.
- Have a continuous WAN connection from the primary site to the disaster recovery site. This connection should have at least enough bandwidth to guarantee that the normal volume of transactions can be replicated to the disaster recovery subscriber at a reasonable pace.
- Have an Oracle database configured at the disaster recovery site to include tables with the same schema as the database at the primary site. Note that this database is intended only for capturing the replicated updates from the primary site, and if any data exists in tables written to by the cache groups when the disaster recovery subscriber is created, that data is deleted.
- Have the same cache group administrator user ID and password at both the primary and the disaster recovery site.

Though it is not absolutely required, you should have a second TimesTen data store configured at the disaster recovery site. This data store can take on the role of a standby master data store, in the event that the disaster recovery subscriber is promoted to an active master data store after the primary site fails.

Rolling out a disaster recovery subscriber

To create a disaster recovery subscriber, follow these steps:

1. Create an active standby pair with AWT cache groups at the primary site as detailed in “Setting up an active standby pair” in the *TimesTen to TimesTen Replication Guide*.
2. Create the disaster recovery subscriber at the disaster recovery site using the **ttRepAdmin** command with the `-duplicate` and `-cacheInitDR` options. You must also specify the cache group administrator and password for the Oracle database at the disaster recovery site using the `-cacheUid` and `-cachePwd` options.

If your data store includes multiple cache groups, you may improve the efficiency of the duplicate operation by using the `-nThreads` option to specify the number of threads that are spawned to flush the cache groups in parallel. Each thread flushes an entire cache group to Oracle and then moves on to the next cache group, if any remain to be flushed. If a value is not specified for `-nThreads`, only one flushing thread is spawned.

For example, to duplicate the standby master data store `mast2`, on the system with the hostname “primary” and the cache user ID “system” and password “manager”, to the disaster recovery subscriber `drsub`, and using two cache group flushing threads, use:

```
ttRepAdmin -duplicate -from mast2 -host primary -cacheInitDR
           -nThreads 2 -cacheUid system -cachePwd manager drsub
```

If you use the **ttRepDuplicateEx** function in C, you must set the `TT_REPDUPE_INITCACHEDR` flag in `ttRepDuplicateExArg.flags` and may optionally specify a value for `ttRepDuplicateExArg.nThreads4InitDR`:

```
int          rc;
ttUtilHandle utilHandle;
ttRepDuplicateExArg arg;
memset( &arg, 0, sizeof( arg ) );
arg.size = sizeof( ttRepDuplicateExArg );
arg.flags = TT_REPDUPE_INITCACHEDR;
arg.nThreads4InitDR = 2;
arg.cacheuid = "system";
arg.cachepwd = "manager";
arg.localHost = "disaster";
rc = ttRepDuplicateEx( utilHandle, "DSN=drsub",
                     "mast2", "primary", &arg );
```

After the subscriber is duplicated, TimesTen automatically configures the asynchronous writethrough replication scheme that propagates updates from the cache groups to the Oracle database, truncates the tables in the Oracle database that correspond to the cache groups in TimesTen, and then flushes all of the data in the cache groups to the Oracle database.

3. Start the replication agent for the disaster recovery subscriber using the **ttRepStart** procedure or the **ttAdmin** command with the option `-restart`. For example:

```
ttAdmin -restart drsub
```

Updates are now replicated from the standby master data store to the disaster recovery subscriber, which then propagates the updates to the Oracle database at the disaster recovery site.

Switching over to the disaster recovery site

When the primary site has failed, you can switch over to the disaster recovery site in one of two ways. If your goal is to minimize risk of data loss at the disaster recovery site, you may roll out a new active standby pair using the disaster recovery subscriber as the active master data store. If the goal is to absolutely minimize the downtime of your applications, at the risk of data loss if the disaster recovery data store later fails, you may instead choose to drop the replication scheme from the disaster recovery subscriber and use it as a single non-replicating data store. You may deploy an active standby pair at the disaster recovery site later.

Creating a new active standby pair after switching over to the disaster recovery site

1. Any read-only applications may be redirected to the disaster recovery subscriber immediately. Redirecting applications that make updates to the data store must wait until Step 7.
2. Ensure that all of the recent updates to the cache groups have been propagated to the Oracle database using the **ttRepSubscriberWait** procedure or the **ttRepAdmin** command with the `-wait` option.

```
ttRepSubscriberWait( null, null, '_ORACLE', null, 600 );
```

If **ttRepSubscriberWait** returns 0x01, indicating a timeout, you may need to investigate to determine why the cache groups are not finished propagating before continuing to Step 3.

3. Stop the replication agent on the disaster recovery subscriber using the **ttRepStop** procedure or the **ttAdmin** command with the `-repstop` option. For example, to stop the replication agent for the subscriber `drsub`, use:

```
call ttRepStop;
```

4. Drop the active standby pair replication scheme on the subscriber using the `DROP ACTIVE STANDBY PAIR` statement. For example:

```
DROP ACTIVE STANDBY PAIR;
```

5. Create a new active standby pair replication scheme using the `CREATE ACTIVE STANDBY PAIR` statement, specifying the disaster recovery subscriber as the active master data store. For example, to create a new active standby pair with the former subscriber `drsub` as the active master and the new data store `drstandby` as the standby master, and using the `return twosafe` return service, use:

```
CREATE ACTIVE STANDBY PAIR drsub, drstandby RETURN TWOSAFE;
```


6. Set the new active standby master data store to the ACTIVE state using the **ttRepStateSet** procedure. For example, on the data store `drsub` in this example, execute:

```
call ttRepStateSet( 'ACTIVE' );
```
7. Any applications which must write to the TimesTen data store may now be redirected to the new active master data store.
8. Continue rolling out the active standby pair as detailed in “Setting up an active standby pair” in the *TimesTen to TimesTen Replication Guide*, starting with step 6.

Switching over to a single data store

1. Any read-only applications may be redirected to the disaster recovery subscriber immediately. Redirecting applications that make updates to the data store must wait until Step 5.
2. Stop the replication agent on the disaster recovery subscriber using the **ttRepStop** procedure or the **ttAdmin** command with the `-repstop` option. For example, to stop the replication agent for the subscriber `drsub`, use:

```
call ttRepStop;
```
3. Drop the active standby pair replication scheme on the subscriber using the DROP ACTIVE STANDBY PAIR statement. For example:

```
DROP ACTIVE STANDBY PAIR;
```
4. Although there is no longer an active standby pair configured, AWT cache groups require the replication agent to be started. Start the replication agent on the data store using the **ttRepStart** procedure or the **ttAdmin** command with the `-repstart` option. For example, to start the replication agent for the data store `drsub`, use:

```
call ttRepStart;
```
5. Any applications which must write to a TimesTen data store may now be redirected to the this data store.

Note: You may choose to roll out an active standby pair at the disaster recovery site at a later time. You may do this by following the steps in “[Creating a new active standby pair after switching over to the disaster recovery site](#)” on page 12, starting at Step 2 and skipping Step 4.

Returning to the original configuration at the primary site

When the primary site is usable again, you may wish to move the working active standby pair from the disaster recovery site back to the primary site. You can do this with a minimal interruption of service by reversing the process that was used to create and switch over to the original disaster recovery site. Follow these steps:

1. Destroy original active master data store at the primary site, if necessary, using the **ttDestroy** command. For example, to destroy a data store called `mast1`, use:

```
ttDestroy mast1
```
2. Create a disaster recovery subscriber at the primary site, following the steps detailed in ["Rolling out a disaster recovery subscriber"](#). Use the original active master data store for the new disaster recovery subscriber.
3. Switch over to new disaster recovery subscriber at primary site, as detailed in ["Switching over to the disaster recovery site"](#). Roll out the standby master data store as well.
4. Roll out a new disaster recovery subscriber at the disaster recovery site, as detailed in ["Rolling out a disaster recovery subscriber"](#).

Query Threshold Warning

This chapter includes the following topics:

- [Overview](#)
- [Setting the query threshold](#)
- [Retrieving the query threshold](#)
- [Setting the query threshold for replication](#)
- [Retrieving the query threshold for replication](#)

Overview

You can configure TimesTen to write a warning to the support log and throw an SNMP trap when the execution of a SQL statement exceeds a specified time duration. Execution continues and is not affected by the query threshold.

The query threshold applies to the following ODBC and JDBC calls:

- `SQLExecute`, `SQLExecDirect` and `SQLFetch` ODBC functions
- `execute()`, `executeBatch()`, `executeQuery()` and `executeUpdate()` methods of the JDBC `Statement` interface
- `next()` method of the JDBC `ResultSet` interface

Thus the query threshold applies to procedure calls as well as SQL statements.

Set the query threshold by one of the following methods:

- Set the `QueryThreshold` general connection attribute in the ODBC.INI file or with the `ttIsql` utility
- Set the `TT_QUERY_THRESHOLD` option with the `SQLSetConnectOption` and `SQLSetStmtOption` ODBC functions
- Set the `QueryThreshold` attribute in the JDBC Connection URL
- Use the `setQueryTimeThreshold` method of the `TimesTenStatement` interface
- Use the `TTCmd::setQueryThreshold` `TTClasses` method

Retrieve the query threshold by one of the following methods:

- `ttIsql` utility
- Use the `SQLGetConnectOption` or `SQLGetStmtOption` ODBC functions
- Use the `TTCmd::getQueryThreshold` `TTClasses` method

The replication agent does not inherit the `QueryThreshold` setting from the ODBC.INI file. Use the `ttRepQueryThresholdSet` built-in procedure to set a query threshold for the replication agent. Alternatively, you can specify the `-repQueryThresholdSet` option of the `ttAdmin` utility. The query threshold for the replication agent applies to SQL execution on detail tables of materialized views, ON DELETE CASCADE operations and some internal operations that execute SQL statements.

You can retrieve the query threshold value for replication by using the `ttRepQueryThresholdGet` built-in procedure or the `-repQueryThresholdGet` option of the `ttAdmin` utility.

You cannot set a query threshold for SQL statements that are executed by the cache agent.

SNMP trap

The name of the SNMP trap is `ttQueryThresholdWarnTrap`. See *Oracle TimesTen In-Memory Database Error Messages and SNMP Traps* for information about configuring SNMP traps.

Warning message

When the query threshold is exceeded, TimesTen writes a warning message to the support log.

Example 3.1

This is an example of a warning message in the support log:

```
2007-12-03 15:55:33.67 Warn:      : 9692: 7373/0x95c5038: QT rep1 7373
DSN=rep1 Transaction Id=2 Transaction Counter=34 Statement Id=1 SQL
statement execution exceeded the QueryThreshold value of 10 seconds.
Statement=select count(*) from a,t;
```

The warning message includes:

- A timestamp: 2007-12-03 15:55:33.67
- The daemon process ID: 9692
- The ID of the process that issued the SQL statement: 7373
- The connection name: `rep1` (following “QT”). (The connection name for the replication agent is RECEIVER.)
- The DSN: `rep1`
- The transaction ID, transaction counter, and statement ID: Transaction Id=2
Transaction Counter=34 Statement Id=1
- The query threshold value in seconds: 10
- The SQL statement: `select count(*) from a,t;`
- The values of any input parameters for the SQL statement

If the SQL statement exceeds the warning message length limit, the statement is continued in subsequent messages. The length of the warning message cannot be configured by the user.

Setting the query threshold

You can specify the number of seconds that a SQL statement can execute before TimesTen writes a warning to the support log and throws an SNMP trap. Execution continues and is not affected by the query threshold.

Set the query threshold by one of the following methods:

- [QueryThreshold general connection attribute](#)
- [Setting the query threshold with ttlsq](#)
- [Setting the query threshold with an ODBC option](#)
- [Setting the query threshold in JDBC](#)
- [Setting the query threshold with TTClasses](#)

For queries executed by the replication agent, see [“Setting the query threshold for replication” on page 22](#).

You cannot set a query threshold for a query that is executed by the cache agent.

QueryThreshold general connection attribute

Use this attribute to write a warning to the support log and throw an SNMP trap when the execution time of a SQL statement exceeds the specified value. For queries executed by the replication agent, see [“Setting the query threshold for replication” on page 22](#). 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 will be issued. The unit is seconds.

Setting

Set QueryThreshold as follows:

Where to set the attribute	How the attribute is represented	Setting
C or Java programs or UNIX ODBC.INI file	QueryThreshold	Nonnegative integer. Default is 0 and indicates that no warning will be issued.
Windows ODBC Data Source Administrator	Query Threshold (secs) field	Nonnegative integer. Default is 0 and indicates that no warnings will be written.

Setting the query threshold with ttisql

By default, the `ttIsql` utility obtains the query threshold setting from the ODBC.INI file. You can override the query threshold value by specifying a value for the query threshold in the connection string:

```
% ttisql -connstr "DSN=mydsn;QueryThreshold=10"
```

The query threshold value of 10 applies to all SQL statements executed by this connection.

You can modify the query threshold value by using the `ttIsql set` command. For example,

```
Command> set querythreshold 20;
```

This sets the query threshold value to 20 for all subsequent SQL statements executed by this connection.

Setting the query threshold with an ODBC option

You can set the query threshold by using the `TT_QUERY_THRESHOLD` option of the `SQLSetConnectOption` and `SQLSetStmtOption` ODBC functions.

By default, the application obtains the query threshold setting from the ODBC.INI file. Setting the `TT_QUERY_THRESHOLD` option in the `SQLSetConnectOption` ODBC function overrides the setting in the ODBC.INI file for the current connection.

Setting the `TT_QUERY_THRESHOLD` option in the `SQLSetStmtOption` ODBC function overrides both the ODBC.INI setting and the `SQLSetConnectOption` setting. It applies to SQL statements executed by this ODBC statement.

Set the query threshold with `SQLSetConnectOption`:

```
RETCODE SQLSetConnectOption(hdbc,  
                             TT_QUERY_THRESHOLD,  
                             seconds)
```

Set the query threshold with `SQLSetStmtOption`:

```
RETCODE SQLSetStmtOption(hdbc,  
                         TT_QUERY_THRESHOLD,  
                         seconds)
```

Setting the query threshold in JDBC

By default, the application obtains the query threshold setting from the ODBC.INI file. You can override the query threshold value for a JDBC Connection by including the `QueryThreshold` attribute in the connection URL for the data store. For example, to set `QueryThreshold` to 5 seconds for *myDSN* data store:

```
jdbc:timesTen:direct:dsn=myDSN;QueryThreshold=5
```

You can also use the `setQueryTimeThreshold` method of the `TimesTenStatement` interface to set the query threshold. This overrides the ODBC.INI setting and the Connection setting. It applies to SQL statements executed by this method. See *Oracle TimesTen In-Memory Database JDBC API Extensions* for more information about using the `setQueryTimeThreshold` method. It is located in `install_dir/doc/ttjava.zip`.

Setting the query threshold with TTClasses

You can set the query threshold by using the `setQueryThreshold` method of the `TTCmd` class:

```
void TTMd::setQueryThreshold(const int seconds, TTStatus &stat)
```


Retrieving the query threshold

You can retrieve the query threshold value by using `ttIsq1`, the `SQLGetConnectOption` or `SQLGetStmtOption` ODBC functions, or `TTClasses`, depending on how the value was set.

Retrieving the QueryThreshold attribute with ttIsq1

You can retrieve the `QueryThreshold` attribute by using `ttIsq1`:

```
Command> show querythreshold;  
Query threshold = 0 seconds (no threshold)
```

Retrieving the TT_QUERY_THRESHOLD ODBC option

You can retrieve the current value of the `TT_QUERY_THRESHOLD` ODBC option by using the `SQLGetConnectOption` or `SQLGetStmtOption` ODBC functions:

```
RETCODE SQLGetConnectOption(hdbc,  
                             TT_QUERY_THRESHOLD,  
                             int seconds)
```

```
RETCODE SQLGetStmtOption(hdbc,  
                          TT_QUERY_THRESHOLD,  
                          int seconds)
```

Retrieving the query threshold with JDBC

You can retrieve the current query threshold by using the `getQueryTimeThreshold()` method of the `TimesTenStatement` interface. See *Oracle TimesTen In-Memory Database JDBC API Extensions*. It is located in `install_dir/doc/ttjava.zip`.

Retrieving the query threshold with TTClasses

You can retrieve the current query threshold by using the `getQueryThreshold` method of the `TTCmd` class:

```
int TTMd::getQueryThreshold(TTStatus &stat)
```

Setting the query threshold for replication

The replication agent does not inherit the QueryThreshold setting from the ODBC.INI file. Use the `ttRepQueryThresholdSet` built-in procedure or the `-repQueryThresholdSet` option of the `ttAdmin` utility to set the query threshold for the replication agent. The query threshold for the replication agent applies to SQL execution on detail tables of materialized views, ON DELETE CASCADE operations and some internal operations that execute SQL statements.

When the query threshold is exceeded, TimesTen writes a warning to the support log and throws an SNMP trap. Execution continues and is not affected by the query threshold.

ttRepQueryThresholdSet

Description This built-in procedure specifies the number of seconds that a query can be executed by the replication agent before TimesTen writes a warning to the support log and throws an SNMP trap. The specified value takes effect the next time the replication agent is started. The query threshold for the replication agent applies to SQL execution on detail tables of materialized views, ON DELETE CASCADE operations and some internal operations that execute SQL statements.

Access Control If Access Control is enabled for your TimesTen instance, this built-in procedure requires ADMIN privileges.

Syntax `ttRepQueryThresholdSet (seconds)`

Parameters `ttRepQueryThresholdSet` has the parameter:

Parameter	Type	Description
<code>seconds</code>	TT_INTEGER NOT NULL	Number of seconds a SQL statement can be executed by the replication agent before TimesTen writes a warning to the support log and throws an SNMP trap. The value must be greater than or equal to 0. Default is 0 and indicates that no warnings will be written.

Result set This built-in procedure returns no results.

Example `Command> call ttRepQueryThresholdSet(4);`

ttAdmin -repQueryThresholdSet

Description This option specifies the number of seconds that a query can be executed by the replication agent before TimesTen writes a warning to the support log and throws an SNMP trap. The specified value takes effect the next time the replication agent is started. The query threshold for the replication agent applies to SQL execution on detail tables of materialized views, ON DELETE CASCADE operations and some internal operations.

The value must be greater than or equal to 0. Default is 0 and indicates that no warning will be sent.

Syntax `ttAdmin [repQueryThresholdSet seconds]
[-connstr connectionString|DSN]`

Result set The `-repQueryThresholdSet` option does not return additional `ttAdmin` output.

Example

```
$ ttAdmin -repQueryThresholdSet 4 myDSN
RAM Residence Policy           : inUse
Replication Agent Policy       : manual
Replication Manually Started   : True
Cache Agent Policy             : manual
Cache Agent Manually Started   : False
```

See also Documentation for the `ttAdmin` utility in *Oracle TimesTen In-Memory Database API Reference Guide*.

Retrieving the query threshold for replication

You can use the `ttRepQueryThresholdGet` built-in procedure or the `-repQueryThresholdGet` option of the `ttAdmin` utility to retrieve the most recently specified replication query threshold.

`ttRepQueryThresholdGet`

Description	This built-in procedure returns the number of seconds that was most recently specified as the query threshold for the replication agent. The number of seconds returned may not be the same as the query threshold in effect. Setting a new value for the query threshold takes effect the next time the replication agent is started.
Access Control	If Access Control is enabled for your TimesTen instance, this built-in procedure requires ADMIN privileges.
Syntax	<code>ttRepQueryThresholdGet</code>
Parameters	This built-in procedure has no parameters.
Result set	This built-in procedure returns the number of seconds that was most recently specified as the query threshold for the replication agent. The number is returned as a <code>TT_INTEGER</code> data type.
Example	<pre>Command> call ttRepQueryThresholdGet; < 4 > 1 row found.</pre>

`ttAdmin -repQueryThresholdGet`

Description	This option returns the number of seconds returns the number of seconds that was most recently specified as the query threshold for the replication agent. The number of seconds returned may not be the same as the query threshold in effect. Setting a new value for the query threshold takes effect the next time the replication agent is started.
Syntax	<code>ttAdmin [-repQueryThresholdGet][--connstr <i>connectionString</i> <i>DSN</i>]</code>
Result set	This option of the <code>ttAdmin</code> utility returns the query threshold in seconds.
Example	<pre>\$ ttAdmin -repQueryThresholdGet myDSN QueryThreshold in seconds : 4 RAM Residence Policy : inUse Replication Agent Policy : manual Replication Manually Started : True</pre>

Cache Agent Policy : manual
Cache Agent Manually Started : False

See also Documentation for the `ttAdmin` utility in *Oracle TimesTen In-Memory Database API Reference Guide*.

Monitoring AWT Cache Group Performance

This chapter includes the following topics:

- [Overview](#)
- [Enabling AWT monitoring](#)
- [Displaying AWT monitoring results](#)

Overview

You can monitor the performance of asynchronous writethrough (AWT) cache groups to determine how much time is spent performing tasks in the AWT workflow. Use the `ttCacheAwtMonitorConfig` procedure to enable monitoring. Use the `ttRepAdmin` utility with the `-awtmoninfo` and `-showstatus` commands to display the monitoring results.

Enabling AWT monitoring

Enable AWT monitoring by calling the `ttCacheAwtMonitorConfig` procedure.

ttCacheAwtMonitorConfig

Description This procedure enables monitoring to determine the amount of time spent in each component of the workflow of an AWT cache group. To display the monitoring results, use the `ttRepAdmin` utility with the `-awtmoninfo` and `-showstatus` commands.

If the replication agent is restarted, monitoring is turned off.

Setting the monitoring state to OFF resets the internal counters of the monitoring tool.

Access Control If Access Control is enabled for your TimesTen instance, this procedure requires ADMIN privileges.

Syntax `ttCacheAwtMonitorConfig('state', samplingFactor)`

Parameters `ttCacheAwtMonitorConfig` has the parameters:

Parameter	Type	Description
<code>state</code>	CHAR (10)	Enables and disables AWT monitoring. Its value can be ON or OFF. Default is OFF.
<code>samplingFactor</code>	TT_INTEGER	Positive integer that specifies the frequency with which the AWT workflow is sampled. If <code>samplingFactor</code> 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 <code>state</code> is set to ON, the default for <code>samplingFactor</code> is 16. If <code>state</code> is set to OFF, the default for <code>samplingFactor</code> is 0.

Result set `ttCacheAwtMonitorConfig` returns the current state and sampling factor. It returns an error if the replication agent is not running or if an AWT cache group has not been created.

Examples

Example 4.1 Retrieve the current state and sampling factor when monitoring is disabled.

```
Command> CALL ttCacheAwtMonitorConfig;
< OFF, 0 >
1 row found.
```

Example 4.2 Enable monitoring and set the sampling frequency to 16.

```
Command> CALL ttCacheAwtMonitorConfig('ON',16);
< ON, 16 >
1 row found.
```


Example 4.3 Disable monitoring.

```
Command> CALL ttCacheAwtMonitorConfig('OFF');  
< OFF, 0 >  
1 row found.
```

Displaying AWT monitoring results

Use the `ttRepAdmin` utility with the `-awtmoninfo` and `-showstatus` commands to display monitoring information for an AWT cache group that has monitoring enabled.

ttRepAdmin -showstatus -awtmoninfo

Description If you have enabled monitoring for AWT cache groups by calling the `ttCacheAwtMonitorConfig` procedure, you can display the monitoring results by using the `-awtmoninfo` option with the `ttRepAdmin -showstatus` command.

Syntax `ttRepAdmin -showstatus -awtmoninfo {DSN | -connStr
connectionString}`

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 management 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 Oracle.
- 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 Oracle.
- 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 Oracle.

The output also includes the percentage of time spent on TimesTen processing, Oracle bookmark management, Oracle execution and Oracle commits.

Example `ttRepAdmin -showstatus -awtmoninfo myDSN`
[other -showstatus output]
...
AWT Monitoring statistics

TimesTen processing time : 0.689000 millisecs (0.164307 %)
Oracle bookmark management 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

See also Documentation for the `ttRepAdmin` utility in *Oracle TimesTen In-Memory Database API Reference Guide*.

Cleaning Up Autorefresh Objects on Oracle

This chapter includes the following topics:

- [Overview](#)
- [Using the autorefresh cleanup script](#)

Overview

TimesTen automatically creates a trigger and a log table for each Oracle base table specified in a READONLY cache group or a cache group with the AUTOREFRESH INCREMENTAL attribute when the cache group is created. The objects are dropped when the cache group is dropped. For more information about these Oracle objects, see “Managing Oracle objects for READONLY, AUTOREFRESH, and AWT cache groups” in *TimesTen Cache Connect to Oracle Guide*.

If a TimesTen data store becomes unavailable without dropping the cache groups, autorefresh objects continue to exist on Oracle. A TimesTen data store is unavailable, for example, when its host machine is taken permanently offline or the data store has been destroyed. Autorefresh objects also continue to exist when a TimesTen data store is no longer being used but its cache groups have not been dropped. Change log records continue to accumulate in the change log tables. This affects autorefresh performance for active data stores. Thus it is desirable to clean up the Oracle objects associated with the unavailable or abandoned data store.

TimesTen provides a script that can be used to clean up the autorefresh objects in the Oracle database. The location of the script is:

```
install_dir/bin/autorefreshCleanup.sql
```

The script removes autorefresh objects related to the data store and host that you specify when you run the script. It does not affect autorefresh objects for other data stores.

Using the autorefresh cleanup script

Use SQL*Plus to run the autorefresh cleanup script on the Oracle database. You must be connected to the Oracle database as a cache administration user to run the script.

The autorefresh cleanup script requires two input parameters: the host name where TimesTen is running and the data store path name. The strings for the host name and the data store path name must be identical to the strings stored on Oracle. You can run the `autorefreshChangeLogInfo.sql` script, located in `install_dir/bin`, to determine the correct host name and data store path name. On Windows, enter the host name and data store name in lower case.

Pass the host name and the data store path name to the script from the command line or from a SQL*Plus session:

- From the command line:

```
$ sqlplus cache_admin_uid@oracle_id @install_dir/bin/autorefreshCleanUp.sql  
"host_name" "data_store_path_name"
```

- From a SQL*Plus session:

```
SQL> @install_dir/bin/autorefreshCleanUp.sql "host_name"  
"data_store_path_name"
```

If you do not specify the parameters, the script prompts for them. “Parameter 1” is the host name; “parameter 2” is the data store path name.

The script displays the SQL that it executes on Oracle to clean up the autorefresh objects.

Example 5.1

In this sample output, the Oracle table that is referenced by the trigger and log table is still being cached by at least one incremental autorefresh cache group. Note that the entry for the specified host and data store is being deleted, but the trigger and the log table are not dropped.

```
Performing cleanup for object_id: 91771 which belongs to table :  
TTUSER.MYTABLE  
Executing: delete from tt_03_agent_status where host = my-pc and  
datastore = c:\data\rep2 and object_id = 91771  
Executing: update tt_03_user_count set usercount =  
:usecount,usecount = 1
```

Example 5.2

In this sample output, the Oracle table that is referenced by the trigger and log table is not cached by any incremental autorefresh cache group. Note that the log table and the trigger are dropped.

```
Performing cleanup for object_id: 83560 which belongs to table :  
TTUSER.MYTABLE  
Executing: delete from tt_03_agent_status where host = my-pc and  
datastore = c:\data\tt60 and object_id = 83560
```

```
Executing: drop table tt_03_83560_L  
Executing: drop trigger tt_03_83560_T  
Executing: delete from tt_03_user_count where object_id =  
object_id1
```


Detection of Dual Active Masters in an Active Standby Pair

Ordinarily, the designation of the active master and standby master data stores in an active standby pair replication scheme is explicitly controlled by the user (see “Administering an Active Standby Pair” in the *TimesTen to TimesTen Replication Guide*). However, in some circumstances the user may not have the ability to modify both the active and standby master data stores when changing the role of the standby master data store to active.

For example, if network communication to the site of an active master data store is interrupted, the user may need the standby master data store at a different site to take over the role of the active, but cannot stop replication on the current active master or change its role manually. Changing the standby master data store to active without first stopping replication on the active master leads to a situation where *both* masters are in the ACTIVE state and accepting transactions. In such a scenario, TimesTen can automatically negotiate the active/standby role of the master data stores when network communication between the stores is restored.

If, during the initial handshake between the data stores, TimesTen determines that the master data stores in an active standby pair replication scheme are both in the state ACTIVE, TimesTen performs the following operations automatically:

- The data store which was set to the state ACTIVE the *most* recently is left in the ACTIVE state and may continue to be connected to and updated by applications.
- The data store which was set to the state ACTIVE the *least* recently is invalidated. All applications are disconnected.
- When the invalidated data store comes back up, TimesTen determines whether any transactions have occurred on the data store that have not yet been replicated to the other master data store. If such transactions have occurred, they are now trapped, and the data store is left in the IDLE state. The data store needs to be duplicated from the active master in order to become a standby master. If there are no trapped transactions, the data store is safe to use as a standby master data store and is automatically set to the STANDBY state.

Transaction Log API (XLA)

In this release, the `ttXlaUpdateDesc_t` header in an XLA update record now includes a flag that indicates cascade and aging deletes on a row.

The flag `TT_CASCDEL` indicate that the XLA update was generated as part of a cascade delete operation and `TT_AGING` as part of an aging operation.

The header file now contains the following flags:

```
/*
 * Values for ttXlaUpdateDesc_t.flags bits
 */
#define TT_UPDCOMMIT 1 /* Implied commit on this record */
#define TT_UPDFIRST 2 /* First record in transaction */
#define TT_UPDREPL 4 /* Record generated by replication */
#define TT_UPDCOLS 8 /* List of updated columns present */
#define TT_UPDINT1 16 /* Internal Use */
#define TT_UPDINT2 32 /* Internal Use */
#define TT_UPDDEFAULT 64 /* Default column values present */
#define TT_UPDINT3 128 /* Internal Use */
#define TT_CASCDEL 256 /* Cascade delete */
#define TT_AGING 512 /* Aging delete */
```


New Error Messages and SNMP Traps

This chapter contains a list and descriptions of error messages that have been added in this release. For a list of all other error messages, see [Oracle TimesTen In-Memory Database Error Messages and SNMP Traps](#).

Error messages

Error number	Error or warning message	Details
2819	Invalid value ' <i>value</i> ' for argument <i>number</i> for procedure <i>name</i> . The valid values are <i><values></i> .	You have passed an invalid value for a parameter in the position indicated in when calling a built-in procedure. The position of the invalid parameter, the invalid value and a list of valid values are displayed. Call the built-in procedure passing a valid value for the parameter.
12252	Could not check Oracle DB	An additional error message from the cache layer accompanies this message. Please review that message to determine the problem and fix.
12253	The Active Standby Oracle Database is the same as the DR Oracle Database	When the ttRepAdmin -duplicate -initCacheDR option is used the DR Oracle Database cannot be the same database as the Active/Standby Oracle subscribing database. The check has identified a tt_03_reppeers entry which is from one of the Active Standby master stores to the DR Oracle Database. Reconfigure the data recovery to resolve this conflict.

Error number	Error or warning message	Details
16268	Failed to retrieve aging configuration for table <owner>.tablename>	This is part of replication. The first time the transmitter sends a transaction for a particular table, TimesTen sends the table definition. This message indicates something went wrong while trying to retrieve the aging information. There is an additional error message detailing the specific failure. Please review that message to determine the problem and fix.
16269	In an Active/Standby scenario, AWT Monitoring should be enabled on the Standby node	In an active standby pair with an AWT cache group, monitoring gives relevant results only on the standby node, which does the propagation of the workload to the Oracle Database. Enabling AWT monitoring on the active node does not record any monitoring information. If the standby node fails and if the active node takes over, calling the ttCacheAwtMonitorConfig built-in procedure automatically enables monitoring the AWT statistics.
16270	AWT Monitoring should not be enabled on the Subscriber node in an Active/Standby scenario	In an active standby pair with subscribers and an AWT cache group, monitoring on a subscriber does not reveal any useful information. Call the ttCacheAwtMonitorConfig built-in procedure to disable monitoring the AWT statistics.
16271	Monitoring could not be configured because replication agent does not have an active AWT thread	This error is thrown if the replication agent is either invalid or in an initialization phase. If the AWT receiver is killed due to an unexpected situation, the replication agent is in an invalid state. It is also possible that the AWT receiver has not been started by the replication agent, in which case it is in an initialization phase. Resolution for this error is to call the ttCacheAwtMonitorConfig built-in procedure again, with the state parameter set to "ON."

Error number	Error or warning message	Details
17041	Failed to create Oracle DR scheme	This is a generic error, there is an accompanying detailed error message concerning the actual failure. Please review that message to determine the problem and the fix.
17042	A Cache DR scheme can only be created on an ACTIVE STANDBY SUBSCRIBER	The ttRepAdmin -duplicate -initCacheDR option can only be used when duplicating to an Active Standby subscriber. You have attempted to create a disaster recovery scheme on a data store that is not involved in Active/Standby replication. This is not supported.
17043	Failed to initialize Oracle DR tables	The initialization of the Oracle Disaster Recovery tables failed. In the user error log, review any error messages that were returned during the operation in which the error occurred for more details. Destroy the destination data store and attempt to re-initialize the disaster recovery process.
17044	Exited early because could not get cache group information	The disaster recovery process exited early because it could not get required information. In the user error log, review any error messages that were returned during the operation in which the error occurred for more details. Destroy the destination data store and attempt to re-initialize the disaster recovery process.
17045	Could not truncate cached tables on Oracle DR site	During disaster recovery, the replication agent could not truncate any cached tables. In the user error log, review any error messages that were returned during the operation in which the error occurred for more details. Destroy the destination data store and attempt to re-initialize the disaster recovery process.

Error number	Error or warning message	Details
17046	Could not flush data to Oracle DR site	During disaster recovery, the replication agent could not flush data to Oracle. In the user error log, review any error messages that were returned during the operation in which the error occurred for more details. Review any error messages that were returned during the operation in which the error occurred for more details. Destroy the destination data store and attempt to re-initialize the disaster recovery process.
17047	Could not commit data to the Oracle DR site	During disaster recovery, the replication agent could not commit data to Oracle. In the user error log, review any error messages that were returned during the operation in which the error occurred for more details. Destroy the destination data store and attempt to re-initialize the disaster recovery process.
17048	Could not create thread to initialize the Oracle DR site	During disaster recovery, the replication agent could not create the thread to initialize Oracle. In the user error log, review any error messages that were returned during the operation in which the error occurred for more details. Destroy the destination data store and attempt to re-initialize the disaster recovery process.
17049	Error invalidating store due to scheme inconsistency	To correct this problem, verify that the schemes involved in this operation are identical.
17050	Conflict Replication Reporting Starting	Conflict reporting suspend/resume is configured on your system and the rate of conflict rate has fallen below the resume value. Reports are now being sent. This is an informational message.

Error number	Error or warning message	Details
17051	Conflict Replication Reporting Stopping	Conflict reporting suspend/resume is configured on your system and the rate of conflict rate exceeds the suspend value. Reports are now suspended. This is an informational message.
47136	The cache agent hasn't fully initialized yet. The cache group <cache_group_name> will be refreshed after initialization is complete.	This is an informational message. A CREATE or ALTER CACHE GROUP statement was issued before the cache agent had a chance to initialize. The CREATE or ALTER operation takes effect after the initialization completes. No action needed from user.

SNMP traps

The following new SNMP trap is added in this release.

Error or warning message	Error, Warning or Information	Details
ttQueryThresholdWarnTrap	Warning	A SQL query exceeded the user-specified threshold. The text of the query can be found in the User Log message. The Transaction ID and the Statement ID of the query can be found both in the trap and the User Log message. After issuing the trap, the query continues executing.

Index

A

AIX 4

autorefresh cleanup script 34

C

Cache Connect to Oracle support 5

ccNUMA 6

D

daemon options

 shmLocalityHint 6

H

HP-UX 3, 4

J

JDBC

 setting query threshold 19

JDK support 3

L

Linux, MontaVista Carrier Grade 4

Linux, SUSE 4, 5

M

Microsoft Windows 4

MontaVista Carrier Grade Linux 4

Q

query threshold

 setting with ttIsql 19

 using JDBC 19

QueryThreshold general connection attribute 16, 18

S

setting query threshold

 TTClasses 20

shared memory daemon option 6

shmget 6

shmLocalityHint daemon option 6

Solaris 4

SQLSetConnectOption ODBC function

 setting the query threshold 19

SQLSetStmtOption ODBC function

 setting the query threshold 19

SUSE LINUX 4, 5

T

Tru64 4

TT_QUERY_THRESHOLD ODBC option 19

ttAdmin utility

 -repQueryThresholdGet option 16

 -repQueryThresholdSet option 16

TTClasses

 setting query threshold 20

ttQueryThresholdWarnTrap SNMP trap 17

ttRepQueryThresholdGet procedure 16

ttRepQueryThresholdSet built-in procedure 16

ttXlaUpdateDesc_t 39

V

Veritas file system options 5

