Oracle® TimesTen In-Memory Database

Installation Guide

11*g* Release 2 (11.2.2)

E21632-17

October 2015

ORACLE®

Oracle TimesTen In-Memory Database Installation Guide, 11*g* Release 2 (11.2.2)

E21632-17

Contents

2 Migration, Backup, and Restoration

3 TimesTen Upgrades

A   Oracle TimesTen In-Memory Database License Information

Index

Preface

Oracle TimesTen In-Memory Database (TimesTen) is a relational database that is memory-optimized for fast response and throughput. The database resides entirely in memory at runtime and is persisted to disk storage for the ability to recover and restart. Replication features allow high availability. TimesTen supports standard application interfaces JDBC, ODBC, and ODP.NET, in addition to Oracle interfaces PL/SQL, OCI, and Pro*C/C++. TimesTen is available separately or as a cache for Oracle Database.

For the latest release notes in your TimesTen distribution, refer to the `readme.html` file in the installation directory. (The version in the documentation library may not be as current.)

Related documents

TimesTen documentation is available on the product distribution media and on the Oracle Technology Network:

http://www.oracle.com/technetwork/database/database-technologies/timesten/documentation/index.html

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 and Linux platforms. Refer to the "Platforms" section in *Oracle TimesTen In-Memory Database Release Notes* 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.

Convention	Meaning
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: `Driver=install_dir/lib/libtten.sl` Replace `install_dir` with the path of your TimesTen installation 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 multiple arguments on a single command line. An ellipsis in a code example indicates that what is shown is only a partial example.
%	The percent sign indicates the UNIX shell prompt.

In addition, TimesTen documentation uses the following special conventions:

Convention	Meaning
install_dir	The path that represents the directory where TimesTen is installed.
TTinstance	The instance name for your specific installation of TimesTen. Each installation of TimesTen must be identified at installation time with a unique instance name. This name appears in the installation path.
bits or *bb*	Two digits, either 32 or 64, that represent either a 32-bit or 64-bit operating system.
release or *rr*	The first three parts in a release number with or without dots. The first three parts of a release number represent a major TimesTen release. For example, 1122 or 11.2.2 represents TimesTen 11*g* Release 2 (11.2.2).
jdk_ver	One or two digits that represent the version number of a major JDK release. For example, 14 is for JDK 1.4 and 5 is for JDK 5.0.
DSN	TimesTen data source name (for the TimesTen database).

Documentation Accessibility

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

Access to Oracle Support

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

What's New

This section summarizes the new features of Oracle TimesTen In-Memory Database release 11.2.2 that are documented in this guide and provides links to more information.

New features in Release 11.2.2.3.0

Windows 7 support

Windows 7 is certified for use with TimesTen. As of this release of the documentation, where features or procedures differ from other Windows platforms, information specific to Windows 7 has been added to this document.

Windows 32-bit client with 64-bit installer

Beginning with this release, the Windows 64-bit installer includes an option for a 32-bit client as well as a 64-bit client. You can install both (default), either, or neither. See "Installing TimesTen" on page 1-23.

New features in Release 11.2.2.1.0

Exalogic Support

TimesTen is supported on Oracle Exalogic Elastic Cloud (Exalogic). The Exalogic certification was performed using the TimesTen with a quarter-rack Exalogic release 1.0 server. For details on installing TimesTen on Exalogic, see "Installing TimesTen on Exalogic systems" on page 1-31.

Migration

This guide includes a chapter on migrating TimesTen databases. In previous releases, this information was found in the *Oracle TimesTen In-Memory Database Operations Guide*. See Chapter 2, "Migration, Backup, and Restoration."

1

TimesTen Installation

This chapter contains configuration information that you should review before installing TimesTen on your system, in the sections:

- Installation instances

- Choosing the appropriate TimesTen components

- Installation prerequisites

- Default installation directories

- Considerations for locations of database files and other user files

- Pre-Install requirements for operating system group and file permissions

You can find a description of the procedures to install, configure and uninstall TimesTen:

- Installing TimesTen on UNIX systems

- Installing TimesTen on Windows systems

- Installing TimesTen Client on Mac OS X systems

- Installing TimesTen on Exalogic systems

> **Note:**   This document does not discuss TimesTen installation on an Exalytics system. Go to the following location for Exalytics documentation:
>
> http://www.oracle.com/technetwork/documentation/exalytics-do
> c-1645656.html

This chapter also contains information to help you further configure TimesTen after installation, work with the demo applications, migrate databases to this release, and view the TimesTen documentation:

- ODBC installation

- Environment variables

- TimesTen Quick Start

- Documentation installation

This chapter contains information that helps you troubleshoot problems that may arise during the installation process and a link to licensing information:

- Installation problems

Installation instances

On UNIX, you can install multiple instances of TimesTen, where an instance is essentially the collection of files and processes related to a TimesTen installation. On Windows, you can install only one instance of any major TimesTen release, where a major release is indicated by the first three parts of the release number, such as 11.2.2. For example, you can install both 11.2.1.9.0 and 11.2.2.5.0 on the same Windows computer, but you cannot install both 11.2.2.0.0 and 11.2.2.5.0.

The TimesTen product can be installed onto local, private disk storage, such that each computer has a private copy of the entire TimesTen installation. Installing a single copy of the TimesTen software onto a shared storage location and then sharing this copy among several computers will not work.

The following sections provide more information about the TimesTen installation instance:

- Instance naming
- Instance port numbers

Instance naming

The instance name is the key used to access all necessary information about an installation of TimesTen.

On UNIX systems, by default, the instance name for this release is `tt1122_32` on 32-bit installations and `tt1122_64` on 64-bit installations. The instance name is case-insensitive, at least one alphanumeric character and up to 255 characters. The name can include underscores (_) or periods (.), but no other special characters.

To install a second instance of the same TimesTen release on UNIX, you must supply a unique instance name and port number. The TimesTen installer can detect if an instance of the particular release of TimesTen exists on the system and prompts you for a new instance name and port number for the main TimesTen daemon.

The instance name on Windows is `tt1122_32` on 32-bit installations and `tt1122_64` on 64-bit installations. Only one instance of any major TimesTen release (such as 11.2.2) can exist on a Windows system.

Instance port numbers

When you install multiple instances of TimesTen on the same computer, specify a unique TCP/IP port number for each TimesTen daemon and each TimesTen server during the installation.

> **Note:** If you will be using replication, however, be aware that if the daemon ports are different for TimesTen databases that replicate to each other, you must use static ports for replication and you must use the `ttRepAdmin` utility `-remoteDaemonPort` option for duplicate operations. You can use the `ttVersion` utility to verify the port number of your installation of TimesTen.

Choosing the appropriate TimesTen components

The TimesTen installer enables you to select the components of TimesTen that you want to install.

- Components available on UNIX
- Components available on Windows

Components available on UNIX

On UNIX, you can install the following components. In addition, the installation script prompts you to optionally install the TimesTen Quick Start sample programs, and the TimesTen documentation (including Quick Start tutorial pages).

Type	Description
TimesTen Client	Installs the TimesTen Client only. No other TimesTen components are installed. Use this installation to allow the TimesTen Client to access the TimesTen Server from another installation.
TimesTen Data Manager	Installs the TimesTen Data Manager only. Use this installation to run the TimesTen Data Manager locally.
TimesTen Client, Server and Data Manager	Installs the TimesTen Data Manager, Client, and Server. Use this installation to perform the following: ■ Allow a Client from another installation to access the TimesTen Server. ■ Allow the TimesTen Client to access either this TimesTen Server installation or another TimesTen Server installation. ■ Allow applications to access the TimesTen Data Manager locally.

If you have installed TimesTen and you would like to add or remove components, you must run the installer and select the option "Upgrade an existing instance," and then select the instance that you would like to change.

Components available on Windows

On Windows, you can specify the components to install by selecting or deselecting the appropriate checkboxes during installation. By default, all components are installed except the TimesTen Data Manager Debug Libraries.

Type	Description
TimesTen Data Manager	Installs the TimesTen Data Manager. Use this installation to run the TimesTen Data Manager locally.
TimesTen Data Manager Debug Libraries	Installs the TimesTen Data Manager debug libraries. Used particularly during the development phase to allow you to debug problems that may occur. By default, the debug libraries are not installed.
TimesTen Server	Installs the TimesTen Data Server and TimesTen Data Manager.
TimesTen Client	Installs the TimesTen Client, which you can use to access the TimesTen Server on a remote computer. **Note**: In the Windows 64-bit installer, there are options for the 64-bit client and the 32-bit client. You can install both (default), either, or neither.
TimesTen Quick Start	Installs the TimesTen Quick Start sample programs.
TimesTen Documentation	Installs the TimesTen Documentation Library and Quick Start tutorial pages.

Installation prerequisites

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

On both UNIX and Windows platforms where JDBC is supported you must have an appropriate version of the JDK installed on your computer to use JDBC.

In addition, be aware that if you use automatic client failover, to accommodate a firewall you must specify a UDP port or port range. (Also see "Configuring automatic client failover" in *Oracle TimesTen In-Memory Database Operations Guide*.)

> **Important:**   Do not install TimesTen as an operating system user whose name matches any of the TimesTen predefined internal users: GRID, PUBLIC, SYS, SYSTEM, or TTREP. If you install TimesTen as a user with one of these names, attempts to create a database fail with a unique constraint violation error.

This section also discusses the platform-specific prerequisites:

- General UNIX requirements
- AIX prerequisites
- HP-UX prerequisites
- Linux prerequisites
- Solaris prerequisites
- Windows requirements

General UNIX requirements

In general, on UNIX systems, you must configure the following:

- The number of semaphores
- Allowable shared memory

In addition, you may need to perform the following:

- Ensure you have the latest operating system patches.
- Configure your file system to allow large files.
- Configure your Java environment.
- Configure your Client/Server environment.
- Configure network settings for replication.

The following sections outline some changes that may be required on any UNIX system. In addition, some of these sections describe changes required for each specific UNIX platform on which TimesTen is supported.

- Veritas file system options
- Semaphores (general UNIX)
- Java

Veritas file system options

On the Veritas file system, if you plan to have TimesTen applications that use `DurableCommits=1`, use the `mincache=direct` and `convosync=direct` options to ensure durability.

Options that convert `dsync` into `sync` or `fdatasync` into `sync` or those that treat all writes such that the file is opened with `O_SYNC` should be avoided.

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

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
```

Semaphores (general UNIX)

TimesTen uses one semaphore set per active database.

In addition, where Client/Server is used with shared memory, there is one more semaphore set for each `ttcserver` process used.

Each active database uses 155 semaphores plus one for each connection.

For platform-specific semaphore information, see "Semaphores (HP-UX)" on page 1-6, "Semaphores (Linux)" on page 1-10, and "Create a project (Solaris)" on page 1-11. In AIX, semaphores are configured dynamically by the kernel.

For related information about the TimesTen `Connections` attribute, which specifies the maximum number of concurrent database connections, see "Connections" in *Oracle TimesTen In-Memory Database Reference*.

Java

If you use JDBC, install the latest JDK and any vendor required patches. Refer to the website of the JDK provider for the patches you may need.

To run 64-bit Java applications on all systems except AIX systems, if you are using the 64-bit JVM, you may need to pass the `-d64` option on the `java` command line.

AIX prerequisites

This section covers AIX prerequisites:

- Large pages (AIX)

Large pages (AIX)

On AIX systems with the required patch levels, TimesTen can use large pages. Using large pages locks the shared segment into memory so it cannot be paged. Users must have the `CAP_BYPASS_RAC_VMM` and `CAP_PROPAGATE` capabilities. The capabilities are granted by a `root` user by editing the `/etc/security/user` file or for locally authenticated users with:

```
# chuser capabilities=CAP_BYPASS_RAC_VMM,CAP_PROPAGATE user_id
```

The system default is to not have any memory allocated to the large page physical memory pool. You can use the vmo command to configure the size of the large page

physical memory pool. The following example allocates 4 GB to the large page physical memory pool.

```
# vmo -r -o lgpg_regions=256 -o lgpg_size=16777216
```

To use large pages for shared memory, you must enable the SHM_PIN shmget() system call with the following command, which persists across system restarts:

```
# vmo -p -o v_pinshm=1
```

HP-UX prerequisites

On HP-UX, the following sections describe steps to perform before installation to improve the performance of TimesTen:

- Semaphores (HP-UX)

- Shared memory (HP-UX)

- Large databases (HP-UX)

- Replication (HP-UX)

For HP-UX 11*iv*2, log in as root and use kctune to check or change kernel parameter settings:

```
# /usr/sbin/kctune
```

Or, alternatively, you can use the **HP System Administration Manager**.

Semaphores (HP-UX)

On HP-UX systems, to connect to more than two databases simultaneously, you typically must increase the value of the kernel parameter semmns. Use a value that allows for the number of expected user connections plus 155, for each database. Refer to "Semaphores (general UNIX)" on page 1-5 for related information.

For example, for two databases with 200 users each, use a setting of at least 2 x (200 + 155) = 710.

Shared memory (HP-UX)

On HP-UX systems, check the value of the parameter shmmax. HP-UX shared memory must be configured so that the maximum size of a shared memory segment is large enough to contain the TimesTen main shared memory segment. Calculate the size of this segment as follows:

```
PermSize + TempSize + LogBufMB + 64 MB
```

These are TimesTen connection attributes that set the sizes of the TimesTen permanent memory region, temporary memory region, and log buffer. (Refer to "Connection Attributes" in *Oracle TimesTen In-Memory Database Reference* for information.) There is also an allowance for 64 MB of overhead, including the database header.

Large databases (HP-UX)

On 64-bit HP-UX systems, if you expect to have databases that are larger than 2 GB, you must enable large files. By default, HP-UX supports files that are no greater than 2 GB.

To enable large files, create the file systems using the newfs command with the -o largefiles option or alter the file systems using the fsadm command with the -o

largefiles option. The following fsadm command alters the file system to enable large files:

```
% /usr/sbin/fsadm -F fstype -o largefiles device_name
```

For example:

```
% /usr/sbin/fsadm -F hfs -o largefiles /dev/vg02/rlvol1
```

Replication (HP-UX)

For replication, TCP send and receive buffers should be increased to a minimum of 512 KB. You may need to embed the following commands into a script that can be run at system boot time.

For HP-UX 11.23 (11*iv*2):

```
# /usr/bin/ndd -set /dev/tcp tcp_xmit_hiwater_lfp 524288
# /usr/bin/ndd -set /dev/tcp tcp_recv_hiwater_lfp 524288
# /usr/bin/ndd -set /dev/tcp tcp_xmit_hiwater_lnp 524288
# /usr/bin/ndd -set /dev/tcp tcp_recv_hiwater_lnp 524288
# /usr/bin/ndd -set /dev/tcp tcp_xmit_hiwater_max 524288
# /usr/bin/ndd -set /dev/tcp tcp_recv_hiwater_max 524288
```

Linux prerequisites

For Linux, TimesTen has been tested with releases of Oracle Linux, Asianux, Red Hat Enterprise Linux, the MontaVista Linux Carrier Grade Edition, and SuSE LINUX Enterprise Server. Consult *Oracle TimesTen In-Memory Database Release Notes* for details about supported releases.

> **Important:** Because Oracle Database does not support MontaVista Linux Carrier Grade Edition, there are significant restrictions to TimesTen functionality on that platform. See *Oracle TimesTen In-Memory Database Release Notes* for details.

The following sections describe steps to perform before installing TimesTen on Linux systems:

- Shared memory (Linux)
- Large pages (Linux)
- Semaphores (Linux)
- IPC Client/Server (Linux)
- Required library packages (Monta Vista)

Shared memory (Linux)

Linux shared memory must be configured so that the maximum size of a shared memory segment is large enough to contain the TimesTen main shared memory segment. Calculate the size of this segment as follows:

```
PermSize + TempSize + LogBufMB + 64 MB
```

These are TimesTen connection attributes that set the sizes of the TimesTen permanent memory region, temporary memory region, and log buffer. (Refer to "Connection Attributes" in *Oracle TimesTen In-Memory Database Reference* for information.) There is also an allowance for 64 MB of overhead, including the database header.

To increase the maximum size of a shared memory segment, log in as `root` and edit the `/etc/sysctl.conf` file to update the `shmmax` memory kernel parameter. This value is in bytes. You may also need to increase the value of the `shmall` memory kernel parameter, which sets the total size of shared memory segments systemwide. This value is in pages, where the page size is typically 4 KB on Linux x86 and x8664 systems. As a minimum, where one memory segment is required, `shmall` should set a size of at least `shmmax/page_size` (converting `page_size` to 4096 bytes).

Consider the following example, where the TimesTen database size (main shared memory segment size) is 64 GB and no other shared memory segments are required.

```
kernel.shmmax=68719476736
kernel.shmall=16777216
```

(64 x 1024 x 1024 x 1024 = 68719476736 and 68719476736 / 4096 = 16777216.)

> **Note:** The preceding discussion and example do not consider the PL/SQL shared memory segment in calculating the desired `shmall` value. The size of the PL/SQL segment is determined by the `PLSQL_MEMORY_SIZE` connection attribute and is 32 MB by default. Given this relatively small size, there is usually enough unused overhead in the main shared memory segment to allow the PL/SQL segment to fit within the `shmall` systemwide limit when calculated as above.

To set these values without restarting the system:

```
% /sbin/sysctl -p
```

You can use the following command to view the values of `max seg size` and `max total shared memory`:

```
% ipcs -lm
```

These values reflect the `shmmax` and `shmall` settings, respectively, but be aware that both of these values are in KB.

Large pages (Linux)

It is advisable to configure large pages, referred to as HugePages in Linux, for more efficient memory management if this is supported on your system. Use of HugePages is required if the size of the TimesTen main shared memory segment is greater than 256 GB.

If there is support for HugePages, values shown by the following command will include HugePages parameters such as `HugePages_Total` and `HugePages_Free`:

```
% cat /proc/meminfo
```

To configure HugePages, you must know or obtain the size of the TimesTen main shared memory segment, the HugePages size on your system, and the group ID of the user (typically the instance administrator) who is running the TimesTen main daemon process:

- See the preceding section, "Shared memory (Linux)" on page 1-7, for information about the main shared memory segment.

- HugePages size is typically 2 MB. Refer to `Hugepagesize` (expressed in KB) from the `meminfo` output to confirm.

- You can use the `id -g` command to obtain the group ID of the instance administrator or other applicable user. For example, for the instance administrator `myadmin`:

```
% id -g myadmin
```

Complete these steps to set up HugePages:

1. The `root` user sets the following HugePages values in the `/etc/sysctl.conf` file:

- `vm.nr_hugepages`: Number of HugePages required

  Calculate this by dividing the TimesTen main shared memory segment size by the HugePages size.

  (Setting `/proc/sys/vm/nr_hugepages` is equivalent, but not persistent and therefore not recommended.)

- `vm.hugetlb_shm_group`: Group ID of the user

Consider an example where the TimesTen main shared memory segment size is 64 GB (65536 MB), the HugePages size is 2 MB, and the group ID of the instance administrator is 900.

```
vm.nr_hugepages = 32768
vm.hugetlb_shm_group = 900
```

When you are done editing `sysctl.conf`, execute the following command to make the changes dynamically and immediately. (Otherwise, edits to `sysctl.conf` do not take effect until the system is restarted.)

```
% /sbin/sysctl -p
```

The preceding example would result in the following values from `meminfo` (2048 KB being equivalent to 2 MB):

```
HugePages_Total: 32768
HugePages_Free: 32768
Hugepagesize: 2048 kB
```

As TimesTen uses HugePages, the `HugePages_Free` value changes accordingly.

2. The `root` user configures (or confirms) `memlock` entries in the `/etc/security/limits.conf` file for the instance administrator or applicable user. These entries, expressed in KB, control the amount of memory that the indicated users can lock. Set both `hard memlock` and `soft memlock`. For a 64-GB main shared memory segment and instance administrator `myadmin`:

```
myadmin soft    memlock    67108864
myadmin hard    memlock    67108864
```

(64 x 1024 x 1024 = 67108864.)

The user must log out and log in again for changes to take effect.

After installing TimesTen   After you have installed TimesTen, enable HugePages in TimesTen by setting `-linuxLargePageAlignment` *Size_in_MB* in the daemon options file, *install_dir*/srv/info/ttendaemon.options. For a HugePages size of 2 MB:

```
-linuxLargePageAlignment 2
```

Run the `ttStatus` utility to confirm that TimesTen is using HugePages. The output should include an entry such as the following.

```
Shared Memory KEY 0x1202031f ID 2457616 (LARGE PAGES, LOCKED)
```

If you change the HugePages configuration after installing and using TimesTen, you must take the following steps for the new configuration to take effect:

1. Unload the database from memory. See "Unloading a database from memory" on page 3-9.

2. Shut down the TimesTen daemon:

   ```
   % ttDaemonAdmin -stop
   ```

   > **Note:** Or use `ttDaemonAdmin -restart` instead of `-stop` and `-start`.

3. Restart the daemon:

   ```
   % ttDaemonAdmin -start
   ```

4. Reload the database into memory. See "Reloading a database into memory" on page 3-11.

   > **Notes:**
   >
   > ■ Because HugePages must be allocated in contiguous available memory space, the requested allocation may not be granted, or may be only partially granted, until after the system is restarted. Check values from `meminfo` as shown above to confirm the allocation. Restarting will grant you the full allocation, assuming enough memory is available in the system.
   >
   > ■ If a TimesTen database less than or equal to 256 GB does not fit into the available HugePages space, regular pages will be used instead. If a database greater than 256 GB does not fit, there will be an error.
   >
   > ■ The TimesTen PL/SQL shared memory segment typically uses HugePages if the feature is configured and there is enough space available, regardless of whether the database does. The PL/SQL shared memory segment has a relatively small size, determined by the `PLSQL_MEMORY_SIZE` connection attribute.
   >
   > ■ On Linux, the HugePages segment is automatically locked, so is not a candidate to be swapped to disk. Therefore, no setting of the TimesTen `MemoryLock` connection attribute is required.

Semaphores (Linux)

Also see "Semaphores (general UNIX)" on page 1-5.

On Linux, you can use the following formula as a guide for semaphores, although in practice, `SEMMNS` can be much less than `SEMMNI * SEMMSL`.

```
SEMMNS = (SEMMNI * SEMMSL)
```

To view existing kernel parameter settings, log in as `root` and use:

```
# /sbin/sysctl -a
```

On Linux systems, the first parameter of kernel.sem must be a minimum of 128. TimesTen uses 155 SEMMSL, plus one for each active connection. You must increase the kernel parameter settings if you plan to use many connections.

For example, if you plan to use 200 connections, specify a kernel.sem setting such as the following in the /etc/sysctl.conf file:

```
kernel.sem = 355 32000 100 128
```

(155 + 200 = 355.)

The first parameter, the maximum number of semaphores per array (SEMMSL), is the most significant for this discussion. This value is related to the maximum number of connections, with some allowance for TimesTen internal connections. The second parameter is maximum semaphores systemwide (SEMMNS), the third parameter is maximum operations per semop call (SEMOPM), and the fourth parameter is maximum arrays (SEMNI).

Then restart the system or run the following command:

```
# /sbin/sysctl -p
```

IPC Client/Server (Linux)

On Linux systems, to enable a given number of ShmIpc Client/Server connections, you must have an appropriate SEMMSL value in the kernel.sem setting in the /etc/sysctl.conf file. For details, see the preceding section, "Semaphores (Linux)".

Then restart the system or run the following command.

```
# /sbin/sysctl -p
```

Required library packages (Monta Vista)

For TimesTen to work properly on MontaVista Linux, include the libaio and libelf library packages.

Solaris prerequisites

On Solaris, before installation, the following sections enable you to improve the performance of TimesTen on your system.

File system options

In addition to the file system options listed in the section "General UNIX requirements" on page 1-4, on Solaris UFS file systems, if you plan to have TimesTen applications that use DurableCommits=1, mount the file system with the -forcedirectio option.

Create a project (Solaris)

For Solaris 10 and 11 systems, you must create a project to manage system resources, such as shared memory, file descriptors and semaphores.

You can create a group project or a user project.

> **Note:** If you create a users group, the instance administrator must run the newtask command each time the TimesTen daemons must be restarted. If the TimesTen daemons start at system start time, add the newtask command to the system startup scripts.

For example, to create a project `timesten` for the group `g500`, with 500 GB of shared memory, 4096 semaphores and 65,535 file descriptors:

1. Login as user `root`.

2. Add the group project.

   ```
   # projadd -G g500 timesten
   ```

3. Modify the shared memory for the group to 500 GB.

   ```
   # projmod -a -K "project.max-shm-memory=(priv,500GB,deny)" timesten
   ```

4. Modify the maximum number of semaphores to 4096.

   ```
   # projmod -a -K "process.max-sem-nsems=(priv,4096,deny)" timesten
   ```

   > **Note:** For each active database, TimesTen consumes a minimum of 155 `SEMMSL` plus one `SEMMSL` for each connection. See related discussion in "Semaphores (general UNIX)" on page 1-5 and "Semaphores (Linux)" on page 1-10.

5. Modify the maximum number of file descriptors to 65,535.

   ```
   # projmod -a -K "process.max-file-descriptor=(priv,65535,deny)" timesten
   ```

6. Run the newtask command before restarting the TimesTen daemons.

   ```
   # newtask -p timesten -c $$
   ```

Or, for example, to create a user project for the user `timesten`, with 500 GB of shared memory, 4096 semaphores and 65,535 file descriptors:

1. Login as user `root`.

2. Add the user project.

   ```
   # projadd -U timesten user.timesten
   ```

3. Modify the shared memory for the group to 500 GB.

   ```
   # projmod -a -K "project.max-shm-memory=(priv,500GB,deny)" user.timesten
   ```

4. Modify the maximum number of semaphores to 4096.

   ```
   # projmod -a -K "process.max-sem-nsems=(priv,4096,deny)" user.timesten
   ```

   > **Note:** For each active database, TimesTen consumes 155 `SEMMSL`, plus one `SEMMSL` for each connection.

5. Modify the maximum number of file descriptors to 65,535.

   ```
   # projmod -a -K "process.max-file-descriptor=(priv,65535,deny)" user.timesten
   ```

Every user and every group are associated to a default project, which is the project under which their processes are run. The project or process settings used by a user are those that occur first in the `/etc/project` file. If you have not modified the `project` file, the system default project settings occur first.

> **Note:** Do not remove the default project settings from the `project` file. Instead, place your project settings at the top of the `project` file above the default settings.

For either the user project method or group project method, you can choose between the following two options for associating your project settings with the specified user or group.

- Edit the `/etc/project` file to move the timesten project entry so that it precedes the default entry.

- Execute the following before restarting daemons. This is required if the project was created with `-G` only.

  ```
  # newtask -p timesten -c $$
  ```

Windows requirements

This section discusses Windows requirements:

- Large pages: grant lock pages in memory (Windows)

- Additional Windows requirements

Large pages: grant lock pages in memory (Windows)

To use large pages on Windows, you must grant `SeLockMemoryPrivilege`, the privilege to lock pages in memory. On Windows 7, complete the following steps:

1. Go to the **Control Panel** and choose **Administrative Tools**.

   The Administrative Tools window opens.

2. Choose **Local Security Policy**.

   The Local Security Policy window opens (Local Security Settings window on Windows XP).

3. In the left pane of the Local Security Policy window, expand **Local Policies** and select **User Rights Assignment**.

4. In the right pane of the Local Security Policy window, choose: **Lock pages in memory** (on Windows XP, also choose **Action > Security...**).

   The Lock pages in memory Properties window opens (the Local Security Policy dialog on Windows XP).

5. Specify **Add User or Group** (**Add** on Windows XP).

   The Select Users, Computers, Service Accounts, or Groups dialog opens (Select Users or Groups dialog on Windows XP).

6. Enter the instance administrator or applicable user name in the field: **Enter the object names to select**. Then specify **Check Names**. (Select the applicable name from the **Name** list and specify **Add** on Windows XP.)

   If the name you entered is not recognized, resolve this in the resulting Name Not Found dialog.

7. Specify **OK** to close the Select Users, Computers, Service Accounts, or Groups dialog box.

8. Specify **OK** to close the Lock pages in memory Properties window.

Additional Windows requirements

Be aware of the following Windows requirements:

- On more recent Windows versions, such as Vista, Windows 2008, and Windows 7, you must have Administrator privileges to perform certain operations, such as starting and stopping the TimesTen daemon.

  To start a command prompt window with Windows Administrator privileges, right-click the `cmd.exe` executable. (In Windows 7, for example, this executable is located in the `C:\Windows\System32` folder.) When the command window opens, it indicates "Administrator" in the title bar.

- The TimesTen debug libraries depend on Visual Studio 2005, 2008, or 2010. If you intend to use the debug libraries, ensure that one of these versions is installed.

Default installation directories

The TimesTen default installation directories for release 11.2.2 are as follows:

- On Windows, `C:\TimesTen`

- On UNIX, `$HOME/TimesTen`

The temporary directory is operating system-dependent. Usually it is located according to the following.

- On Windows it is according to the `%TMP%` environment variable. This typically points to a location such as the following, for example:

  ```
  C:\Documents and Settings\username\Local Settings\Temp
  ```

  Or the equivalent on Window 7, where `C:\Users` replaces `C:\Documents and Settings`.

- On Solaris and Linux: `/tmp`

- On HP-UX and AIX: `/var/tmp`

You can change the location of your temporary directory by setting the `TMP` environment variable on Windows. On UNIX, you can change the location of your temporary directory by setting the `TMPDIR` environment variable.

> **Note:** TimesTen does not support file path names that contain multibyte characters. Ensure that the installation path, database path, transaction log path, and temporary file path do not contain any multibyte characters. On Windows, the complete temporary directory path must be less than 190 characters for the installation to complete successfully.

Considerations for locations of database files and other user files

During installation, if you have elected to install the TimesTen Quick Start sample programs, the installer prompts you for a location for the `DemoDataStore` directory. On UNIX systems, by default, this is located under the `info` directory that is under the TimesTen installation path. It is strongly recommended that you choose an alternate location, outside of the TimesTen installation path, for this directory. (On Windows systems, the default location is under the user's application data folder.)

It is also strongly advised that you not store any database files (checkpoint and log files) or any other user files anywhere under the TimesTen installation path. Any files

under the installation path, including files not installed by TimesTen, may be removed during upgrade or uninstallation operations.

Pre-Install requirements for operating system group and file permissions

The following sections describe creating the operating system groups and setting the correct directory permissions for TimesTen:

- TimesTen instance administrators and users groups

- Directory and file permissions

- Creating UNIX TimesTen instance administrators and users groups

TimesTen instance administrators and users groups

For security, access to the TimesTen installation is restricted to members of a single operating system group, under which TimesTen is installed. This group is known as the *TimesTen users group*. Only users that are members of the TimesTen users group are allowed to perform direct driver connections to TimesTen and perform operations on TimesTen databases. Any users connecting to a TimesTen database through a client connection are not required to be members of the TimesTen users group.

The user that installs TimesTen is the *instance administrator*. The instance administrator must be a member of the *TimesTen instance administrators group*, and must also be a member of the TimesTen users group.

- On Windows, which does not have the same concept of "instance administrators group" as UNIX, the TimesTen users group is effectively equivalent. Therefore, the instance administrator on a Windows installation must be a member of the TimesTen users group to install TimesTen. In addition, all users who perform a direct driver connection must be a member of the TimesTen users group.

- On UNIX, the TimesTen instance administrators group and the TimesTen users group can be the same or different operating system groups, as follows.

  - TimesTen instance administrators group. Any user installing TimesTen must be a member of this group. This group must be granted read and write access to /etc/TimesTen, which contains information about all TimesTen instances installed on the system.

  - TimesTen users group. The instance administrator must also be a member of this group to install TimesTen. After installation, only members of this operating system group are allowed to make direct driver connections to TimesTen and perform operations on TimesTen databases.

  The details on how to create both operating system groups on UNIX are included in "Creating UNIX TimesTen instance administrators and users groups" on page 1-16.

Directory and file permissions

When installed, read and write permissions on TimesTen files and directories is limited to members of the TimesTen users group, unless TimesTen was installed as "world accessible".

The following sections describe directory and file permissions for Windows and UNIX systems.

Permissions and instance registry prerequisites for TimesTen

On Windows, TimesTen files and directories are accessible only to members of the TimesTen users group.

If you choose to install TimesTen as world accessible, which is an option during the installation, TimesTen files and directories are accessible to everyone. In this case, anyone can perform any action on the TimesTen database files and shared memory segments. This is not recommended. Enable this option only if all users on this computer are trusted and you want to disable all operating system-level access control for this installation.

For more information on operating system groups, see "TimesTen instance administrators and users groups" on page 1-15.

On Windows, information about TimesTen is contained in the operating system registry.

On UNIX, TimesTen maintains a registry of all TimesTen instances installed on a given computer in `/etc/TimesTen`. The instance registry itself is not required for operation, but it is essential for correct installation and uninstallation of TimesTen. Before installing TimesTen, ensure that the user installing TimesTen is a member of the instance administrators group and has read and write permissions on the `/etc/TimesTen` directory.

The details on how to set the directory permissions for `/etc/TimesTen` to the instance administrators group are included in the next section, "Creating UNIX TimesTen instance administrators and users groups".

> **Note:** Checkpoint files and log files for databases should be installed on separate devices. TimesTen writes a message to the support log if the transaction log files and checkpoint files are on the same file system.

Creating UNIX TimesTen instance administrators and users groups

The following details the pre-installation procedures to create the required operating system groups and set the directory permissions for the UNIX TimesTen installation.

- Create the TimesTen users group
- Create the TimesTen instance registry and instance administrators group

Create the TimesTen users group

During installation, you must specify the TimesTen users group. By default, the TimesTen users group for the instance is the primary operating system group of the user installing TimesTen. If you want the TimesTen users group to be other than the installer primary group, you must specify the name of the group during installation.

Alternatively, you can make the TimesTen instance world accessible. However, this is not recommended.

The only way to change the TimesTen users group is to uninstall and reinstall the TimesTen instance, providing the new group name during reinstallation.

If you do not have an operating system group for TimesTen users, the following outlines certain procedures that must be performed once as user `root` before installing TimesTen to create the TimesTen users group.

1. Create an operating system group under which the TimesTen instance can be installed. The name `timesten` is suggested for this operating system group, but you can choose any name that you prefer.

   > **Note:** Throughout this manual, for our examples, `timesten` is the name of the TimesTen users group.

2. Add the user who is installing and any users who are using TimesTen to the TimesTen users group that you just created.

3. Provide the name of this group, if different from the default TimesTen users group, during the installation at the appropriate time.

The directory and file permissions for the TimesTen installation have the group specified as the group you defined during the installation. This sets the permissions to restrict read and write access for most directories, files, checkpoint files, transaction log files, shared memory segments, and semaphores to this defined group. There are exceptions for certain resources as determined by TimesTen. See "Directory and file permissions" on page 1-15 for more information on permissions.

   > **Note:** On Solaris systems, you must be installed as `root` to use `MemoryLock` with a setting of 1 or 2. Databases in a non-`root` instance of TimesTen can use settings 3 and 4 for this attribute on Solaris systems.

Create the TimesTen instance registry and instance administrators group

On UNIX platforms, the instance registry is located in the directory `/etc/TimesTen`. Initial creation of the `/etc/TimesTen` directory may require `root` access. Creation of this directory is a once per computer, pre-installation step.

If the user installing TimesTen does not have read and write access to the `/etc/TimesTen` directory, the following outlines certain procedures that must be performed once as user `root` before installing TimesTen.

1. Create an operating system group for the TimesTen instance administrators group. You can name this group as desired, but examples will use the name `ttadmin`. For example, on Linux or Solaris:

   ```
   # groupadd ttadmin
   ```

2. If the directory `/etc/TimesTen` does not exist, create it.

   ```
   # mkdir /etc/TimesTen
   ```

   The disk space required for the files in this directory is at least 100 KB.

3. Assign ownership and permissions on the `/etc/TimesTen` directory to the TimesTen instance administrators group so that only the instance administrator may access and execute. At installation time, the `instance_info` file is added to the `/etc/TimesTen` directory. This file must be readable and writable by the instance administrators group.

   Before installing TimesTen, set the permission mode for `/etc/TimesTen` to 770, and permissions for all files under `/etc/TimesTen` to 660.

The following commands modify the group ownership of the TimesTen directory to be the `ttadmin` group and changes the permissions for all files in this directory to read and write for members of the `ttadmin` group:

```
# chgrp -R ttadmin /etc/TimesTen
# chmod 770 /etc/TimesTen/
# chmod 660 /etc/TimesTen/*
```

4. You can now install TimesTen on UNIX systems. The installer verifies the existence and permissions of `/etc/TimesTen` and fails if the permissions are not correct.

Installing TimesTen on UNIX systems

The instance may be installed in any directory to which the TimesTen instance administrator has sufficient permission.

> **Note:** Before beginning installation, ensure that the prerequisites defined in "Installation prerequisites" on page 1-4 have been met.

The following sections provide instructions on installing TimesTen on UNIX systems.

- Installing TimesTen
- Informational messages on UNIX systems
- Changing the daemon port number on UNIX
- Uninstalling TimesTen on UNIX systems

Installing TimesTen

To install TimesTen on your UNIX system, use the following steps.

1. Download the appropriate TimesTen ZIP/TAR file for the current release. File names indicate the platform and release number.

2. Log in as the TimesTen instance administrator.

3. Extract the contents of the file.

4. Change to the platform directory:

```
% cd platform
```

For example on a Linux system:

```
% cd linux8664
```

5. Run the TimesTen setup script:

```
% ./setup.sh
```

> **Note:** If a user installs TimesTen as `root`, the installer gives the following warning: "You are about to install TimesTen as `root`. TimesTen daemon processes run with `root` privileges."
>
> If you click `OK` to install as `root`, then the instance administrator is `root`, and any actions or applications that must be performed by the instance administrator must be run as `root`.

While no options are required to install TimesTen, the `setup.sh` script takes these options:

Option	Description
`-install`	Installs TimesTen (enabled by default).
`-uninstall`	Uninstalls TimesTen. Also see "Uninstalling TimesTen on UNIX systems" on page 1-22.
`-batch` *filename*	Allows installation or uninstallation without prompts. The batch file *filename* is optional. If *filename* is specified, the installation reads all installation prompts from the file. If *filename* is not specified or does not contain an instance name, TimesTen installs a default instance, using tt1122_*bits* (_32 or _64) for the instance name. If an instance with the same name exists on the installation computer, the installation procedure fails. It is recommended that you create the batch file and specifically indicate the instance name of the installation.
`-help`	Displays the help message.
`-installDoc`	Installs the TimesTen documentation library and Quick Start tutorial pages. (You can do a documentation-only installation.)
`-quickstart`	Installs the TimesTen Quick Start sample programs (to an existing installation).
`-record` *filename*	Records responses to the installation questions into the file specified by *filename*. The file can then be used as the parameter to the `-batch` option.
`-verbose`	Displays extra installation information.

6. Enter your response to the setup script prompts.

> **Note:** To install or uninstall TimesTen without having to respond to prompts, use the `-batch` flag with the `setup.sh` script.
>
> It is recommended that you re-create the response file using `-record` each time changes are made. Because answers to new installation questions may not be present in the original silent installation response file, unexpected results can occur.

The setup script performs the following actions (unless your answers resulted in termination of the installation process).

- Prompts you to choose one of the following:
  - Install a new instance.

- – Upgrade an existing instance (*in-place upgrade*). You can use this option to upgrade to a new patch release within the same major release, such as from 11.2.2.4.0 to 11.2.2.5.0. (See "In-place upgrades" on page 3-7.)

  Important: Before upgrading, ensure that you do not have any databases or important files under the TimesTen installation directory. See "Considerations for locations of database files and other user files" on page 1-14 for related information.

  - – Display information about an existing instance.

  - – Quit the installation.

- ■ Prompts you to choose the default instance name or choose an instance name for your TimesTen instance.

  > **Note:**   Each TimesTen installation is identified by a unique instance name. The instance name must be a non-null alphanumeric string of 255 characters or less.

- ■ Prompts you to install one of the following components:

  - – Client/Server and Data Manager

  - – Data Manager only

  - – Client only

- ■ Prompts you for the location of your TimesTen instance. By default, installs the instance in a subdirectory under `$HOME/TimesTen`. The subdirectory is named according to the TimesTen instance name. TimesTen documentation refers to this installation directory as *install_dir*.

- ■ Prompts you for the location of the TimesTen daemon home directory. The default location is *install_dir*/info.

  The daemon home directory is the working directory for the TimesTen daemon and related processes. Among other items, it contains files written and read by the daemon, including the daemon options file (`ttendaemon.options`), process ID file (`timestend.pid`), user and support log files, and DBI files. It also contains TimesTen `.ini` files such as `sys.odbc.ini`, `sys.ttconnect.ini`, and `cluster.oracle.ini`.

- ■ Prompts you for the location of TimesTen daemon log files. In this release, separate support and error logs replace the daemon log. The default location is *install_dir*/info.

- ■ Prompts you to specify the daemon port number. (The default port number, 53396 for 64-bit systems and 53392 for 32-bit systems, is shown unless already in use.)

> **Notes:**
>
> - When you install multiple instances of TimesTen on the same computer, specify a unique TCP/IP port number for each TimesTen daemon and each TimesTen server during the installation.
>
> - If you will be using replication, however, be aware that if the daemon ports are different for TimesTen databases that replicate to each other, you must use static ports for replication and you must use the `ttRepAdmin` utility `-remoteDaemonPort` option for duplicate operations. You can use the `ttVersion` utility to verify the port number of your installation of TimesTen.

- Prompts you to set the TimesTen users group or choose world accessibility. For more information on these options, see "Pre-Install requirements for operating system group and file permissions" on page 1-15 for details on the TimesTen users group and file permissions. You can do the following:

  1. Restrict access to group *default group*.

  2. Restrict access to a different group.

  3. Make the TimesTen instance world accessible (not recommended). Choose this option only if all users on this computer are trusted and you want to disable all operating system-level access control for this installation.

- Prompts you to confirm whether PL/SQL should be installed for the instance. The default answer is "yes". If you choose not to install PL/SQL, you can use the TimesTen `ttmodinstall` utility to install it later. For more information, see "ttmodinstall" in *Oracle TimesTen In-Memory Database Reference*.

  > **Note:** Installing PL/SQL is encouraged, although it increases the size of some TimesTen libraries. Some features depend on it.

- Prompts you to set the `TNS_ADMIN` environment variable. The value is a directory location where the `tnsnames.ora` file can be found.

  You can skip this. If you do not specify the location at installation time, you can set it later with the `ttmodinstall` utility. However, you must specify it before using the TimesTen Application-Tier Database Cache (TimesTen Cache).

- Prompts you to specify the server port number. (The default port number, 53397 for 64-bit systems and 53393 for 32-bit systems, is shown unless already in use.)

- Prompts you to install Quick Start sample programs and the TimesTen documentation. The TimesTen Quick Start programs can take up to 64 MB of disk space. The default directory is *install_dir*/quickstart and *install_dir*/doc.

- Prompts for a location for the demo databases. When you install the Quick Start sample programs, this is where the TimesTen demo database files are installed. The default location is the *install_dir*/info/DemoDataStore directory. See the section "Considerations for locations of database files and other user files" on page 1-14 for recommendations on choosing an alternative location.

- Prompts you to indicate whether you want to install TimesTen replication with Oracle Clusterware. Prompts you for the path into which to install the Oracle

Clusterware installation on this computer and the port number for the TimesTen Clusterware agent.

Be aware that TimesTen 11.2.2 supports Clusterware releases 11.2.0.2 and 11.2.0.3 only. (TimesTen 11.2.1 supports Clusterware releases 11.1.0.7, 11.2.0.2, and 11.2.0.3.)

- The installer checks for any nodes where Oracle Clusterware is currently configured and prompts you to specify a node list for TimesTen replication with Oracle Clusterware. (Oracle Clusterware can be installed or configured after your TimesTen installation, using the `ttmodinstall` utility.)

- Removes any previous installation of this release of TimesTen if you are installing an upgrade.

- Installs the TimesTen components into the appropriate directories.

- Starts the daemon. (See "Starting and stopping the TimesTen daemon" in *Oracle TimesTen In-Memory Database Operations Guide* for related information.)

Informational messages on UNIX systems

As the TimesTen daemon operates, it generates error, warning, informational and debug messages for TimesTen system administration and for debugging applications. At installation time, you determine whether these messages go into a file, which is typical, or to the `syslog` facility. Using the `syslog` facility is discouraged, but you can refer to your operating system documentation if you are interested in doing so.

If messages are logged using `syslog`, the `LOG_USER` syslog facility is used by default.

To specify the `syslog` facility used to log TimesTen Daemon and subdaemon messages, on a separate line of the `ttendaemon.options` file add:

```
-facility name
```

Possible name values are: `auth`, `cron`, `daemon`, `local0-local7`, `lpr`, `mail`, `news`, `user`, or `uucp`.

Changing the daemon port number on UNIX

Though the instance registry enforces TCP/IP port uniqueness for TimesTen instances, the possibility of the TimesTen main daemon port conflicting with ports used by non-TimesTen applications always exists.

The `ttmodinstall` utility allows the instance administrator to change the port number on which the main TimesTen daemon listens. If you have not stopped the TimesTen daemon before using `ttmodinstall`, the utility stops the daemon before changing the port number. After the port change, the daemon is automatically restarted. This feature is useful if you install TimesTen and later find that the port is in use.

The utility is run from the command line and takes the `-port` option with the new port number as an argument. For example:

```
% ttmodinstall -port 12345
```

See *Oracle TimesTen In-Memory Database Reference* for more details on `ttmodinstall`.

Uninstalling TimesTen on UNIX systems

To uninstall all TimesTen components, follow these steps:

1. Log in as the TimesTen instance administrator.

2. The TimesTen setup script is in the *install_dir*/bin directory. Run the script with the -uninstall option in a directory outside of the installation directory, by typing:

```
% install_dir/bin/setup.sh -uninstall
```

Uninstalling removes all TimesTen libraries and executables and also stops and uninstalls the daemon and Server. You can execute ps to verify that all TimesTen processes have terminated. To verify that TimesTen has been successfully uninstalled, verify that the *install_dir* no longer exists.

> **Important:** Before uninstalling, ensure that you do not have any databases or important files under the TimesTen installation directory. See "Considerations for locations of database files and other user files" on page 1-14 for related information.

Installing TimesTen on Windows systems

This section discusses installation and related issues for Windows systems. Before beginning installation, ensure that the conditions defined in "Installation prerequisites" on page 1-4 have been met.

Be aware of the following operational notes:

- On a Windows system, you cannot have multiple instances of the same TimesTen major release. For example, you cannot have both an 11.2.2.4.0 and an 11.2.2.5.0 installation. But you can have TimesTen instances from different major releases, such as any 11.2.2.*x.x* installation along with any 11.2.1.*x.x* installation.

- TimesTen provides separate installers for 32-bit and 64-bit installations. On Windows 64-bit systems, you can do either a 64-bit installation (typical) or a 32-bit installation. However, similarly to the previous restriction, you cannot do both a 64-bit installation and a 32-bit installation of the same release. You can only have a 32-bit installation and a 64-bit installation simultaneously if they are from different major releases, such as a 64-bit installation of an 11.2.2.*x.x* release and a 32-bit installation of an 11.2.1.*x.x* release.

- The TimesTen 64-bit installation provides both a 64-bit client and a 32-bit client. It is typical to use the 64-bit client, but you can install either or both.

The following sections provide instructions on installing TimesTen on Windows systems.

- Installing TimesTen
- Installing TimesTen in silent mode
- Verifying installation
- Verifying TimesTen Client and Server installation
- Informational messages on Windows systems
- Uninstalling TimesTen on Windows systems

Installing TimesTen

To install TimesTen on your Windows system, follow these steps:

1. Download the appropriate TimesTen ZIP file for the current release. File names indicate the platform and release number.

2. Extract the contents of the file.

3. Run the `setup.exe` executable, which is located in the `Win64` or `Win32` directory from the ZIP file.

Notes:

■ In Windows 7, you must right-click `setup.exe` and select "Run as administrator" from the resulting dropdown list. (In earlier Windows versions, you have the necessary administrative privileges by default when you execute `setup.exe`.)

■ Each time you execute `setup.exe`, the installer checks for previous TimesTen installations. On Windows, you can install only one instance of any major TimesTen release, where a major release is indicated by the first three parts of the release number, such as 11.2.2. If there is already an installation from the same TimesTen major release line (such as an 11.2.2.4.0 installation if you are trying to install 11.2.2.5.0), the installer returns an error message asking you to uninstall the previous release.

■ TimesTen cannot be installed in a mapped network drive. Attempting to do so results in an error.

The TimesTen installer performs the following actions.

■ Prompts you for the location of the installation. By default, TimesTen is installed in a subdirectory under `C:\TimesTen`. The subdirectory is named according to the TimesTen instance name, such as `C:\TimesTen\tt1122_32` (32-bit installation) or `tt1122_64` (64-bit installation). TimesTen documentation refers to this installation directory as *install_dir*.

■ Prompts you to select the components that you would like to install:

  – TimesTen Data Manager

  – TimesTen Data Manager Debug Libraries

  – TimesTen Server

  – TimesTen Client

    In the 64-bit installation, there are options for both a 64-bit client and a 32-bit client. You can choose both (default), either, or neither. If you install the 32-bit client, it is installed in the `tt1122_64\client32` directory.

  – Optional Components

    * TimesTen Quick Start sample programs

    * TimesTen Documentation (including Quick Start tutorial pages)

  For more information, see "Components available on Windows" on page 1-3.

■ Prompts for the location to install the demo databases. When you install the TimesTen Quick Start sample programs, the demo database files are installed in the `DemoDataStore` directory that defaults to the following location:

  `%APPDATA%\TimesTen\DemoDataStore\`

  This is a location such as the following, for example:

  `C:\Documents and Settings\username\Application Data\TimesTen\DemoDataStore\`

Or the equivalent on Window 7, where `C:\Users` replaces `C:\Documents and Settings`. For example:

`C:\Users\`*username*`\Application Data\TimesTen\DemoDataStore\`

Note that `Application Data` may be a symbolic link (to `AppData\Roaming` on Windows 7, for example). See the section "Considerations for locations of database files and other user files" on page 1-14 for suggestions on selecting a location for the demo database files.

- Prompts you to set the `TNS_ADMIN` environment variable. The value is a directory location where the `tnsnames.ora` file can be found.

  You can leave this field blank. If you do not specify the location at installation time, you can set it later with the `ttmodinstall` utility. However, you must specify it before using the TimesTen Application-Tier Database Cache (TimesTen Cache).

- Prompts you to select the Program Folder. Browse to choose the folder that you want for this installation either from existing folders or a new folder. The default is `TimesTen 11.2.2 (32-bit)` or `TimesTen 11.2.2 (64-bit)`.

- Asks if you want permissions on this installation to be readable and writable by anyone who has access to the computer. This is not recommended. If disabled, permissions are restricted to users who are members of the TimesTen users group. See "Pre-Install requirements for operating system group and file permissions" on page 1-15 for details on permissions and world accessibility. Choose this option only if you want to disable all operating system-level access control for this installation.

- Prompts you to confirm whether PL/SQL should be installed for the instance. The default answer is "yes". If you choose not to install PL/SQL, you can use the TimesTen `ttmodinstall` utility to install it later. For more information, see "ttmodinstall" in *Oracle TimesTen In-Memory Database Reference*.

  > **Note:** Installing PL/SQL is encouraged, although it increases the size of some TimesTen libraries. Some features and functionality depend on it.

- Prompts you to register environment variables. If selected, the installation program adds TimesTen directories to the system environment variables `PATH`, `LIB`, and `INCLUDE` and sets other appropriate variables. If you decide not to register the environment variables at installation time, you can set the environment variables at any time after installation on a per session basis by running the batch file *install_dir*`\bin\ttenv.bat`. See "Setting environment variables for TimesTen" on page 1-35 for information about `ttenv`.

  > **Note:** This is not advisable if you are installing multiple releases of TimesTen, in which case the path could cause unexpected behavior. (Multiple releases must be from different major release lines. Multiple patch releases within the same major release are not allowed on Windows.)

- Prompts you to select the JDK version, if any, to add to the `CLASSPATH` variable.

- Displays your installation selections before continuing to install TimesTen.

- Prompts you to display the release notes and launch the Quick Start. For information on the Quick Start, see "TimesTen Quick Start" on page 1-39.

> **Note:** The Windows installer does not include an option for the location of the daemon home directory. Its location is *install_dir*\srv\info. This is the working directory for the TimesTen daemon and related processes. Among other items, it contains files written and read by the daemon, including the daemon options file (ttendaemon.options), process ID file (timestend.pid), user and support log files, and DBI files. It also contains TimesTen .ini files such as sys.odbc.ini, sys.ttconnect.ini, and cluster.oracle.ini.

Installing TimesTen in silent mode

TimesTen enables you to save installation options to a batch file, referred to as a response file, that you can later use to install TimesTen without having to answer each option in a dialog box.

- To create a response file in the default location, use the following command:

```
C:> setup.exe /r
```

  With this command, TimesTen walks you through a normal setup operation and, by default, saves your responses to the file C:\Windows\setup.iss.

- To later do a silent installation using the options saved to the default response file, use the following command:

```
C:> setup.exe /s
```

  This acquires the installation options from the response file. No dialog boxes appear. (Some information pop-up dialogs may still appear, however, such as the one that informs you that the services are being started.)

Alternatively, you can use the f1 option to specify the name and location of the response file.

- To create the response file mysetup.iss in C:\Temp, use the following command:

```
C:> setup.exe /r /f1"C:\Temp\mysetup.iss"
```

- To use this response file for a silent installation, use the following command.

```
C:> setup.exe /s /f1"C:\Temp\mysetup.iss"
```

> **Note:** Batch files from releases older than TimesTen Release 11.2.2 should not be used to install this release. All new prompts in the installer for this release are assigned default answers and may produce unexpected results when batch files from different releases are used.

Verifying installation

To verify that TimesTen has been properly installed, check that the driver files are available and that the services are running:

1. Confirm that the TimesTen 11.2.2 shortcut—for example, "TimesTen 11.2.2 (64-bit)"—appears under **All Programs**.

2. From **Administrative Tools**, select **Data Sources (ODBC)**. This opens the ODBC Data Source Administrator.

3. Choose the **Drivers** tab. Check to see that the **TimesTen Data Manager 11.2.2** driver is installed. (If you installed TimesTen Client, that driver should be listed as well. See the next section, "Verifying TimesTen Client and Server installation".) Click **OK**.

4. From **Administrative Tools**, select **Services** and check that the **TimesTen Data Manager 11.2.2** service has status "Started". (See "Starting and stopping the TimesTen daemon" in *Oracle TimesTen In-Memory Database Operations Guide* for related information.)

Verifying TimesTen Client and Server installation

Perform the following steps to verify that the Client and Server have been properly installed.

> **Note:** The instructions in this section are valid if you are installing 32-bit TimesTen on 32-bit Windows or 64-bit TimesTen on 64-bit Windows. However, if you are installing 32-bit TimesTen on 64-bit Windows, verify the TimesTen ODBC entries by executing the following, which is the 32-bit version of the ODBC Data Source Administrator:
>
> ```
> %WINDIR%\SysWOW64\odbcad32.exe
> ```

1. From **Administrative Tools**, select **Data Sources (ODBC)**. This opens the ODBC Data Source Administrator.

2. Choose the **Drivers** tab. Check to see that the **TimesTen Client 11.2.2** driver is installed. (You should also see the **TimesTen Data Manager 11.2.2** driver. See the preceding section, "Verifying installation".) Click **OK**.

   Note: If you install the 32-bit client in a 64-bit installation, you must use the 32-bit ODBC Data Source Administrator, mentioned in the note above, to view its driver entry.

3. Choose the **System DSN** tab.

4. Select the `sampledbCS_1122` sample database and click **Configure**.

> **Notes:**
>
> - This step assumes you installed TimesTen Quick Start sample programs, including the sample databases.
> - The `sampledb_1122` DSN is used for direct connections. The `sampledbCS_1122` DSN is used for client/server connections.

This opens the TimesTen Client Data Source Setup dialog.

5. Click **Test Oracle TimesTen Server Connection** to attempt a connection to the server.

   The ODBC Administrator attempts to connect to the TimesTen Server and displays a message to let you know if it was successful. When you click this button, the TimesTen Client verifies that the following are true.

- ODBC, Windows sockets, and the TimesTen Client are installed on the system.

- The TimesTen Server you have selected is defined.

- The TimesTen Server is running.

6. Click **Test Data Source Connection** to attempt a connection to the data source on the TimesTen Server.

    The ODBC Data Source Administrator attempts to connect to the TimesTen data source and displays a dialog to let you know if it was successful. When you click **Test Data Source Connection**, the TimesTen Client verifies that:

- The data source you have chosen is defined on the server.

- The TimesTen Client can connect to the data source.

Informational messages on Windows systems

TimesTen writes error messages into the `tterrors.log` file. This file is located in the *install_dir*\srv\info directory. You can use the `ttDaemonLog` utility to view messages logged by the TimesTen Data Manager. For a description of the system administration utilities, see "Utilities" in *Oracle TimesTen In-Memory Database Reference*.

Uninstalling TimesTen on Windows systems

You can uninstall TimesTen on Windows as you would any other program.

> **Important:** Before uninstalling, ensure that you do not have any databases or important files under the TimesTen installation directory. See "Considerations for locations of database files and other user files" on page 1-14 for related information.

1. Go to **Add/Remove Programs** (or **Programs and Features** in Windows 7).

2. In the list of programs, select TimesTen. For example, select "Oracle TimesTen In-Memory Database 11.2.2 (64-bit)".

3. Click **Remove** (or **Uninstall** in Windows 7).

    You will be asked whether you want to remove DSN definitions associated with the installation. If you answer Yes, then all definitions are removed. This includes user-defined DSNs as well as those created by the TimesTen installation process.

    Important: If you are uninstalling as part of the equivalent of an in-place upgrade (discussed in "Performing an in-place upgrade" on page 3-9), you must keep your DSNs.

To verify that removal was successful, check the following:

- TimesTen 11.2.2 has been removed from **All Programs**.

- TimesTen Data Manager 11.2.2 has been removed from the Services list.

- The TimesTen 11.2.2 drivers have been removed from the **Drivers** tab in the ODBC Data Source Administrator.

See "Verifying installation" on page 1-26 for information about the Services list and ODBC Data Source Administrator.

Installing TimesTen Client on Mac OS X systems

The following sections provide instructions on installing the TimesTen Client on Mac OS X systems.

- Installing TimesTen Client
- Uninstalling TimesTen Client on Mac OS X systems

Installing TimesTen Client

To install the TimesTen Client on your Mac OS X system, use the following steps.

1. Download the appropriate TimesTen Client ZIP/TAR file for the current release. File names indicate the platform and release number.

2. Log in as the TimesTen instance administrator.

3. Extract the contents of the file.

4. Change to the platform directory:

   ```
   % cd macos64
   ```

5. Run the TimesTen setup script:

   ```
   % ./setup.sh
   ```

   While no options are required to install the TimesTen Client, the setup.sh script takes these options:

Option	Description
-install	Installs TimesTen Client (enabled by default).
-uninstall	Uninstalls TimesTen Client.
	Also see "Uninstalling TimesTen Client on Mac OS X systems" on page 1-31.
-batch *filename*	Allows installation or uninstallation without prompts. The batch file *filename* is optional.
	If *filename* is specified, the installation reads all installation prompts from the file.
	If *filename* is not specified or does not contain an instance name, TimesTen installs a default instance, using tt1122 for the instance name. If an instance with the same name exists on the installation computer, the installation procedure fails.
	It is recommended that you create the batch file and specifically indicate the instance name of the installation.
-record *filename*	Records responses to the installation questions into the file specified by *filename*. The file can then be used as the parameter to the -batch option.
-installDoc	Installs the TimesTen documentation library and Quick Start tutorial pages. (You can do a documentation-only installation.)
-quickstart	Installs the TimesTen Quick Start sample programs (to an existing installation).
-help	Displays the help message.
-verbose	Displays extra installation information.

6. Enter your response to the setup script prompts.

> **Note:** To install or uninstall TimesTen Client without having to respond to prompts, use the -batch flag with the setup.sh script.
>
> It is recommended that you re-create the response file using -record each time changes are made. Because answers to new installation questions may not be present in the original silent installation response file, unexpected results can occur.

The setup script performs the following actions (unless your answers resulted in termination of the installation process).

- Prompts you to choose the default instance name or choose an instance name for your TimesTen Client instance.

> **Note:** Each TimesTen Client installation is identified by a unique instance name. The instance name must be a non-null alphanumeric string of 255 characters or less.

- Prompts you to choose the location for your TimesTen Client instance. Choose one of the following:

```
[1] $HOME
[2] $PWD
[3] Specify a location
[q] Quit the installation
```

  TimesTen installs the instance in a subdirectory under *specified_dir*/TimesTen. The subdirectory is named according to the TimesTen Client instance name. TimesTen documentation refers to this installation directory as *install_dir*.

- Prompts you to set the TimesTen users group or choose world accessibility. For more information on these options, see "Pre-Install requirements for operating system group and file permissions" on page 1-15 for details on the TimesTen users groups and file permissions. You can do the following:

  – Restrict access to group *default_group*.

  – Restrict access to a different group.

  – Make the TimesTen instance world accessible (not recommended). Choose this option only if all users on this computer are trusted and you want to disable all operating system-level access control for this installation.

- Prompts you to set TNS_ADMIN environment variable. The value is a directory location where the tnsnames.ora file can be found.

  You can skip this. If you do not specify the location at installation time, you can set it later with the ttmodinstall utility. However, you must specify it before using the TimesTen Application-Tier Database Cache.

- Prompts you to install the Quick Start Sample Programs and the TimesTen Documentation. The default directories for the Quick Start Sample Programs and the TimesTen Documentation are *install_dir*/quickstart and *install_dir*/doc, respectively.

  The Quick Start Sample Programs can take up to 64 MB of disk space. Alternatively, you can choose to install the TimesTen Documentation only.

- Prompts you for the name of the host running the TimesTen server. The value is the logical name, domain name, or IP address of the computer where the TimesTen server is running. This value is set in all of the following locations:

    – The `TTC_Server` attribute value for all sample databases definitions found in the `odbc.ini` file

    – The logical server name in the `ttconnect.ini` file

    – The `Network_Address` attribute value for the aforementioned logical server name in the `ttconnect.ini` file

    For more information, see "Defining a logical server name" in the *Oracle TimesTen In-Memory Database Operations Guide*.

- Prompts you for the TCP/IP port number of the TimesTen server. (The default port number is shown.) This value is set as the `TCP_Port` attribute for the aforementioned logical server name in the `ttconnect.ini` file.

    For more information, see "TCP_Port" in the *Oracle TimesTen In-Memory Database Reference*.

- Installs the TimesTen Client components in the appropriate directories.

Uninstalling TimesTen Client on Mac OS X systems

To uninstall all TimesTen Client components, follow these steps:

1. Log in as the TimesTen Client instance administrator.

2. The TimesTen Client setup script is in the *install_dir*/bin directory. Run the script with the -uninstall option in a directory outside of the installation directory, by typing:

```
% ./install_dir/bin/setup.sh -uninstall
```

Uninstalling removes all TimesTen libraries and executables. To verify that the TimesTen Client has been successfully uninstalled, verify that the *install_dir* no longer exists.

Installing TimesTen on Exalogic systems

To install TimesTen on an Exalogic system, you follow the prompts in the TimesTen installer as described in the procedures in "Installing TimesTen on UNIX systems" on page 1-18. Also, ensure that you have met the prerequisites for Oracle Linux installations. See "Linux prerequisites" on page 1-7

It is recommended that you install TimesTen on each of the compute nodes of an Exalogic system.

> **Note:** Installing TimesTen binaries on the shared disk for use on multiple compute nodes is not supported.

Some considerations on the location of the TimesTen installation files are:

- If the TimesTen checkpoint and transaction log files must reside on the ZFS shared storage, you must add the -allowNetworkFiles option to the `ttendaemon.options` file. See "Allowing database access over NFS-mounted systems" in *Oracle TimesTen In-Memory Database Operations Guide*.

- The database and log directory must reside outside the TimesTen Instance home directory to enable simple rolling upgrades.

It is strongly recommended that the TimesTen installation, checkpoint and transaction log files for each node be stored in the locations shown in Table 1–1.

Table 1–1   Installation, checkpoint file, and transaction log file locations

Component	Directory
Installation	`/non-NFS_mounted_file_system_path/TimesTen/TTinstance`
Database	`/u01/machinename/general/path` or `/u01/common/images/machinename/TTinstance`
Log directory	`/u01/machinename/general/username/TTinstance`

Installing Oracle Clusterware for use with TimesTen

Install Oracle Clusterware according to instructions in Oracle Database documentation:

1. Go go http://www.oracle.com/pls/db112/homepage.

2. Select the "Installing and Upgrading" link.

3. Refer to the *Grid Infrastructure Installation Guide* for your platform.

Also see "Overview of Installing Oracle Clusterware" in *Oracle Clusterware Administration and Deployment Guide*.

Be aware that TimesTen 11.2.2 supports Clusterware releases 11.2.0.2 and 11.2.0.3 only. (TimesTen 11.2.1 supports Clusterware releases 11.1.0.7, 11.2.0.2, and 11.2.0.3.)

Follow these recommendations:

- Create an NFS share on the ZFS 7320 Storage Appliance to be used for both OCR and the voting disk. Specific NFS parameters are required so that the NFS share can be used as a voting disk, refer to "Managing Oracle Cluster Registry and Voting Disks" in *Oracle Clusterware Administration and Deployment Guide* for more details.

- Install Clusterware on the ZFS 7320 Storage Appliance. The solid state disk (SSD) file system may not have sufficient space.

  > **Note:** Installing in a shared Oracle home is not recommended as it does not allow rolling upgrades of Oracle Clusterware. Instead, each compute node should have its own installation of Clusterware.

- Install Clusterware on each compute node. The path to the Clusterware installation must be the same on each compute node. Therefore, it is required to set up a directory structure that allows each compute node to use the same path to access its own Clusterware installation.

  - On the shared storage, a separate directory exists for each compute node:

    * For compute node 1, the directory is:

      `/export/compute_node_1/general`

    * For compute node 2, the directory is:

      `/export/compute_node_2/general`

    *   For compute node *n*, the directory is:

```
/export/compute_node_n/general
```

–   Use NFS mount to map the node specific directory to the same path on each compute node:

    *   On host 1, do as follows.

```
mkdir -p /opt/oracle
mount storage-server:/export/compute_node_1/general /opt/oracle
```

    *   On host 2, do as follows.

```
mkdir -p /opt/oracle
mount storage-server:/export/compute_node_2/general /opt/oracle
```

    *   On host *n*, do as follows.

```
mkdir -p /opt/oracle
mount storage-server:/export/compute_node_n/general /opt/oracle
```

    *   On each host, install Oracle Clusterware in `/opt/oracle/crs`.

ODBC installation

WINDOWS

The ODBC SDK redistributable components are installed in `C:\Windows\System32` on Windows systems. Microsoft only permits TimesTen to redistribute portions of the ODBC SDK; those portions are installed automatically (if they are not present). Other components—Microsoft sample programs, online help files, and C language header files—are available separately from Microsoft as part of the Microsoft ODBC SDK, which can be installed separately as required. Additionally, the ODBC C language header files and ODBC online help are bundled as part of Microsoft Visual Studio .NET 2003 or Microsoft Visual Studio 2005, 2008, or 2010. Most TimesTen developers do not need to install the SDK separately.

UNIX

On UNIX systems, no separate SDK installation is required.

Environment variables

This section describes various environment variables that may have to be set, depending on the features of TimesTen that your application uses. The following table summarizes, in alphabetical order, the environment variables detailed in this section and other parts of this guide. Some of these environment variables are platform-specific.

Environment variable	Settings	For additional information, see:
`LIB`, `LIBPATH`, `LD_LIBRARY_PATH`, `LD_LIBRARY_PATH_64`, `DYLD_LIBRARY_PATH`	Include the `lib` directory under the TimesTen installation directory.	"Shared library path environment variable" on page 1-37

Environment variable	Settings	For additional information, see:
NLS_LANG	Set this to specify a character set for the connection. The setting in the sys.odbc.ini or user odbc.ini file is used by default if not overridden by NLS_LANG. Setting the character set explicitly is recommended. The default is typically AMERICAN_AMERICA.US7ASCII. Notes: ■ TimesTen uses NLS_LANG for the character set only, ignoring what precedes the dot ("."). You can specify just the character set as follows (with the dot still required): NLS_LANG=.WE8ISO8859P1 ■ On Windows, the NLS_LANG setting is searched for in the registry, HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\NLS_LANG, if it is not in the environment. If your program has trouble connecting to TimesTen, confirm the NLS_LANG setting is valid and indicates a character set supported by TimesTen. TimesTen uses the Oracle Instant Client to make these connections.	"Character sets" in *Oracle TimesTen In-Memory Database C Developer's Guide*. This discussion is oriented toward OCI but includes more general information and a link to Oracle Database documentation for further discussion. "Supported character sets" in *Oracle TimesTen In-Memory Database Reference*.
ODBCINI	Include the location where the odbc.ini file used by TimesTen databases is to be found.	"ODBCINI environment variable" on page 1-36
PATH	Include the bin directory under the TimesTen installation directory. On Windows, also include the path to the Oracle Database installation if you are using the TimesTen Cache option.	"PATH environment variable" on page 1-35, "Shared library path environment variable" on page 1-37 and "Installing TimesTen on Windows systems" on page 1-23
SYSODBCINI	Set to the location where the sys.odbc.ini file used by TimesTen system databases is to be found. The default location is usually sufficient, so this can usually be left unset.	"SYSODBCINI environment variable" on page 1-36
SYSTTCONNECTINI	Set to the location for the sys.ttconnect.ini file used by TimesTen Client applications to define logical server names. The default location is usually sufficient, so this can usually be left unset.	"SYSTTCONNECTINI environment variable" on page 1-36
TMP or TMPDIR	Set to the location of the temporary directory. TimesTen uses this directory during recovery and other operations.	"Default installation directories" on page 1-14
TNS_ADMIN	If using TimesTen Cache, you must set this to the location of the tnsnames.ora file.	"TNS_ADMIN environment variable" on page 1-36
Java	For Java applications, there are certain environment variables that must be set.	"Java environment variables" on page 1-37

The following sections describe environment variables in TimesTen and how to set them.

- Setting environment variables for TimesTen
- Environment variable descriptions

Setting environment variables for TimesTen

If, after installation, you want to set the environment variables to standard TimesTen settings, source the `ttenv` script on UNIX or run the `ttenv.bat` batch file on Windows. This includes setting the path and classpath (and library path on UNIX systems) so that TimesTen utilities can be executed, among other things. You must invoke `ttenv` before starting TimesTen in order for any of the changes to take effect.

For UNIX platforms, use the appropriate script in the *install_dir*/bin directory, either `ttenv.sh` or `ttenv.csh`, depending on your shell. Following are examples, assuming your local directory is the `bin` directory (otherwise specify the full path).

For a Bourne-type shell, such as `sh`, `bash`, `zsh`, or `ksh`:

```
./ttenv.sh
```

For a `csh` or `tcsh` shell:

```
source ttenv.csh
```

For a Windows platform, execute the *install_dir*\bin\ttenv.bat batch file.

> **Note:** On Windows, the path, classpath, and library path are already set during installation if "Register environment variables" is enabled, which is the case by default.

Environment variable descriptions

Key environment settings are accomplished through `ttenv` as noted in the preceding section. In case further information is needed, the following sections provide more details on environment variables and indicates which are set by `ttenv`.

The following environment variables are described:

- PATH environment variable
- ODBCINI environment variable
- SYSODBCINI environment variable
- SYSTTCONNECTINI environment variable
- TNS_ADMIN environment variable
- Shared library path environment variable
- Java environment variables

PATH environment variable

TimesTen provides utilities for managing and debugging TimesTen applications. To make these utilities available, include *install_dir*/bin in the PATH setting.

> **Note:** This variable is set by `ttenv` or, on Windows, during installation if environment variables are registered (default).

ODBCINI environment variable

TimesTen applications use the `odbc.ini` file to define data sources and their connection attributes. (For a description of connection attributes, see "Connection Attributes" in *Oracle TimesTen In-Memory Database Reference*.) By default on UNIX platforms, TimesTen first looks for the user `odbc.ini` file in the home directory of the user running the TimesTen application. To override the name and location of this file at run-time, set the `ODBCINI` environment variable to indicate a desired path and file name before launching the TimesTen application.

If TimesTen cannot locate a user DSN file, TimesTen looks for the `sys.odbc.ini` file in *install_dir*/info. (Also see the next section about the `SYSODBCINI` environment variable.) For more information, see "Overview of user and system DSNs" in *Oracle TimesTen In-Memory Database Operations Guide*.

SYSODBCINI environment variable

TimesTen applications use the `sys.odbc.ini` file to define system data sources and their connection attributes. Any user can use a system data source. On Windows, system DSNs are defined from the **System DSN** tab of the ODBC Data Source Administrator. On UNIX, system DSNs are defined in the file *install_dir*/info/sys.odbc.ini. To override the name and location of this file at run-time, set the `SYSODBCINI` environment variable to the path name of a `sys.odbc.ini` file before launching the TimesTen application.

TimesTen first looks for a DSN in the user `odbc.ini` file. (Also see the previous section about the `ODBCINI` environment variable.) If it is not found there, TimesTen looks in the `sys.odbc.ini` file.

SYSTTCONNECTINI environment variable

TimesTen client applications use the `sys.ttconnect.ini` file to define logical server names. For a description of logical server names, see "Working with the TimesTen Client and Server" in *Oracle TimesTen In-Memory Database Operations Guide*. By default on UNIX platforms, TimesTen looks in *install_dir*/info/sys.ttconnect.ini. To override the name and location of this file at run-time, set the `SYSTTCONNECTINI` environment variable before launching the TimesTen Client application.

On Windows systems, you can configure logical server names using the **ODBC Data Source Administrator**.

TNS_ADMIN environment variable

On platforms where TimesTen Cache is supported, to work with Oracle Database data you must set the `TNS_ADMIN` environment variable to indicate the directory of the `tnsnames.ora` file.

> **Note:** On UNIX systems, this variable is set if you source `ttenv`. On Windows, you can optionally set it during installation when prompted.

The `ttmodinstall` utility with the `-tns_admin` option enables you to set a value for this environment variable after installation. See *Oracle TimesTen In-Memory Database Reference* for more details on `ttmodinstall`.

Shared library path environment variable

On Linux systems, add *install_dir*/lib to the LD_LIBRARY_PATH environment variable.

On Solaris systems, add *install_dir*/lib to the LD_LIBRARY_PATH or LD_LIBRARY_PATH_64 environment variable, as appropriate.

On AIX systems, add *install_dir*/lib to the LIBPATH environment variable.

On Windows systems, add *install_dir*\lib to the LIB environment variable.

On Mac OS X systems, add *install_dir*/lib:*install_dir*/ttoracle_home/instantclient_11_2 to the DYLD_LIBRARY_PATH environment variable.

> **Note:** On UNIX systems, the applicable variable is set if you source ttenv. On Windows, LIB is set during installation if environment variables are registered (default).

Java environment variables

The following sections provide more detail about the environment variables that affect the environment for TimesTen Java applications.

- Set the CLASSPATH variable
- Set the shared library path variable
- Set the THREADS_FLAG variable (UNIX only)
- Set the PATH variable

Set the CLASSPATH variable Java classes and class libraries are found on CLASSPATH. Before executing a Java program that loads any of the TimesTen JDBC drivers, the CLASSPATH environment variable must contain the class library file:

install_dir/lib/ttjdbc*jdk_ver*.jar

Where *jdk_ver* indicates the version of the JDK that you are using. For example, for JDK 5.0, *jdk_ver* is 5 and the file name would be ttjdbc5.jar. For JDK 6, *jdk_ver* is 6 and the file name would be ttjdbc6.jar.

> **Notes:**
>
> - This variable is set by ttenv or, on Windows, during installation if environment variables are registered (default).
>
> - If multiple JAR files are listed in the CLASSPATH, ensure the TimesTen JAR file is listed first.

On UNIX, CLASSPATH elements are separated by colon. For example (sh type shell):

```
CLASSPATH=.:install_dir/lib/ttjdbc6.jar
export CLASSPATH
```

Or (csh type shell):

```
setenv CLASSPATH .:install_dir/lib/ttjdbc6.jar
```

On Windows, CLASSPATH elements are separated by semicolons.

Also, on Windows, do not use quotes when setting the CLASSPATH environment variable even if a directory path name contains spaces.

For example, this is correct:

```
set CLASSPATH=.;install_dir\lib\ttjdbc6.jar
```

This is incorrect:

```
set CLASSPATH=.;"install_dir\lib\ttjdbc6.jar"
```

If in doubt about the JDK version you have installed on your system, enter:

```
> java -version
```

If you are going to use the JMS/XLA interface, then you also must add the following to your CLASSPATH:

```
install_dir/lib/timestenjmsxla.jar
install_dir/3rdparty/jms1.1/lib/jms.jar
install_dir/lib/orai18n.jar
```

For example, your CLASSPATH would look like the following example (replacing install_dir as appropriate):

```
.:install_dir/lib/ttjdbc6.jar:install_dir/lib/timestenjmsxla.jar
:install_dir/3rdparty/jms1.1/lib/jms.jar:install_dir/lib/orai18n.jar
```

By default, JMS/XLA looks for a configuration file called jmsxla.xml in the current working directory. To use another name or location for the file, you must specify it as part of the environment variable in the InitialContext class and add the location to the CLASSPATH setting. See *Oracle TimesTen In-Memory Database Java Developer's Guide* for more information about the jmsxla.xml configuration file.

Set the shared library path variable Before running a Java program that loads the TimesTen JDBC driver, the shared library path for your system environment variable must be set to include the TimesTen install_dir/lib directory. The name of the variable used for the shared library path depends on the system used.

System	Name of Variable
Linux	LD_LIBRARY_PATH
Solaris	LD_LIBRARY_PATH
AIX	LIBPATH
Windows	PATH
Mac OS X	DYLD_LIBRARY_PATH

Set the THREADS_FLAG variable (UNIX only) The TimesTen JDBC driver uses native threads. Green threads are not supported.

On some UNIX platforms, to use the native threads package, you must set the THREADS_FLAG environment variable to native. How you set the flag depends on your shell.

In csh, the syntax is:

```
setenv THREADS_FLAG native
```

In sh, the syntax is as follows.

```
THREADS_FLAG=native
export THREADS_FLAG
```

Set the PATH variable Ensure the `javac` and `java` executables are both on your executable search path.

TimesTen Quick Start

During installation, you have options to install TimesTen documentation, which includes Quick Start tutorial pages, and TimesTen Quick Start sample programs. Quick Start materials are installed by default under the directory *install_dir*/quickstart. On UNIX you have the option of specifying an alternative location. Regardless of where you install Quick Start, the home page for further information is *install_dir*/quickstart.html.

Quick Start provides tutorials, sample programs, and sample code for administration, access control, application development, replication, and caching, including the following areas.

Configuration and setup:

- Creating, loading, and unloading a TimesTen database

- Setting up user accounts and privileges

- Using TimesTen utilities and built-in procedures for system operations

- Setting up TimesTen Application-Tier Database Cache to cache an Oracle database

- Setting up replication using active standby pairs and Oracle Clusterware

- Configuring a Java EE or J2EE application server to work with TimesTen

- Using Oracle SQL Developer with TimesTen

- Using Oracle Enterprise Manager with TimesTen

Application development:

- C applications using ODBC and XLA

- C applications using Oracle Call Interface (OCI) or the Pro*C/C++ Precompiler

- C++ applications using TimesTen TTClasses and XLA

- Java applications using JDBC and JTA

- PL/SQL applications

Performance and best practices:

- Response time sample program

- Programming tips

- Database schema setup tips

- Hardware configuration tips

- Operating system configuration tips

Through the Quick Start home page, you can find information to set up and run the sample programs if you also installed them.

- Sample program schema and setup: The `build_sampledb` script creates a sample database and demo schema. You must run this before you start using the sample programs.

- Sample program environment and setup: The `ttquickstartenv` script, a superset of the `ttenv` script generally used for TimesTen setup, sets up the demo environment. You must run this each time you enter a session where you want to compile and run any of the sample programs.

- Sample programs and setup: Quick Start sample programs are in subdirectories under the `quickstart/sample_code` directory. For instructions on compiling and running the programs, see the README files in the subdirectories. Also see "Getting Started" and the various API links under "Sample Programs" on the Quick Start home page.

Documentation installation

TimesTen documentation, including Quick Start tutorial pages, is installed along with the TimesTen product unless you choose not to install the documentation. (Quick Start sample programs are installed as a separate option.) Documents are provided in HTML and PDF format. The HTML can be viewed in your browser. The PDF can be viewed with the Adobe Acrobat Reader. If you do not have the Acrobat Reader, it is available from the Adobe Systems website: http://www.adobe.com

Documentation is installed in the *install_dir*/doc directory.

TimesTen documentation is also available through the Oracle Technology Network at the following location:

http://www.oracle.com/technetwork/database/database-technologies/timesten/documentation/index.html

This documentation library may be more current than the library installed with TimesTen.

Installation problems

To avoid problems during installation, ensure you have met all prerequisites. Using information in this installation guide and *Oracle TimesTen In-Memory Database Release Notes*, confirm the following.

- You have installed all required operating system patches.

- You are running a supported version of the operating system.

- You have made all required kernel configuration changes.

- You have sufficient disk space.

- On UNIX, you are a member of the TimesTen administrators group. See "TimesTen instance administrators and users groups" on page 1-15.

- On Windows, you are installing as a member of the local `Administrators` group.

2

Migration, Backup, and Restoration

This chapter describes migration, backup and restoration for the TimesTen database, covering the following topics:

- Overview: copying, migrating, and restoring a database
- Backing up and restoring a database
- Migrating a database
- Altering a database
- Globalization support during migration

Overview: copying, migrating, and restoring a database

TimesTen utilities for copying, restoring, and migrating a database enable you to perform a variety of migration and upgrade tasks. Relevant utilities include the following:

- The `ttBackup` utility: Creates an image copy of the database that is compatible only within the same major release line and on the same platform. This is a general backup utility, but you can also use it to back up data for minor upgrades (for example, from 11.2.2.*x.x* to 11.2.2.*y.y*) or for migrations that do not require any changes to the size or structure of the database.

  This utility saves one checkpoint file and a copy of the log files to create the image.

- The `ttRestore` utility: Restores a database that was backed up using `ttBackup`.

- The `ttMigrate` utility: Exports a database to a binary file or imports a database from a binary file created by `ttMigrate`. (For those familiar with Oracle Database, this is similar in nature to the Oracle Database export and import features.) This utility exports the database in a release-neutral but platform-specific format. You can use it for major upgrades (for example, from 11.2.1.*x.x* to 11.2.2.*y.y*).

  To move between platforms, use the `ttMigrateCS -c` client/server version to export a remote database into a file on your target system, in order to ensure the appropriate format for the target system. Then use `ttMigrate -r` to restore the contents of the file into the database.

  > **Note:** The `ttMigrateCS` version of `ttMigrate` is UNIX-only. On Windows systems, you can do the equivalent by using `ttMigrate` to connect to the source system from the target system through a defined TimesTen client DSN.

- The `ttBulkCp` utility: Writes a database table to an ASCII file, or restores a table to a database from an ASCII file created by `ttBulkCp`.

- The `ttSchema` utility: Outputs the SQL DDL statements used to create the schema of a database or selected objects (such as tables, views, or sequences) of a database. This is useful in conjunction with `ttBulkCp`, to give you the SQL statement to create a table before restoring its data from `ttBulkCp`.

- The `ttDestroy` utility: If the database will be in a new location, you can optionally use this to destroy the old database once the new one is confirmed to be functioning properly.

See "Utilities" in *Oracle TimesTen In-Memory Database Reference* for details about these utilities.

> **Note:** You can alternatively use corresponding C functions for `ttBackup` and `ttRestore`. See "TimesTen Utility API" in *Oracle TimesTen In-Memory Database C Developer's Guide*.

Typical tasks may include the following.

TimesTen upgrades:

- For a TimesTen minor or patch upgrade: Move to a TimesTen release in the same major release line, without changing the size or structure of the database. You can do an offline upgrade using `ttBackup` and `ttRestore`. "Moving to a different minor or patch release of TimesTen (offline upgrade)" on page 3-13.

  You can also, more typically, perform an in-place upgrade of the TimesTen installation. See "In-place upgrades" on page 3-7.

- For a TimesTen major upgrade: Move to a new major release of TimesTen.

  - You can do an offline upgrade using `ttMigrate -c` from the old release to export the data, and `ttMigrate -r` from the new release to import the data. See "Moving to a different major release of TimesTen" on page 3-12.

  - You can do an online upgrade with replication (as applicable for your situation), using `ttMigrate`. See "Online upgrades with replication" on page 3-8.

Database migrations:

- Move to a different system of the same platform type using the same major TimesTen release line. Use `ttBackup` and `ttRestore`. See "Moving a database to a different computer (same platform)" on page 2-5.

- Move to a different hardware or software platform, or from a 32-bit system to a 64-bit system. Use `ttMigrateCS` and `ttMigrate`. See "Moving a database to a different platform" on page 2-6.

Database alterations:

- Move to a different directory on the same system. Use `ttBackup` and `ttRestore`. See "Moving a database to a different directory" on page 2-7.

- Reduce database size. Use `ttMigrate`. See "Reducing database size" on page 2-8.

> **Note:** You can also use this procedure to compact the database in order to reduce fragmentation.

Other tasks:

■ Take a snapshot of a database and then restore the database in the same state. Use `ttBackup` and `ttRestore`.

■ Move selected tables of a database to a new location or platform. You can use `ttMigrate` if the target is the same platform. You can use `ttMigrate` with `ttMigrateCS` if the target is a different platform. (Alternatively, you can use `ttBulkCp` with `ttSchema` to move to a different platform, but `ttMigrate` with `ttMigrateCS` is generally easier and faster.)

■ Change the metadata (for example, change a column name). Use `ttBulkCp` and `ttSchema`.

> **Note:** Reset connection attributes to reflect the changes you made, such as `DataStore` when you move the database to a different directory or `PermSize` when you reduce database size.
>
> See "Connection Attributes" in *Oracle TimesTen In-Memory Database Reference* for details about these connection attributes.

Backing up and restoring a database

The TimesTen backup and restore facility enables you to create a backup of any TimesTen database to restore it at a later time. The primary use for the backup and restore facility is to allow the restoration of a recent state of a database that has been lost. For details about using the TimesTen backup and restore facility, see "ttBackup" and "ttRestore" in *Oracle TimesTen In-Memory Database Reference*.

> **Note:** You cannot use these utilities in a client-only installation.

TimesTen backup features

Every database backup contains the information needed to restore the database as it existed at the *backup point*, which is the time the backup began. Restoration of a database from a given backup restores the modifications of all transactions that committed before the backup point.

A backup operation is atomic: If it completes successfully, it produces a backup you can use to restore a database to the state of its backup point. If it fails for any reason, it leaves the files of any existing backup intact and its backup point unchanged.

TimesTen writes a database backup to a location specified by a *backup path*, which consists of a *directory name* and an optional *basename*. You must specify the backup directory and basename when the backup is created. The basename defaults to the basename of the database itself if you do not specify a basename.

> **Important:**
>
> - Do not manually change the contents of the backup directory. The addition, removal, or modification of any file in the backup directory, except for modifications made by `ttBackup` and `ttRestore` themselves, may compromise the integrity of the backup. In this event, restoration of the database from the backup may not be possible.
>
> - Databases containing cache groups can be backed up with the `ttBackup` utility; however, when restoring such a backup, special consideration is required. 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. To restore a database that contains cache groups, see "Backing up and restoring a database with cache groups" in *Oracle TimesTen Application-Tier Database Cache User's Guide*.

Types of backup provided

TimesTen supports both *full* and *incremental* backups. TimesTen also allows *stream* backups.

- *Stream*: A stream backup writes the database backup file to `stdout`.

- *Full*: A full backup saves the entire database. For full backups, you must have enough disk space available to hold both the existing backup and the new backup, until the new backup succeeds.

- *Incremental*: An incremental backup augments an existing *incremental-enabled* backup of the same database. An incremental backup moves the backup point of an existing backup forward in time by augmenting the backup with all of the transaction log records created since its last backup point.

  An incremental backup typically completes much faster than a full backup, as it has less data to copy. The performance gain of incremental backups over full backups comes at the cost of increased disk usage and longer restoration times. Use incremental backups in concert with full backups to achieve a balance between backup time, disk usage, and restoration time.

  Before you can perform an incremental backup, you must first enable your backup to allow for incremental backups by executing the `ttBackup` utility command with the `-fileFullEnable` or the `-fileIncrOrFull` options. In either case, if your backup was not previously enabled for incremental, a full file backup is performed before the backup is enabled for subsequent incremental backups. TimesTen supports the creation of up to eight incremental-enabled backup instances for each database. If you attempt to start a ninth incremental backup, TimesTen returns an error.

  If you restore a database from a backup, regardless of whether the backup was enabled or disabled for incremental, the restored database is disabled for incremental backups. Thus, if you want incremental backups, you must again execute the `ttBackup` utility command with the `-fileFullEnable` or the `-fileIncrOrFull` option to enable incremental backups.

A set of files containing backup information for a given database, residing at a given backup path, is known as a *backup instance*. A given backup instance must be explicitly enabled for incremental backups.

The files of the existing backup may be modified by a failed full or incremental backup, but not in a way that compromises the ability to restore from them.

The total list of backup types supported by TimesTen are as follows:

Backup type	File or stream	Full or incremental	Incremental-enabled	Comment
fileFull	File	Full	No	This is the default.
fileFullEnable	File	Full	Yes	
fileIncremental	File	Incremental	Yes	Fails if incremental backup is not possible.
fileIncrOrFull	File	Either	Yes	Performs fileIncremental if possible, or fileFullEnable otherwise.
streamFull	Stream	Full	No	
incrementalStop	None	None	No	Takes no backup; just disables existing incremental-enabled backup.

Migrating a database

This section discusses migration of TimesTen databases to a new location, covering the following scenarios:

- Moving a database to a different computer (same platform)
- Moving a database to a different platform

These procedures are for standalone TimesTen databases. Replication and TimesTen Cache scenarios are not considered here.

> **Note:** As discussed in appropriate subsections following, when a database is moved between platforms or bit levels, ttMigrateCS (the client/server version of ttMigrate) or Windows equivalent is required. This is to remotely access the database to be saved, in order to have the data file created on the local target system, assuring proper format for the target system. The Windows equivalent is using ttMigrate to connect to the source system from the target system through a defined TimesTen client DSN.

Moving a database to a different computer (same platform)

You can use the ttBackup and ttRestore utilities to move a database between two computers that have the same major release of TimesTen, the same CPU architecture, and the same operating system.

To copy a database from one system to another with the same CPU architecture and operating system:

1. Disconnect applications from the database.

2. Back up the database on the original system using ttBackup. For example:

   ```
   ttBackup -dir /tmp/dump -fname salesdata SalesData
   ```

3. Move the backup to the new system.

4. Create a DSN for the database on the new system. See "Creating a Data Manager DSN on UNIX" or "Creating a Data Manager DSN on Windows" in *Oracle TimesTen In-Memory Database Operations Guide*.

5. Confirm the DataStore connection attribute is set correctly for the location on the new system.

6. Restore the backup on the new system using ttRestore. For example:

```
ttRestore -dir /tmp/dump -fname salesdata NewSalesData
```

7. As desired, once the database is operational on the new system, use the ttDestroy utility to destroy the database on the old system, and delete the old DSN.

Moving a database to a different platform

The internal format of a TimesTen database differs between platforms. You can use this procedure to move a database from one platform to another, such as from Solaris to Linux or from a 32-bit platform to a 64-bit platform, and reformat it for the target platform.

1. Disconnect applications from the database.

2. From the target system, use ttMigrateCS (or Windows equivalent) to connect in client/server mode to the database on the originating system and use the -c option to create a data file for the database on the target system. For example:

```
ttMigrateCS -c salesdata1 salesdata.dat
```

3. On the target system, create a DSN for the database. See "Creating a Data Manager DSN on UNIX" or "Creating a Data Manager DSN on Windows" in *Oracle TimesTen In-Memory Database Operations Guide*.

4. Recreate the database using AutoCreate=1, such as in the following example:

```
ttIsql -connstr "dsn=salesdata2;AutoCreate=1" -e "quit"
```

The database will be empty at this point.

5. Confirm the DataStore connection attribute is set correctly for the location on the new system.

6. On the target system, import the file created by ttMigrateCS into the new database using the ttMigrate utility with the -r option and the -relaxedUpgrade option. For example:

```
ttMigrate -r -relaxedUpgrade salesdata2 salesdata.dat
```

7. As desired, once the database is operational on the new system, use the ttDestroy utility to destroy the database on the old system, and delete the old DSN.

> **Notes:** Be aware of the following when moving between 32-bit and 64-bit platforms.
>
> - When changing bit levels, the database cannot be involved in a replication scheme.
>
> - TimesTen does not support replication between a 32-bit database and a 64-bit database.

Altering a database

This section discusses scenarios where the database remains on the same system, covering the following topics:

- Moving a database to a different directory

- Reducing database size

These procedures are for standalone TimesTen databases. Replication and TimesTen Cache scenarios are not considered here.

> **Note:** As discussed in appropriate subsections following, when a database is moved between platforms or bit levels, `ttMigrateCS` (the client/server version of `ttMigrate`) or Windows equivalent is required. This is to remotely access the database to be saved, in order to have the data file created on the local target system, assuring proper format for the target system. The Windows equivalent is using `ttMigrate` to connect to the source system from the target system through a defined TimesTen client DSN.

Moving a database to a different directory

The TimesTen daemon identifies a database by the full path name of the database checkpoint files. To move a TimesTen database to a different directory, back up the database using the `ttBackup` utility, create a new DSN definition that specifies the new database path name, then restore the database into its new location using the `ttRestore` utility.

Examples in the following procedure move a database from `/old/SalesData/sales` with a database name `SalesData` to `/new/SalesData/sales` with database name `NewSalesData`, using the `/tmp/dump` directory for temporary storage. (The examples are for a UNIX system. Do the equivalent on a Windows system.)

1. Create a temporary directory. For example:

   ```
   mkdir /tmp/dump
   ```

2. Disconnect applications from the database.

3. Back up the database to the temporary directory using `ttBackup`. For example:

   ```
   ttBackup -dir /tmp/dump -fname salesdata SalesData
   ```

4. Unload the database from memory. For more information, see "Unloading a database from memory" on page 3-9.

5. Create a DSN definition for the new location and specify the new database path. For example: `/new/SalesData/sales/NewSalesData`.

6. Confirm the `DataStore` connection attribute is set correctly for the new location.

7. Restore the backup in the new location. For example:

   ```
   ttRestore -dir /tmp/dump -fname salesdata NewSalesData
   ```

8. Remove the temporary directory. For example:

   ```
   rm -r /tmp/dump
   ```

9. After you confirm the database is operational in the new location, optionally use the `ttDestroy` utility to destroy it in the old location, and delete the old DSN.

Reducing database size

Once a database has been defined with a particular size for the permanent region (indicated by the `PermSize` DSN attribute), it cannot be loaded at a smaller size, even if tables or rows are deleted. A copy of the database made with `ttBackup` also has the permanent region size of the database embedded in it.

To reduce the allocated size of the permanent region of a database, you must save a copy using the `ttMigrate` utility then re-create the database with a smaller permanent region size and restore the data.

Perform these steps to reduce the permanent region size of a database:

1. Disconnect applications from the database.

2. Use the `ttMigrate -c` option to create a data file for the database. For example:

   ```
   ttMigrate -c salesdata /tmp/salesbackup
   ```

3. Unload the database from memory. For more information, see "Unloading a database from memory" on page 3-9.

4. To modify the original DSN rather than create a new one, you must destroy the original database files using the `ttDestroy` utility before restoring from the backup.

5. Create a new DSN definition for the new copy of the database with a smaller `PermSize` value.

6. Recreate the database using `AutoCreate=1`, such as in the following example:

   ```
   ttIsql -connstr "dsn=salesdata;AutoCreate=1" -e "quit"
   ```

   The database will be empty at this point.

7. Restore the backup, using the `ttMigrate -r` and `-relaxedUpgrade` options. For example:

   ```
   ttMigrate -r -relaxedUpgrade salesdata /tmp/salesbackup
   ```

   Notes:

   - The permanent region size of a database cannot be reduced below the size that is required by the data currently stored in the database. This value can be determined by querying the `perm_in_use_size` column of the system table `sys.monitor`.

   - You can also use this procedure to compact the database in order to reduce fragmentation caused by partially full table pages or fragmentation of the heap buffers that store index nodes and out-of-line values.

Globalization support during migration

The `ttMigrate` utility saves one or more migration objects from a TimesTen database into a binary data file or restores the objects from the binary data files into a TimesTen database. Migration objects include tables, cache group definitions, views, and sequences.

The following topics describe what occurs with globalization issues during migration:

- Object migration and character sets

- Migration and length semantics
- Migrating linguistic indexes
- Migrating cache group tables

Also see the description of `ttMigrate` in the *Oracle TimesTen In-Memory Database Reference*.

Object migration and character sets

The `ttMigrate` utility tags each object it saves with the object's character set. By default, `ttMigrate` stores object data in the database character set; however, you can select a different character set by using the `-saveAsCharset` option. You can specify this option in create mode (`-c`) or append mode (`-a`).

When you use `ttMigrate` to restore an object, the data is implicitly converted to the database character set of the target database, if necessary. Character set conversion may result in data loss if the database character set of the target database cannot represent all of the data that it receives.

> **Notes:**
>
> - If you use the `-saveAsCharset` option when saving an object, you cannot use `ttMigrate -r -exactUpgrade` to restore it.
> - The `ttMigrate` utility issues a warning whenever there is an implicit or explicit character set conversion while saving or restoring data.
> - If character set conversion is requested when migrating databases, performance may be slower than if character set conversion is not requested.

If you know that the data is encoded in the database character set of the target database, use the `-noCharsetConversion` option when restoring (`-r`). When using this option, `ttMigrate` assumes that the data uses the same database character set of the target database.

When you restore untagged character data from a database that was created before release 7.0 into a database from release 7.0 and later, `ttMigrate` treats the data as if it is in the database character set of the target database. All TimesTen databases from release 7.0 and earlier use the `TIMESTEN8` character set.

If you are migrating from a TimesTen database that uses `TIMESTEN8` to a TimesTen database that uses a different character set, the following may occur:

- The query result may differ in the new TimesTen database with the new character set. The user application may work on a multibyte character set and use the `TIMESTEN8` character set to store the character string as it is. When querying the data using the `LIKE` predicate (or any scalar functions) to match a substring, the query engine may match a binary pattern that does not begin or end at a character boundary under `TIMESTEN8` character set, since it is a single byte character set. Every byte is treated as a character even it is actually in the middle of a multibyte character.

  Another possible issue arises if the user partitions a long character string and stores it in separate rows. The string may be reconstructed later by concatenating the values from multiple rows. This may work with the `TIMESTEN8` character set.

However, when using a multibyte character set, if the partition is not on the character boundary, the string value can be changed. In this case, ensure that the string is partitioned on the character boundary.

■ Performance may be slower with queries that use predicates or scalar functions on character strings in databases with a character set other than `TIMESTEN8`.

Migration and length semantics

The `ttMigrate` utility saves length semantic information about `CHAR` and `VARCHAR2` columns. It restores the length semantic information when restoring objects into databases created in TimesTen release 7.0 or later.

When objects are migrated back into a TimesTen release before 7.0, columns with character semantics are converted to byte semantics and the column length is adjusted to match the byte length of the original columns.

When objects are migrated from a release before 7.0 to release 7.0 and later, byte semantics is used.

Migrating linguistic indexes

The `ttMigrate` utility supports migration of linguistic indexes into TimesTen releases that support them. When migrating back to a TimesTen release before 7.0, `ttMigrate` issues a warning indicating that the linguistic indexes cannot be restored. Migration of the table proceeds without the linguistic indexes.

Migrating cache group tables

You cannot restore cache group tables containing `NCHAR`/`NVARCHAR2` columns to a release before 7.0. Releases before 7.0 do not allow these data types in cache group tables.

3

TimesTen Upgrades

This chapter describes the process of upgrading from one release of TimesTen to a newer release of TimesTen. It includes the following topics:

- Preliminary considerations
- Upgrade modes
- Performing an in-place upgrade
- Performing an offline upgrade
- Performing an online upgrade with replication
- Performing an upgrade with active standby pair replication
- Performing an offline TimesTen upgrade when using Oracle Clusterware
- Performing an online TimesTen upgrade when using Oracle Clusterware
- Upgrades when using parallel replication
- Performing a Client/Server online upgrade
- Record of upgrades with replication configured

> **Note:** For general information about TimesTen utilities to use during upgrades, refer to "Overview: copying, migrating, and restoring a database" on page 2-1.

Preliminary considerations

This section covers some areas to consider before upgrading TimesTen:

- Data type considerations
- Database character set considerations
- Location of existing database files
- Access control when upgrading from releases prior to 11.2.1
- Replication considerations

Data type considerations

This section covers data type considerations, particularly those that are relevant if you are upgrading from a release of TimesTen earlier than 7.0. It discusses the following topics.

- Data type compatibility

- Data type considerations when upgrading from a TimesTen release before 7.0

Data type compatibility

TimesTen supports a selection of Oracle Database data types in addition to the original TimesTen data types that are maintained for backward compatibility. For details on all data types, see "Type specifications" in *Oracle TimesTen In-Memory Database SQL Reference*. Because some Oracle Database data types have the same names as the backward-compatible TimesTen data types, a set of aliases exists for addressing the data types. Which data types the aliases refer to depends on the `TypeMode` setting for the database. See "TypeMode" in *Oracle TimesTen In-Memory Database Reference* for more information.

TimesTen backward-compatible data types are replication-compatible with the data types in releases of TimesTen before 7.0. However, TimesTen backward-compatible data types are not compatible with TimesTen Cache. Use only Oracle Database data types with TimesTen Cache. To use TimesTen Cache, you must convert any original TimesTen data types to the Oracle Database data types when performing a database migration with `ttMigrate`, described in *Oracle TimesTen In-Memory Database Reference*. See "Converting TimesTen data types to Oracle Database data types", following shortly, for details.

Oracle Database data types are not replication-compatible with releases of TimesTen before 7.0. To perform an upgrade that requires replication with a release of TimesTen from before 7.0, you must upgrade the original data types as TimesTen data types. See "Preserving TimesTen data types" on page 3-3 for more information.

Data type considerations when upgrading from a TimesTen release before 7.0

When performing an upgrade from a release before TimesTen 7.0, you must choose whether to preserve the data types in your database as TimesTen data types, or convert them to Oracle Database data types. Your planned use for the database and your preferred upgrade method have an impact on this decision.

The following subsections cover each approach:

- Converting TimesTen data types to Oracle Database data types

- Preserving TimesTen data types

Converting TimesTen data types to Oracle Database data types If you intend to use your database with TimesTen Cache, you must convert your TimesTen data types to Oracle Database data types.

To convert the data types from a release before TimesTen 7.0 to Oracle Database data types, you must use the `-convertTypesToOra` option for `ttMigrate` (described in *Oracle TimesTen In-Memory Database Reference*) when you restore your database as part of your upgrade procedure. For example, if you restore the database `salesdata` as part of an upgrade procedure, you may use the following to upgrade the data types to Oracle Database data types:

```
ttMigrate -r -convertTypesToOra salesdata salesdata.mig
```

See "TimesTen to Oracle Database data type conversions" in *Oracle TimesTen In-Memory Database Reference* for more information.

> **Notes:**
>
> - You cannot perform an online upgrade using replication if you are converting from TimesTen to Oracle Database data types, as data types may only replicate to like data types. (This replication restriction is not an issue when upgrading from TimesTen 7.0 or later and data types have been converted to Oracle Database data types.)
>
> - Because the Oracle Database and TimesTen versions of some data types behave slightly differently, you should thoroughly test any applications written for releases of TimesTen before 7.0 with the Oracle Database data types before deploying them with the new release of TimesTen.

Preserving TimesTen data types If you intend to perform an online upgrade using replication, you must preserve your data types as TimesTen data types. See "Online upgrades with replication" on page 3-8 for more information.

If you choose to preserve the data types in a database from a release before TimesTen 7.0 as TimesTen data types, you are not required to use any special options when restoring the database with `ttMigrate`. The data types from a release before TimesTen 7.0 are automatically restored as TimesTen data types.

> **Note:** The default `TypeMode` setting is `0`, for Oracle type mode. In this mode, a data type such as `CHAR` has the semantics of the Oracle Database `CHAR` data type rather than the TimesTen `CHAR` data type. To guarantee compatibility with applications written for TimesTen releases before 7.0, you should configure the DSN for your database with a `TypeMode` setting of `1` before restoring the database with `ttMigrate` (described in *Oracle TimesTen In-Memory Database Reference*) as part of the upgrade procedure.
>
> See "TypeMode" in *Oracle TimesTen In-Memory Database Reference* for additional information.

Database character set considerations

This section covers database character set considerations, including those that are relevant if you are upgrading from a release of TimesTen earlier than 7.0. It discusses the following topics:

- Database character set specification

- Character set restrictions when upgrading from a TimesTen release before 7.0

- Database character set conversion

Database character set specification

TimesTen requires a database to be configured to support a specific character set when it is created. The character set for the database is specified using the database attribute `DatabaseCharacterSet`, described in *Oracle TimesTen In-Memory Database Reference*. The value of this attribute determines which characters may be input to and output from character fields, and how character data is stored and sorted. See "Choosing a database character set" in *Oracle TimesTen In-Memory Database Operations Guide* and

"Supported character sets" in *Oracle TimesTen In-Memory Database Reference* for more information.

Be aware of the following points.

- If you plan to use the database with TimesTen Cache, or you plan to use the `ttLoadFromOracle` built-in procedure to populate TimesTen tables from an Oracle database, you must specify a value for `DatabaseCharacterSet` that is the same as the character set specified for the Oracle database to which the TimesTen database connects.

- Replication is not possible between databases with different character sets.

Character set restrictions when upgrading from a TimesTen release before 7.0

If you perform an upgrade from a release before TimesTen 7.0, consider the following important restrictions.

- Because databases created with releases of TimesTen before 7.0 do not have a database character set specified, a special database character set, `TIMESTEN8`, was created, which allows replication compatibility between databases created by the current release of TimesTen and those created by earlier releases. If you plan to perform the TimesTen upgrade as an online upgrade with replication (see "Performing an online upgrade with replication" on page 3-14), then you must specify a `DatabaseCharacterSet` of `TIMESTEN8` in your DSN.

- If you use TimesTen Client/Server and intend to connect to a database in an upgraded TimesTen instance with an application linked to a Client ODBC library from a release before TimesTen 7.0, you must specify a `DatabaseCharacterSet` of `TIMESTEN8` in your DSN to ensure compatibility. See "Performing a Client/Server online upgrade" on page 3-48.

> **Note:** The `TIMESTEN8` database character set is intended for use only when transitioning from a release of TimesTen before 7.0. When you no longer need your database to replicate to a pre-7.0 release of TimesTen, or to connect to a pre-7.0 client application, you should use `ttMigrate`, described in *Oracle TimesTen In-Memory Database Reference*, to convert your database to a database character set other than `TIMESTEN8`. See the next section, "Database character set conversion", for details.

Database character set conversion

In some cases, you may need to change the configured database character set as part of the upgrade process. There are two particular cases in which a database character set change is required:

- You must change your database character set from the one that you originally specified to a new one that fits your requirements more closely.

- You have specified the database character set as `TIMESTEN8` to upgrade from a release of TimesTen before 7.0 using online upgrade with replication or Client/Server. After the upgrade is complete for all databases and client applications, you should convert each database from this special transitional character set to the national character set you prefer to use for your region.

You can use `ttMigrate` to convert a database from any character set to any other character set by completing the following steps.

1. Save the database to a file using `ttMigrate`. For example, to save the database `salesdata` to the file `salesdata.mig`, use the command.

   ```
   ttMigrate -c salesdata salesdata.mig
   ```

2. Destroy the database, using `ttDestroy`.

   ```
   ttDestroy salesdata
   ```

3. Change the value of the DSN attribute `DatabaseCharacterSet` for your database to the value specifying the new character set. For example, if you want your database to use the `WE8ISO8859P1` character set, use the following line in your `ODBCINI` file.

   ```
   DatabaseCharacterSet=WE8ISO8859P1
   ```

4. Load the database from the file using `ttMigrate`. TimesTen automatically converts the character data from the character set the file was saved with to the character set specified in the DSN.

   ```
   ttMigrate -r salesdata salesdata.mig
   ```

   > **Notes:**
   >
   > - It is possible that character data is lost in the conversion process if no mapping exists from one character set to the other for a given character.
   >
   > - If you are upgrading from a release earlier than TimesTen 7.0 and converting from the `TIMESTEN8` character set, use the `ttMigrate -noCharsetConversion` option. This ensures that no character values are changed when the data is loaded into the DSN using the new character set. For example:
   >
   >   ```
   >   ttMigrate -r -noCharsetConversion salesdata salesdata.mig
   >   ```

Location of existing database files

As noted in "Considerations for locations of database files and other user files" on page 1-14, it is strongly recommended that you not store any database files or user files under the TimesTen installation path, where they may be removed during upgrade or uninstallation operations.

If, however, you do have databases under the installation directory, you can use the following general steps to move each database elsewhere:

1. Use the `ttBackup` utility to back up the database.

2. Use the `ttDestroy` utility to destroy the database.

3. Update the DSN definition for the database so that the `DataStore` and `LogDir` connection attributes reference a location outside of the installation path.

4. Use the `ttRestore` utility to restore the database to the new location.

For more information about these utilities and attributes, see "Utilities" and "Connection Attributes" in *Oracle TimesTen In-Memory Database Reference*.

Also see "Moving a database to a different directory" on page 2-7.

Access control when upgrading from releases prior to 11.2.1

Access control is mandatory in TimesTen beginning with release 11.2.1. If you have been using earlier releases of TimesTen without access control and you do not want to initially secure your database objects after upgrading to the current release, you can grant the ADMIN system privilege to PUBLIC using the following SQL command:

```
GRANT ADMIN TO PUBLIC;
```

If you grant the ADMIN privilege to PUBLIC, all users have unrestricted access to all database objects and are able to perform all administrative tasks, except for those tasks that must be performed as the instance administrator. Granting the ADMIN privilege to PUBLIC should be viewed as a temporary workaround, for upgrade purposes only.

> **Note:** This is never recommended as a long-term approach, as it results in an inherently insecure system.

Replication considerations

When using the ttMigrate utility to upgrade a database that is part of a replication scheme, you must restore the replication scheme (using ttMigrate -r) to the same database from which it was originally created (using ttMigrate -c). Attempting to restore the migration file to a different database results in an error, and the replication scheme will not be restored.

Upgrade modes

The type of upgrade you can perform—in-place, online, or offline—depends on the level of upgrade—major or minor—among other factors. A *major upgrade* is one between different major releases of TimesTen, such as from 11.2.1.*x.x* to 11.2.2.*x.x*. By contrast, TimesTen databases from different patch releases of the same major release are structurally equivalent or identical, so upgrading to a new patch release, such as from 11.2.2.6.0 to 11.2.2.7.0, is considered a minor upgrade.

The following subsections describe the types of upgrades:

- In-place upgrades
- Offline upgrades
- Online upgrades with replication
- Online upgrades with Client/Server

For a minor upgrade, you are generally not required to migrate your existing databases and can perform an in-place upgrade (on UNIX systems only).

For a major upgrade, you can usually perform an online upgrade, depending on your situation.

Situations that require offline upgrades are noted in that section.

> **Important:** There are steps you should take before performing any upgrade:
>
> - While preparing for the upgrade, ensure that all databases are unloaded from memory before upgrading TimesTen. For a description of the procedures for unloading a database from memory, see "Unloading a database from memory" on page 3-9.
>
> - During the installation of any new release, you should disconnect your application. The TimesTen daemon process from the prior release is stopped during uninstallation, effectively disconnecting all applications that were still connected to any databases in the instance.

In-place upgrades

In-place upgrades are available on UNIX systems for moving to a new patch release of TimesTen, such as from release 11.2.2.6.0 to release 11.2.2.7.0. There are also equivalent uninstallation and installation steps you can perform on Windows systems.

To perform an in-place upgrade, use the UNIX installer option to upgrade an existing instance. The uninstallation of the old release and installation of the new release take place in a single operation. New software libraries and binaries are installed in the same location as before. If you choose default responses, existing databases will be preserved (assuming they were not located under the *install_dir* directory) without the necessity of using the TimesTen backup and migration utilities.

During the in-place upgrade, all applications must disconnect from databases in the TimesTen instance being upgraded.

On Windows systems, you can perform equivalent steps to move between patch releases by first uninstalling TimesTen, keeping your DSNs when prompted, then installing the new release. As with an in-place upgrade on UNIX, this does not require use of the backup and migration utilities.

See "Performing an in-place upgrade" on page 3-9.

> **Important:** Before upgrading, ensure that you do not have any databases or important files under the TimesTen installation directory. See "Considerations for locations of database files and other user files" on page 1-14 and "Location of existing database files" on page 3-5 for related information.

Offline upgrades

While an offline upgrade is in progress, databases are not available to applications. Offline upgrades usually require enough disk space for two copies of each database in the TimesTen instance being upgraded.

You can use offline upgrades to do the following:

- Move to a new major release of TimesTen, using `ttMigrate`.

- Move to a new minor or patch release of TimesTen, using `ttBackup` and `ttRestore`.

> **Note:** For a minor or patch upgrade, it is more typical to use an in-place upgrade or Windows equivalent, as discussed in the preceding section, "In-place upgrades".

Offline upgrades require all applications to disconnect from TimesTen during the upgrade procedure. Databases must also be unloaded from shared memory. Offline upgrades require you to use the TimesTen `ttMigrate` utility (for a major upgrade) or `ttBackup` utility. (See "ttMigrate" and "ttBackup" in *Oracle TimesTen In-Memory Database Reference*.)

See "Performing an offline upgrade" on page 3-11.

Online upgrades with replication

When upgrading to a new major release of TimesTen, you may have a mission-critical database that must remain continuously available to your applications. You can use TimesTen replication to keep two copies of a database synchronized, even when the databases are from different releases of TimesTen, allowing your applications to stay connected to one copy of the database while the TimesTen instance for the other database is being upgraded. When the upgrade is finished, any updates that have been made on the active database are transmitted immediately to the database in the upgraded TimesTen instance, and your applications can then be switched with no data loss and no downtime. For more information, see "Performing an online upgrade with replication" on page 3-14.

The online upgrade process supports only updates to user tables during the upgrade. Data definition changes such as `CREATE TABLE` or `CREATE INDEX` are not replicated. In addition, all tables to be replicated must have a `PRIMARY KEY` or a unique index on non-nullable columns.

Because two copies of the database (or two copies of each database, if there are more than one) are required during the upgrade, you must have available twice the memory and disk space normally required, if performing the upgrade on a single system.

> **Notes:**
>
> - Online major upgrades for active standby pairs with cache groups are only supported for read-only cache groups.
>
> - Online major upgrades for active standby pairs that are managed by Oracle Clusterware are not supported.
>
> - Replication is not supported between 32-bit and 64-bit databases. Tables migrated using the `ttMigrate -inline` command should not be replicated with tables that do not have inline columns, as inline columns cannot be replicated with not inline columns.

Online upgrades with Client/Server

If you are upgrading a TimesTen Client/Server installation to a new major release, you can minimize downtime by performing a Client/Server online upgrade. During this process, TimesTen clients from the previous release are able to continue to communicate with a database in a TimesTen instance that has been upgraded to the new release. See "Performing a Client/Server online upgrade" on page 3-48.

The Client/Server online upgrade process minimizes, but does not eliminate, the interruption of client application access to the database in the TimesTen instance being upgraded. To maintain nearly continuous availability of a database to all clients, you may use the approach discussed in the preceding section, "Online upgrades with replication". This procedure keeps an identical copy of the database available to the old release of TimesTen Server, while you perform the upgrade. Once the original database is available to the new release of TimesTen Server, you may stop the old release and start the new release, listening on the same port. The only interruption in availability using this method is the very brief period after the old server is stopped and before the new server is started.

Performing an in-place upgrade

You can perform an in-place upgrade of a TimesTen instance on a UNIX system if you are moving between patch releases (staying within the same major TimesTen release), such as upgrading from one 11.2.2.*x.x* release to a higher 11.2.2.*x.x* release. This does not require migrating the databases.

On Windows systems, there is no installer option for such an upgrade. Instead, even when just moving between patch releases, you must perform a separate uninstallation and installation of TimesTen. As with an in-place upgrade on UNIX, migrating the databases is not required.

Before the upgrade, for each database in the instance, all applications must disconnect from the database and the database must be unloaded from shared memory.

The following subsections discuss the process for performing an in-place upgrade or Windows equivalent:

- Unloading a database from memory
- Performing the upgrade
- Reloading a database into memory

Unloading a database from memory

A TimesTen database remains loaded in shared memory if any applications or TimesTen agents, such as the cache agent or replication agent, are connected to it. A database may also be kept in shared memory for particular RAM policy configurations set using the `ttAdmin` utility, even when no applications or agents are connected. (See "ttAdmin" in *Oracle TimesTen In-Memory Database Reference*.)

Use the following steps to unload a database from memory in a TimesTen instance being upgraded. All steps must be completed for each database in the TimesTen instance.

1. Disconnect all applications from the database.

2. If the replication agent is running on the database, set the replication state to `pause` and stop the replication agent. This example references an active standby pair, `activedb` and `standbydb`, where `activedb` belongs to the TimesTen instance being upgraded. The example sets the replication state from `activedb` to `standbydb` to `pause`, then stops the replication agent on `activedb`. Assume `standbydb` does not belong to the TimesTen instance being upgraded.

    ```
    ttRepAdmin -receiver -name activedb -state pause standbydb
    ttAdmin -repStop activedb
    ```

3. If the cache agent is running on the database, stop the cache agent. This example references a database `cachedb` that uses TimesTen Cache and belongs to the TimesTen instance being upgraded. (The example does not consider replication. If the cache agent is running, this step must be performed regardless of whether the database is configured for replication.)

```
ttAdmin -cacheStop cachedb
```

4. Unload the database from memory using `ttAdmin`. The first example assumes RAM policy is set to `manual` and references a database `upgradedb` that belongs to the TimesTen instance being upgraded. (The example does not consider replication or cache. This step must be performed regardless of whether the database is configured for replication or cache.)

```
ttAdmin -ramUnload upgradedb
```

Or, if the RAM policy is set to `always`, change it to `manual`, then unload the database from memory. For example:

```
ttAdmin -ramPolicy manual -ramUnload upgradedb
```

Or, if the RAM policy is `inUse` and a grace period is set, set the grace period to 0 or wait for the grace period to elapse. This results in the database being unloaded. For example:

```
ttAdmin -ramGrace 0 upgradedb
```

5. Use the `ttStatus` utility to verify that the database has been unloaded from memory. The utility will indicate "There are no connections to the data store." (When a database is loaded into memory, there are automatic subdaemon connections.)

See "ttStatus" in *Oracle TimesTen In-Memory Database Reference*.

6. Optionally stop the TimesTen daemon. (During an in-place upgrade, the daemon is automatically stopped.)

```
ttDaemonAdmin -stop
```

Also see "Reloading a database into memory", following shortly.

> **Note:** The `activedb`, `cachedb` and `upgradedb` databases, for example, can be the same database. A TimesTen database can have both replication and cache configured, and a RAM policy other than manual.

Performing the upgrade

To perform an in-place upgrade on a UNIX system, use the TimesTen installer option to upgrade an existing instance (not install a new instance). UNIX installer options are covered in "Installing TimesTen" on page 1-18.

The uninstallation of the old release and installation of the new release take place in a single process. New software libraries and binaries are installed in the same location as before. You are prompted whether to remove files in *install_dir*/info and *install_dir*/network/admin/samples. Presuming you choose the default "no" responses, existing databases will be preserved (assuming they were not located under the *install_dir* directory) and their configurations will be preserved (in the info directory) without the databases having to be migrated.

On Windows systems, there is no installer option for such an upgrade. Instead, when moving between patch releases (staying within the same major TimesTen release), you must perform an uninstallation of TimesTen followed by an installation of the new release. Uninstallation on Windows is covered in "Uninstalling TimesTen on Windows systems" on page 1-28. Presuming you choose to save your DSNs while uninstalling, existing databases will be preserved (assuming they were not located under the *install_dir* directory) and their configurations will be preserved without the databases having to be migrated. Installation on Windows is covered in "Installing TimesTen" on page 1-23.

> **Important:** Before upgrading, ensure that you do not have any databases or important files under the TimesTen installation directory. See "Considerations for locations of database files and other user files" on page 1-14 and "Location of existing database files" on page 3-5 for related information.

Reloading a database into memory

After unloading the databases from memory (as shown in "Unloading a database from memory" on page 3-9) and upgrading the TimesTen instance, use commands such as the following to reload a database into memory. (These examples are independent of each other.) Such steps must be performed, as applicable, for each database in the instance that was upgraded.

1. Confirm the TimesTen daemon has started. (During an in-place upgrade, the daemon should start automatically.) If necessary:

    ```
    ttDaemonAdmin -start
    ```

2. Use `ttAdmin` to reload the database into memory. If the RAM policy was `manual` for the database `upgradedb` before the upgrade:

    ```
    ttAdmin -ramLoad upgradedb
    ```

    Or, if the RAM policy was `always` before the upgrade, change it back to `always`. This results in the database being reloaded. For example:

    ```
    ttAdmin -ramPolicy always upgradedb
    ```

    Or, if the RAM policy is `inUse` and you changed the grace period to 0, change it back to its original setting. For example:

    ```
    ttAdmin -ramPolicy inUse -ramGrace 300 upgradedb
    ```

3. To restart replication (assuming `activedb` of an active standby pair was in the TimesTen instance being upgraded):

    ```
    ttRepAdmin -receiver -name activedb -state start standbydb
    ttAdmin -repStart activedb
    ```

4. To restart TimesTen Cache (assuming `cachedb` uses cache and was in the TimesTen instance being upgraded):

    ```
    ttAdmin -cacheStart cachedb
    ```

Performing an offline upgrade

You can do an offline upgrade by exporting any database into an external file and then importing the database with the desired changes. These upgrade procedures require

that all applications be disconnected from the database and that the database be unloaded from shared memory. For applications that require continuous availability, see "Performing an online upgrade with replication" on page 3-14.

This section covers the following topics:

- Moving to a different major release of TimesTen
- Moving to a different minor or patch release of TimesTen (offline upgrade)

These procedures are for standalone TimesTen databases. Replication and TimesTen Cache scenarios are not considered here.

Moving to a different major release of TimesTen

You can have multiple TimesTen major releases installed on a system at the same time. However, TimesTen databases created by one major release cannot be accessed directly by applications of a different major release. To migrate data between TimesTen major releases, for example from TimesTen 11.2.1 to 11.2.2, you must export the data using the `ttMigrate` utility from the old release and import it using the `ttMigrate` utility from the new release.

> **Note:** For additional information about particular scenarios where an offline upgrade is required for a major TimesTen upgrade, see "Offline upgrades for an active standby pair with cache groups" on page 3-31 and "Performing an offline TimesTen upgrade when using Oracle Clusterware" on page 3-34.

The general steps in an offline upgrade include the following. This procedure assumes you will install the new release while the old release is still installed, which is generally advisable until you are sure databases are correctly configured and fully operational on the new release.

On the old release, for each database:

1. Disconnect all applications from the database.

2. Use `ttMigrate` with the `-c` option to save a copy of the database contents. For example:

   ```
   ttMigrateCS -c salesdata salesdata.dat
   ```

3. Unload the database from memory. For more information, see "Unloading a database from memory" on page 3-9. For an offline upgrade, stop the TimesTen daemon (`ttDaemonAdmin -stop`) after unloading the database from memory.

4. As necessary, use the `ttDestroy` utility to destroy the old database. This step is required if the new database will be in the same location as the old database.

   Important: It is advisable to first use `ttBackup` to back up the database, as a precaution. The backup can be deleted once the database is fully operational on the new release.

Once the preceding steps have been completed for each database, install the new release of TimesTen. For more information, see Chapter 1, "TimesTen Installation".

Then confirm that the `PATH` and `LD_LIBRARY_PATH` environment variables on your system are set appropriately to point to the new TimesTen release.

On the new release, do the following for each database.

1. Confirm the `DataStore` connection attribute is set correctly after the upgrade.

2. Recreate the database using `AutoCreate=1`, such as in the following example:

```
ttIsql -connstr "dsn=salesdata;AutoCreate=1" -e "quit"
```

The database will be empty at this point.

3. Use `ttMigrate` with the `-r` and (typically) `-relaxedUpgrade` options to restore the backed up database to the new TimesTen release. For example:

```
ttMigrate -r -relaxedUpgrade salesdata salesdata.dat
```

Once databases are correctly configured and fully operational on the new release, optionally uninstall the old release. Uninstallation is also covered in Chapter 1, "TimesTen Installation".

Moving to a different minor or patch release of TimesTen (offline upgrade)

You can use the `ttBackup` and `ttRestore` utilities to move to a new minor or patch release within the same major release line.

> **Notes:**
>
> - For a minor or patch upgrade, it is more typical to use an in-place upgrade or Windows equivalent, as discussed in "Performing an in-place upgrade" on page 3-9.
>
> - On Windows, you cannot have two different releases from the same major release (such as 11.2.2.6.0 and 11.2.2.7.0) installed simultaneously on the same system.

The general steps in an offline upgrade to a new minor or patch release include the following.

On the old release, for each database:

1. Disconnect all applications from the database.

2. Back up the database on the original system using `ttBackup`. For example:

```
ttBackup -dir /tmp/dump -fname salesdata SalesData
```

3. Unload the database from memory. For more information, see "Unloading a database from memory" on page 3-9. For an offline upgrade, stop the TimesTen daemon (`ttDaemonAdmin -stop`) after unloading the database from memory.

4. As necessary, use the `ttDestroy` utility to destroy the old database. This step is required if the new database will be in the same location as the old database.

Once the preceding steps have been completed for each database, uninstall the old release and install the new release of TimesTen. For more information, see Chapter 1, "TimesTen Installation".

Then confirm that the `PATH` and `LD_LIBRARY_PATH` environment variables on your system are set appropriately to point to the new TimesTen release.

On the new release, for each database:

1. Confirm the `DataStore` connection attribute is set correctly after the upgrade.

2. Restore the backup on the new system using `ttRestore`, such as in the following example.

```
ttRestore -dir /tmp/dump -fname salesdata SalesData
```

Performing an online upgrade with replication

"Performing an offline upgrade" on page 3-11 discussed how to perform various maintenance operations on TimesTen databases that require that all applications be stopped. This section describes how to use the TimesTen replication feature to perform online upgrades for applications that require continuous data availability.

Typically, applications that require high availability of their data use TimesTen replication to keep at least one extra copy of their databases up to date. An online upgrade works by keeping one of these two copies available to the application while the other is being upgraded. The procedures described in this section assume that you have a bidirectional replication scheme configured and running for two databases, as described in "Unidirectional or bidirectional replication" in *Oracle TimesTen In-Memory Database Replication Guide*.

Be aware of the following:

- Replication functions across releases only if the database of the more recent release of TimesTen was upgraded from a database of the older release of TimesTen. A database created in the more recent release of TimesTen is not guaranteed to replicate correctly with the older release. For example, replication between a database created in a 7.0.6.*x.x* release of TimesTen and a database created in an 11.2.2.*x.x* release of TimesTen is not supported. However, if one database was created in the 7.0.6.*x.x* release, and the peer database was also created in the 7.0.6.*x.x* release and then migrated to a TimesTen 11.2.2.*x.x* release, replication between them is supported.

- For active standby pairs, as applicable, see "Online upgrades for an active standby pair with no cache groups" on page 3-20 and "Online upgrades for an active standby pair with cache groups" on page 3-25. Online major upgrades for active standby pairs with cache groups are only supported for read-only cache groups. Instead see "Offline upgrades for an active standby pair with cache groups" on page 3-31.

- For use of Oracle Clusterware, as applicable, see "Performing an online TimesTen upgrade when using Oracle Clusterware" on page 3-35. Online major upgrades are not supported for active standby pairs managed by Oracle Clusterware. Instead see "Performing an online TimesTen upgrade when using Oracle Clusterware" on page 3-35.

The following sections describe how to perform an online upgrade with replication.

- Procedural overview
- Limitations
- Requirements
- Upgrade steps
- Online upgrade example

Procedural overview

Upgrading a database from an older TimesTen release to a newer release is performed by disconnecting all applications from one of two replicated copies of a database,

making a backup of the database with the `ttMigrate` utility from the older release, loading the backup into a newer release database using the `ttMigrate` utility from the newer release, and then reconnecting all applications to the database in the upgraded TimesTen instance.

The general steps in an online upgrade include the following.

1. Disconnect all applications from the database in the TimesTen instance being upgraded.

2. Shut down replication in the TimesTen instance being upgraded.

3. Back up the database residing in the TimesTen instance being upgraded using the older release's `ttMigrate` with the `-c` option.

4. Install the newer release of TimesTen on the system being upgraded.

5. Restore the replicated database in the upgraded TimesTen instance using the newer release's `ttMigrate` with the `-r` option.

6. Restart replication on the upgraded TimesTen instance.

7. Reconnect all applications to the database in the upgraded TimesTen instance.

> **Note:** After `ttMigrate` has been used, all autorefresh cache groups in post-upgrade database that is not part of an active standby pair have `AUTOREFRESH STATE` set to `OFF`, no matter how it was set on the database before the upgrade. Reset `AUTOREFRESH STATE` to `ON` by using the `ALTER CACHE GROUP` statement.

To maintain continuous availability, applications continue to run on one copy of the database while the upgrade is performed on the TimesTen instance of the disconnected copy of the database. TimesTen replication retains updates made to the active copy of the database during the upgrade period, and then transfers and applies the updates to the post-upgrade database when replication is restarted. When the replicated updates have been completely applied, the applications may be reconnected to the post-upgrade database.

Limitations

Online upgrades can be performed only with databases for which all the user tables meet the replication requirements. All the user tables must contain either a `PRIMARY KEY` declaration or have a unique index declared over non-nullable columns.

Requirements

To perform online upgrades with replication, replication must be configured to use static ports. See "Port assignments" in *Oracle TimesTen In-Memory Database Replication Guide*.

If you are performing an online upgrade on a single system where a bidirectional replication configuration does not exist, you must ensure that enough memory and disk space are available to support two copies of the database where the upgrade is being performed. Both the original database and its copy remain active for the duration of the upgrade. To maintain the performance of your production applications, you may want to create the copy of the database on a second system.

Additional disk space must be allocated to hold a backup copy of the database made by the `ttMigrate` utility. The size of the backup copy is typically about the same as the

in-use size of the database. This size may be determined by querying the `sys.monitor` table, using `ttIsql` as follows.

```
Command> SELECT perm_in_use_size FROM sys.monitor;
```

Upgrade steps

The following timeline illustrates the steps for performing an online upgrade while replication is running. The *upgrade* system is the system on which the database upgrade is being performed, and the *active* system is the system containing the database to which the application remains connected.

> **Note:** The following steps are for a standard upgrade. Upgrading from a database in TimesTen 11.2.1 that has the connection attribute `ReplicationApplyOrdering` set to `0`, or from a database in TimesTen 11.2.1 or TimesTen 11.2.2 that has `ReplicationParallelism` set to `<2`, requires that you re-create the database, even if the releases are from the same major release line.

Step	Upgrade system	Active system
1.	Configure replication to replicate to the active system using static ports.	Configure replication to replicate to the upgrade system using static ports.
2.		Connect all applications to the active database, if they are not connected.
3.	Disconnect all applications from the upgrade database.	
4.		Set replication to the upgrade system to the `PAUSE` state.
5.	Wait for updates to propagate to the active system.	
6.	Stop replication.	
7.	Back up the database with `ttMigrate -c`.	
8.	Stop the TimesTen daemon for the old release of TimesTen.	
9.	Install the new release of TimesTen.	
10.	Create a DSN for the post-upgrade database for the new TimesTen release, using the ODBC Data Source Administrator on Windows or the `.odbc.ini` file on UNIX. Adjust parallelism options for the DSN.	
11.	Restore the database from the backup with `ttMigrate -r`.	
12.	Clear the replication bookmark and logs using `ttRepAdmin -receiver -reset` and by setting replication to the active system to the `stop` and then the `start` state.	
13.	Start replication.	

Step	Upgrade system	Active system
14.		Set replication to the upgrade system to the start state, ensuring that the accumulated updates propagate once replication is restarted.
15.		Start replication.
16.		Wait for all of the updates to propagate to the upgrade system.
17.	Reconnect all applications to the post-upgrade database.	

After the above procedures are carried out on the upgrade system, the active system can be upgraded using the same steps.

Online upgrade example

This section describes how to perform an online upgrade in a scenario with two bidirectionally replicated TimesTen databases, using a specific example.

> **Note:** TimesTen 11.2.1 introduced the cache grid feature. By default, this feature is enabled, and requires additional configuration before cache groups may be created on a database. If you are upgrading an environment containing cache groups from a release preceding 11.2.1, you should set the CacheGridEnable attribute to 0 in each DSN definition before beginning. For more information, see "Create a cache grid" in *Oracle TimesTen Application-Tier Database Cache User's Guide*.

In the following discussion, the two TimesTen systems are referred to as the *upgrade* system, on which the TimesTen instance (with its databases) is being upgraded, and the *active* system, which remains operational and connected to the application for the duration of the upgrade. After the procedure is completed, the same steps can be followed to upgrade the active system. However, you may prefer to delay conversion of the active system to first test the upgraded TimesTen instance.

The upgrade system in this example consists of the database upgrade on the server upgradehost. The active system consists of the database active on the server activehost.

Follow the steps listed here in the order they are presented. The online upgrade procedures are as follows.

Step	Upgrade system	Active system
1.	Use ttIsql to alter the replication scheme repscheme, setting static replication port numbers so that the databases can communicate across releases:	Use ttIsql to alter the replication scheme repscheme, setting static replication port numbers so that the databases can communicate across releases:
	Command> call ttRepStop;	Command> call ttRepStop;
	Command> ALTER REPLICATION repscheme ALTER STORE upgrade ON upgradehost SET PORT 40000 ALTER STORE active ON activehost SET PORT 40001;	Command> ALTER REPLICATION repscheme ALTER STORE upgrade ON upgradehost SET PORT 40000 ALTER STORE active ON activehost SET PORT 40001;
	Command> call ttRepStart;	Command> call ttRepStart;

Step	Upgrade system	Active system
2.	Disconnect all production applications connected to the database. Any workload being run on the upgrade system must start running on the active system instead.	Use the ttRepAdmin utility to pause replication from the database active to the database upgrade: `ttRepAdmin -receiver -name upgrade -state pause active` This command temporarily stops the replication of updates from the database active to the database upgrade, but it retains any updates made to active in the database transaction log files. The updates made to active during the upgrade procedure are applied later, when upgrade is brought back up. For details on setting the replication state, see "Setting the replication state of subscribers" in *Oracle TimesTen In-Memory Database Replication Guide*.
3.	Wait for all replication updates to be sent to the database active. You can verify that all updates have been sent by applying a recognizable update to a table reserved for that purpose on the database upgrade. When the update appears in the database active, you know that all previous updates have been sent.	
4.	Stop the replication agent with ttAdmin: `ttAdmin -repStop upgrade` From this point on, no updates are sent to the database active.	Stop the replication agent with ttAdmin: `ttAdmin -repStop active` From this point on, no updates are sent to the database upgrade. For details on starting and stopping replication agents, see "Starting and stopping the replication agents" in *Oracle TimesTen In-Memory Database Replication Guide*.
5.	Use ttRepAdmin to stop replication from the database upgrade to the database active: `ttRepAdmin -receiver -name active -state stop upgrade` This step prevents active from accumulating updates to send to upgrade and resets some replication bookmarks.	
6.	Use ttMigrate to back up the database upgrade. If the database is very large, this step could take a significant amount of time. If sufficient disk space is free on the /backup file system, use the following ttMigrate command: `ttMigrate -c upgrade /backup/upgrade.dat`	

Step	Upgrade system	Active system

7.	If the `ttMigrate` command is successful, destroy the database `upgrade`.
	
	To destroy a permanent database (`Temporary=0`), use `ttDestroy`:
	
	`ttDestroy upgrade`
	
	To destroy a temporary database (`Temporary=1`), use `ttAdmin`:
	
	`ttAdmin -ramUnload upgrade`
		Restart the replication agent on the database `active`:
		
		`ttAdmin -repStart active`

8.	Install the new release of TimesTen.	Resume replication from `active` to `upgrade` by setting the replication state to `start`:
		
		`ttRepAdmin -receiver -name upgrade`
		` -start start active`

9.	Use `ttMigrate` to load the backup created in step 6. into the database `upgrade` for the new release of TimesTen:
	
	`ttMigrate -r upgrade`
	`/backup/upgrade.dat`
	
	If the database is temporary (`Temporary=1`), first use `ttAdmin -ramLoad`:
	
	`ttAdmin -ramLoad upgrade`
	
	Note: In this step, you must use the `ttMigrate` utility supplied with the new release of TimesTen to which you are upgrading.	

10.	Use `ttRepAdmin` to clear the replication bookmark and logs by resetting the receiver state for the database `active` and then setting replication to the `stop` state and then the `start` state:
	
	`ttRepAdmin -receiver -name active`
	`    -reset upgrade`
	`ttRepAdmin -receiver -name active`
	`    -state stop upgrade`
	`sleep 10`
	`ttRepAdmin -receiver -name active`
	`    -state start upgrade`
	`sleep 10`
	
	Note: The `sleep` command is to ensure that each state takes effect, as the state change can take up to 10 seconds depending on the resources and operating system of the computer.	

11.	Use `ttAdmin` to start the replication agent on the new database `upgrade` and to begin sending updates to the database `active`: `ttAdmin -repStart upgrade`	

Step	Upgrade system	Active system
12.	Verify that the database `upgrade` is receiving updates from the database `active`. You can verify that updates are sent by applying a recognizable update to a table reserved for that purpose in the database `active`. When the update appears in `upgrade`, you know that replication is operational.	If the applications are still running on the database `active`, let them continue until the database `upgrade` has been successfully migrated and you have verified that the updates are being replicated correctly from `active` to `upgrade`.
13.		Once you are sure that updates are replicated correctly, you can disconnect all of the applications from the database `active` and reconnect them to the database `upgrade`. After verifying that the last of the updates from `active` are replicated to `upgrade`, the TimesTen instance with `active` is ready to be upgraded. **Note**: You may choose to delay upgrading the TimesTen instance with `active` to the new TimesTen release until sufficient testing has been performed with the database `upgrade` in the new TimesTen release.

Performing an upgrade with active standby pair replication

Active standby pair replication provides high availability of your data to your applications. With active standby pairs, unless you want to perform an upgrade to a new major release of TimesTen in a configuration that also uses asynchronous writethrough cache groups, you can perform an online upgrade to maintain continuous availability of your data during an upgrade of TimesTen. This section describes the following procedures:

- Online upgrades for an active standby pair with no cache groups

- Online upgrades for an active standby pair with cache groups

- Offline upgrades for an active standby pair with cache groups

> **Note:** Only asynchronous writethrough or read-only cache groups are supported with active standby pairs.

Online upgrades for an active standby pair with no cache groups

This section includes the following topics for online upgrades in a scenario with active standby pairs and no cache groups:

- Online minor upgrade for standby master and subscriber

- Online minor upgrade for active master

- Online major upgrade for active standby pair

Also see "Performing an online upgrade with replication" on page 3-14 for an overview, limitations, and requirements.

Online minor upgrade for standby master and subscriber

To perform an online minor upgrade to a new TimesTen patch release for the standby master database and subscriber databases, complete the following tasks on each database. For this procedure, assume there are no cache groups.

1. Stop the replication agent on the database using the `ttRepStop` built-in procedure or the `ttAdmin` utility. For example, to stop the replication agent for standby database `master2`, you may use:

    ```
    ttAdmin -repStop master2
    ```

2. Install the new TimesTen patch release. See "Performing an in-place upgrade" on page 3-9.

3. Restart the replication agent using the `ttRepStart` built-in procedure or the `ttAdmin` utility:

    ```
    ttAdmin -repStart master2
    ```

Online minor upgrade for active master

To perform an online minor upgrade to a new TimesTen patch release for the active master database, you must first reverse the roles of the active and standby master databases, then perform an in-place upgrade. For this procedure, assume there are no cache groups.

1. Pause any applications that are generating updates on the active master database.

2. Execute the `ttRepSubscriberWait` built-in procedure on the active master database, using the DSN and host of the standby master database. For example, to ensure that all transactions are replicated to the standby master `master2` on the host `master2host`:

    ```
    call ttRepSubscriberWait( null, null, 'master2', 'master2host', 120 );
    ```

3. Stop the replication agent on the current active master database using the `ttRepStop` built-in procedure or the `ttAdmin` utility. For example, to stop the replication agent for the active master database `master1`:

    ```
    ttAdmin -repStop master1
    ```

4. Execute the `ttRepDeactivate` build-in procedure on the current active master database. This puts the database in the `IDLE` state:

    ```
    call ttRepDeactivate;
    ```

5. On the standby master database, set the database to the `ACTIVE` state using the `ttRepStateSet` built-in procedure. This database becomes the active master in the active standby pair:

    ```
    call ttRepStateSet( 'ACTIVE' );
    ```

6. Resume any applications that were paused in step 1, connecting them to the database that is now acting as the active master (in this example, database `master2`).

    > **Note:** At this point, replication will not yet occur from the new active database to subscriber databases. Replication will resume after the host for the new standby database has been upgraded and the replication agent of the new standby database is running.

7. Upgrade the TimesTen instance of the former active master database, which is now the standby master database. See "Performing an in-place upgrade" on page 3-9.

8. Restart replication on the database in the upgraded TimesTen instance, using the `ttRepStart` built-in procedure or the `ttAdmin` utility:

```
ttAdmin -repStart master2
```

9. To make the database in the newly upgraded TimesTen instance the active master database again, see "Reversing the roles of the active and standby databases" in *Oracle TimesTen In-Memory Database Replication Guide*.

Online major upgrade for active standby pair

When you perform an online upgrade for an active standby pair to a new major release of TimesTen, you must explicitly specify the TCP/IP port for each database. If your active standby pair replication scheme is not configured with a `PORT` attribute for each database, you must use the following steps to prepare for the upgrade. For this procedure, assume there are no cache groups. (Online major upgrades for active standby pairs with cache groups are only supported for read-only cache groups.)

1. Stop the replication agent on every database using the call `ttRepStop` built-in procedure or the `ttAdmin` utility. For example, to stop the replication agent on the database `master1`:

```
ttAdmin -repStop master1
```

2. On the active master database, use the `ALTER ACTIVE STANDBY PAIR` statement to specify a `PORT` attribute for every database in the active standby pair. For example, to set a `PORT` attribute for the databases `master1` on the host `master1host`, `master2` on the host `master2host`, and `subscriber1` on the host `subscriber1host`:

```
ALTER ACTIVE STANDBY PAIR
 ALTER STORE master1 ON "master1host" SET PORT 30000
 ALTER STORE master2 ON "master2host" SET PORT 30001
 ALTER STORE subscriber1 ON "subscriber1host" SET PORT 30002;
```

3. Destroy the standby master database and all of the subscribers using the `ttDestroy` utility. For example, to destroy the database `subscriber1`:

```
ttDestroy subscriber1
```

4. Follow the normal procedure to start an active standby pair and duplicate the standby and subscriber databases from the active master. See "Setting up an active standby pair with no cache groups" in *Oracle TimesTen In-Memory Database Replication Guide*.

To upgrade the TimesTen instances of the active standby pair, first upgrade the instance of the standby master. While this node is being upgraded, there is no standby master database, so updates on the active master database are propagated directly to the subscriber databases. Following the upgrade of the standby node, the active and standby roles are switched and the new standby node is created from the new active node. Finally, the subscriber nodes are upgraded.

1. Instruct the active master database to stop replicating updates to the standby master by executing the `ttRepStateSave` built-in procedure on the active master database. For example, to stop replication to the standby master database `master2` on host `master2host`:

```
call ttRepStateSave( 'FAILED', 'master2', 'master2host' );
```

2. Stop the replication agent on the standby master database using the `ttRepStop` built-in procedure or the `ttAdmin` utility. The following example stops the replication agent for the standby master database `master2`.

```
ttAdmin -repStop master2
```

3. Use the `ttMigrate` utility to back up the standby master database to a binary file.

```
ttMigrate -c master2 master2.bak
```

For information about `ttMigrate`, refer to *Oracle TimesTen In-Memory Database Reference*

4. Destroy the standby master database, using the `ttDestroy` utility.

```
ttDestroy master2
```

5. Uninstall the old release and install the new release of TimesTen on the standby master host, `master2host`. See "Installing TimesTen on UNIX systems" on page 1-18 or "Installing TimesTen on Windows systems" on page 1-23.

6. In the new TimesTen instance on `master2host`, use `ttMigrate` to restore the standby master database from the binary file created earlier. (This example performs a checkpoint operation after every 20 megabytes of data has been restored.)

```
ttMigrate -r -C 20 master2 master2.bak
```

7. Start the replication agent on the standby master database using the `ttRepStart` built-in procedure or the `ttAdmin` utility.

```
ttAdmin -repStart master2
```

When the standby master database in the upgraded TimesTen instance has become synchronized with the active master database, this standby master database moves from the `RECOVERING` state to the `STANDBY` state. The standby master database also starts sending updates to the subscribers. You can determine when the standby master database is in the `STANDBY` state by calling the `ttRepStateGet` built-in procedure.

```
call ttRepStateGet;
```

8. Pause any applications that are generating updates on the active master database.

9. Execute the `ttRepSubscriberWait` built-in procedure on the active master database, using the DSN and host of the standby master database. For example, to ensure that all transactions are replicated to the standby master `master2` on the host `master2host`:

```
call ttRepSubscriberWait( null, null, 'master2', 'master2host', 120 );
```

10. Stop the replication agent on the active master database using the `ttRepStop` built-in procedure or the `ttAdmin` utility. For example, to stop the replication agent for the active master database `master1`:

```
ttAdmin -repStop master1
```

11. On the standby master database, set the database to the `ACTIVE` state using the `ttRepStateSet` built-in procedure. This database becomes the active master in the active standby pair.

```
call ttRepStateSet( 'ACTIVE' );
```

12. Instruct the new active master database (`master2`, in our example) to stop replicating updates to what is now the standby master (`master1`) by executing the `ttRepStateSave` built-in procedure on the active master database. For example, to stop replication to the standby master database `master1` on host `master1host`:

```
call ttRepStateSave( 'FAILED', 'master1', 'master1host' );
```

13. Destroy the former active master database, using the `ttDestroy` utility.

```
ttDestroy master1
```

14. Uninstall the old release and install the new release of TimesTen on `master1host`.

15. If you are upgrading from a TimesTen release earlier than 11.2.1, when access control was introduced, you must create a user with `ADMIN` privileges on the new active master database to duplicate that database. (If you are upgrading from 11.2.1 or higher, you can presumably use an `ADMIN` user that was migrated over by `ttMigrate`.) For example, to create the user `pat` with the password `patpwd` on the standby master database:

```
CREATE USER pat IDENTIFIED BY patpwd;
GRANT ADMIN TO pat;
```

16. Create a new standby master database by duplicating the new active master database, using the `ttRepAdmin` utility. For example, to duplicate the database `master2` on the host `master2host` to the database `master1`, use the following on the host containing the database `master1`:

```
ttRepAdmin -duplicate -from master2 -host master2host -uid pat -pwd patpwd
 -setMasterRepStart master1
```

17. Start the replication agent on the new standby master database using the `ttRepStart` built-in procedure or the `ttAdmin` utility.

```
ttAdmin -repStart master1
```

18. Stop the replication agent on the first subscriber database using the `ttRepStop` built-in procedure or the `ttAdmin` utility. For example, to stop the replication agent for the subscriber database `subscriber1`:

```
ttAdmin -repStop subscriber1
```

19. Destroy the subscriber database using the `ttDestroy` utility.

```
ttDestroy subscriber1
```

20. Uninstall the old release and install the new release of TimesTen on the subscriber host.

21. Create the subscriber database by duplicating the new standby master database, using the `ttRepAdmin` utility, as follows.

```
ttRepAdmin -duplicate -from master1 -host master1host -uid pat -pwd patpwd
 -setMasterRepStart subscriber1
```

22. Start the replication agent for the duplicated subscriber database using the `ttRepStart` built-in procedure or the `ttAdmin` utility.

```
ttAdmin -repStart subscriber1
```

23. Repeat step 18 through step 22 for each other subscriber database.

Online upgrades for an active standby pair with cache groups

This section includes the following topics for online minor upgrades in a scenario with active standby pairs and cache groups:

- Online minor upgrade for standby master and subscriber (cache groups)
- Online minor upgrade for active master (cache groups)
- Online major upgrade for active standby pair (read-only cache groups)

Also see "Performing an online upgrade with replication" on page 3-14 for an overview, limitations, and requirements.

Online minor upgrade for standby master and subscriber (cache groups)

To perform an online minor upgrade to a new TimesTen patch release for the standby master database and subscriber databases, in a configuration with cache groups, complete the following tasks on each database (with exceptions noted).

1. Stop the replication agent on the database using the `ttRepStop` built-in procedure or the `ttAdmin` utility. For example, to stop the replication agent for standby database `master2`:

   ```
   ttAdmin -repStop master2
   ```

2. Stop the cache agent on the standby database using the `ttCacheStop` built-in procedure or the `ttAdmin` utility:

   ```
   ttAdmin -cacheStop master2
   ```

3. Install the TimesTen patch. See "Performing an in-place upgrade" on page 3-9.

4. Restart the cache agent on the standby database using the `ttCacheStart` built-in procedure or the `ttAdmin` utility:

   ```
   ttAdmin -cacheStart master2
   ```

5. Restart the replication agent using the `ttRepStart` built-in procedure or the `ttAdmin` utility:

   ```
   ttAdmin -repStart master2
   ```

   > **Note:** Steps 2 and 4, stopping and restarting the cache agent, are not applicable for subscriber databases.

Online minor upgrade for active master (cache groups)

To perform an online minor upgrade to a new TimesTen patch release for the active master database, in a configuration with cache groups, perform the following steps. You must first reverse the roles of the active and standby master databases, then perform an in-place upgrade.

1. Pause any applications that are generating updates on the active master database.

2. Execute the `ttRepSubscriberWait` built-in procedure on the active master database, using the DSN and host of the standby master database. For example, to ensure that all transactions are replicated to the standby master `master2` on the host `master2host`:

   ```
   call ttRepSubscriberWait( null, null, 'master2', 'master2host', 120 );
   ```

3. Stop the replication agent on the current active master database using the `ttRepStop` built-in procedure or the `ttAdmin` utility. For example, to stop the replication agent for the active master database `master1`:

```
ttAdmin -repStop master1
```

4. Stop the cache agent on the current active master database using the `ttCacheStop` built-in procedure or the `ttAdmin` utility:

```
ttAdmin -cacheStop master1
```

5. Execute the `ttRepDeactivate` build-in procedure on the current active master database. This puts the database in the `IDLE` state:

```
call ttRepDeactivate;
```

6. On the standby master database, set the database to the `ACTIVE` state using the `ttRepStateSet` built-in procedure. This database becomes the active master in the active standby pair:

```
call ttRepStateSet( 'ACTIVE' );
```

7. Resume any applications that were paused in step 1, connecting them to the database that is now acting as the active master (in this example, database `master2`).

8. Upgrade the TimesTen instance for the former active master database, which is now the standby master database. See "Performing an in-place upgrade" on page 3-9.

9. Restart the cache agent on the post-upgrade database using the `ttCacheStart` built-in procedure or the `ttAdmin` utility:

```
ttAdmin -cacheStart master1
```

10. Restart replication on the post-upgrade database using the `ttRepStart` built-in procedure or the `ttAdmin` utility:

```
ttAdmin -repStart master1
```

11. To make the post-upgrade database the active master database again, see "Reversing the roles of the active and standby databases" in *Oracle TimesTen In-Memory Database Replication Guide*.

Online major upgrade for active standby pair (read-only cache groups)

Complete the following steps to perform a TimesTen major upgrade from an 11.2.1.*x.x* release to an 11.2.2.*x.x* release in a scenario with an active standby pair with read-only cache groups.

These steps assume an active master database `master1` on host `master1host` and a standby master database `master2` on host `master2host`.

> **Note:** For more information on the built-in procedures and utilities discussed here, see "Built-in Procedures" and "Utilities" in the *Oracle TimesTen In-Memory Database Reference*.

1. On the active master host, run the `ttAdmin` utility to stop the replication agent for the active master database.

```
ttAdmin -repStop master1
```

2. On the active master database, use the DROP ACTIVE STANDBY PAIR statement to drop the active standby pair. For example, from the ttIsql utility:

```
Command> DROP ACTIVE STANDBY PAIR;
```

3. On the active master database, use the CREATE ACTIVE STANDBY PAIR statement to create a new active standby pair with the cache groups excluded. Ensure that you explicitly specify the TCP/IP port for each database.

```
Command> CREATE ACTIVE STANDBY PAIR master1 ON "master1host",
          master2 ON "master2host"
       STORE master1 ON "master1host" PORT 20000
       STORE master2 ON "master2host" PORT 20010
       EXCLUDE CACHE GROUP cacheuser.readcache;
```

> **Note:** You can use the cachegroups command within the ttIsql utility to identify all the cache groups defined in the database. In this example, readcache is a read-only cache group owned by the cacheuser user.

4. On the active master database, call the ttRepStateSet built-in procedure to set the replication state for the active master database to ACTIVE.

```
Command> call ttRepStateSet('ACTIVE');
```

To verify that the replication state for the active master database is set to ACTIVE, call the ttRepStateGet built-in procedure.

```
Command> call ttRepStateGet();
< ACTIVE, NO GRID >
1 row found.
```

5. On the active master database, call the ttRepStart built-in procedure to start the replication agent.

```
Command> call ttRepStart();
```

6. On the standby master host, run the ttAdmin utility to stop the replication agent for the standby master database.

```
ttAdmin -repStop master2
```

7. On the standby master host, run the ttAdmin utility to stop the cache agent for the standby master database.

```
ttAdmin -cacheStop master2
```

8. On the standby master host, run the ttDestroy utility to destroy the standby master database. You must either add the -force option or first drop all cache groups.

```
ttDestroy -force master2
```

9. Create a new standby master database by duplicating the active master database with the ttRepAdmin utility. For example, to duplicate the master1 database on the master1host host of the master2 database, run the following on the host containing the master2 database.

```
ttRepAdmin -duplicate -from master1 -host master1host -UID pat -PWD patpwd
  -keepCG -cacheUid cacheuser -cachePwd cachepwd master2
```

> **Note:** You need a user with `ADMIN` privileges defined in the active master database for it to be duplicated. In this example, the `pat` user identified by the `patpwd` password has `ADMIN` privileges.
>
> To keep the cache group tables, you need a cache administration user while adding the `-keepCG` option. In this example, the `cacheuser` user identified by the `cachepwd` password is a cache administration user.

10. On the new standby master database, use the `DROP CACHE GROUP` statement to drop all the cache groups.

    ```
    Command> DROP CACHE GROUP cacheuser.readcache;
    ```

11. On the standby master host, run the `ttMigrate` utility to back up the standby master database to a binary file.

    ```
    ttMigrate -c master2 master2.bak
    ```

12. On the standby master host, run the `ttDestroy` utility to destroy the standby master database.

    ```
    ttDestroy master2
    ```

13. Uninstall the old release and install the new release of TimesTen on the standby master host. See "Installing TimesTen on UNIX systems" on page 1-18 or "Installing TimesTen on Windows systems" on page 1-23.

14. In the new TimesTen instance on the standby master host, run the `ttMigrate` utility to restore the standby master database from the binary file created earlier.

    ```
    ttMigrate -r -C 20 master2 master2.bak
    ```

    > **Note:** This example performs a checkpoint operation after every 20 MB of data has been restored.

15. On the standby master database, use the `CREATE USER` statement to create a new cache administration user.

    ```
    Command> CREATE USER cacheuser2 IDENTIFIED BY cachepwd;
    Command> GRANT CREATE SESSION, CACHE_MANAGER, CREATE ANY TABLE, DROP ANY TABLE
            TO cacheuser2;
    ```

    > **Note:** You must create the new cache administration user in the Oracle database and grant the user the minimum set of privileges required to perform cache grid and cache group operations. See "Create users in the Oracle database" in the *Oracle TimesTen Application-Tier Database Cache User's Guide*.

16. Connect to the standby master database as the cache administration user, and call the `ttCacheUidPwdSet` built-in procedure to set the new cache administration user name and password. Ensure you specify the cache administration user password for the Oracle database in the `OraclePWD` connection attribute within the connection string.

```
ttIsql "DSN=master2;UID=cacheuser2;PWD=cachepwd;OraclePWD=oracle"
Command> call ttCacheUidPwdSet('cacheuser2','oracle');
```

17. On the standby master database, call the `ttCacheStart` built-in procedure to start the cache agent.

```
Command> call ttCacheStart();
```

18. On the standby master database, call the `ttRepStart` built-in procedure to start the replication agent.

```
Command> call ttRepStart();
```

The replication state will automatically be set to `STANDBY`. You can call the `ttRepStateGet` built-in procedure to confirm this. (This occurs asynchronously and may take a little time.)

```
Command> call ttRepStateGet();
< STANDBY, NO GRID >
1 row found.
```

19. On the standby master database, use the `CREATE READONLY CACHE GROUP` statement to create all the read-only cache groups.

```
Command> CREATE READONLY CACHE GROUP cacheuser2.readcache
         AUTOREFRESH INTERVAL 10 SECONDS
         FROM oratt.readtbl
            (keyval NUMBER NOT NULL PRIMARY KEY, str VARCHAR(32));
```

> **Note:** Ensure that the cache administration user has `SELECT` privileges on the cache group tables in the Oracle database. In this example, the `cacheuser2` user has `SELECT` privileges on the `readtbl` table owned by the `oratt` user in the Oracle database. For more information, see "Create the Oracle Database tables to be cached" in the *Oracle TimesTen Application-Tier Database Cache User's Guide*.

20. On the standby master database, use the `LOAD CACHE GROUP` statement to load the data from the Oracle database tables into the TimesTen cache groups.

```
Command> LOAD CACHE GROUP cacheuser2.readcache
         COMMIT EVERY 200 ROWS;
```

21. Pause any applications that are generating updates on the active master database.

22. On the active master database, call the `ttRepSubscriberWait` built-in procedure using the DSN and host of the standby master database. For example, to ensure that all transactions are replicated to the `master2` database on the `master2host` host:

```
Command> call ttRepSubscriberWait(NULL,NULL,'master2','master2host',120);
```

23. On the active master database, call the `ttRepStop` built-in procedure to stop the replication agent.

```
Command> call ttRepStop();
```

24. On the active master database, call the `ttRepDeactivate` built-in procedure to set the replication state for the active master database to `IDLE`.

```
Command> call ttRepDeactivate();
```

25. On the standby master database, call the `ttRepStateSet` built-in procedure to set the replication state for the standby master database to `ACTIVE`. This database and its host become the active master in the active standby pair replication scheme.

```
Command> call ttRepStateSet('ACTIVE');
```

> **Note:** In this example, the `master2` database on the `master2host` host just became the active master in the active standby pair replication scheme. Likewise, the `master1` database on the `master1host` host is henceforth considered the standby master in the active standby pair replication scheme.

26. On the new active master database, call the `ttRepStop` built-in procedure to stop the replication agent.

```
Command> call ttRepStop();
```

27. On the active master database, use the `ALTER CACHE GROUP` statement to set the `AUTOREFRESH` mode of all cache groups to `PAUSED`.

```
Command> ALTER CACHE GROUP cacheuser2.readcache
        SET AUTOREFRESH STATE PAUSED;
```

28. On the active master database, use the `DROP ACTIVE STANDBY PAIR` statement to drop the active standby pair.

```
Command> DROP ACTIVE STANDBY PAIR;
```

29. On the active master database, use the `CREATE ACTIVE STANDBY PAIR` statement to create a new active standby pair with the cache groups included. Ensure you explicitly specify the TCP/IP port for each database.

```
Command> CREATE ACTIVE STANDBY PAIR master1 ON "master1host",
        master2 ON "master2host"
       STORE master1 ON "master1host" PORT 20000
       STORE master2 ON "master2host" PORT 20010;
```

30. On the active master database, call the `ttRepStateSet` built-in procedure to set the replication state for the active master database to `ACTIVE`.

```
Command> call ttRepStateSet('ACTIVE');
```

31. On the active master database, call the `ttRepStart` built-in procedure to start the replication agent.

```
Command> call ttRepStart();
```

32. Resume any applications that were paused in step 21, connecting them to the new active master database.

33. On the new standby master host, run the `ttDestroy` utility to destroy the new standby master database.

```
ttDestroy master1
```

34. Uninstall the old release and install the new release of TimesTen on the standby master host.

35. Create a new standby master database by duplicating the active master database with the `ttRepAdmin` utility. For example, to duplicate the `master2` database on the

master2host host to the master1 database, run the following on the host
containing the master1 database:

```
ttRepAdmin -duplicate -from master2 -host master2host -UID pat -PWD patpwd
  -keepCG -cacheUid cacheuser2 -cachePwd cachepwd master1
```

36. On the standby master host, run the ttAdmin utility to start the cache agent for the
standby master database.

```
ttAdmin -cacheStart master1
```

37. On the standby master host, run the ttAdmin utility to start the cache agent for the
standby master database.

```
ttAdmin -repStart master1
```

Offline upgrades for an active standby pair with cache groups

Performing a TimesTen major upgrade in a scenario with an active standby pair with
asynchronous writethrough cache groups requires an offline upgrade. This is
discussed in the subsection that follows.

Offline major upgrade for active standby pair (cache groups)

Complete the following steps to perform a TimesTen major upgrade from an 11.2.1.*x.x*
release to an 11.2.2.*x.x* release in a scenario with an active standby pair with cache
groups. You must perform this upgrade offline.

> **Note:** This procedure is not supported for upgrading from a 7.0.*x* or
> earlier release of TimesTen. For that case, an alternative is to exclude
> cache groups from your replication scheme before the upgrade, use
> the procedure for upgrading without cache groups, then include the
> previously excluded cache groups into the replication scheme after the
> upgrade. For an upgrade without cache groups, see "Online major
> upgrade for active standby pair" on page 3-22.

These steps assume an active master database master1 on host master1host and a
standby master database master2 on host master2host. (For information about the
built-in procedures and utilities discussed here, refer to "Built-In Procedures" and
"Utilities" in *Oracle TimesTen In-Memory Database Reference*.)

1. Stop any updates to the active database before you upgrade.

2. From master1, call the ttRepSubscriberWait built-in procedure to ensure that all
data updates have been applied to the standby database, where *numsec* is the
desired wait time.

```
call ttRepSubscriberWait(null, null, 'master2', 'master2host', numsec);
```

3. From master2, call ttRepSubscriberWait to ensure that all data updates have
been applied to the Oracle database.

```
call ttRepSubscriberWait(null, null, '_ORACLE', null, numsec);
```

4. On master1host, use the ttAdmin utility to stop the replication agent for the active
database.

```
ttAdmin -repStop master1
```

5. On `master2host`, use `ttAdmin` to stop the replication agent for the standby database.

```
ttAdmin -repStop master2
```

6. On `master1host`, call the `ttCacheStop` built-in procedure or use `ttAdmin` to stop the cache agent for the active database.

```
ttAdmin -cacheStop master1
```

7. On `master2host`, call `ttCacheStop` or use `ttAdmin` to stop the cache agent for the standby database.

```
ttAdmin -cacheStop master2
```

8. On `master1host`, use the `ttMigrate` utility to back up the active database to a binary file.

```
ttMigrate -c master1 master1.bak
```

9. On `master1host`, use the `ttDestroy` utility to destroy the active database. You must either use the `-force` option or first drop all cache groups. If you use `-force`, run the script `cacheCleanup.sql` afterward.

```
ttDestroy -force /data_store_path/master1
```

The `cacheCleanup.sql` script is a SQL*Plus script, located in the *TimesTen_install_dir*/oraclescripts directory, that you run after connecting to the Oracle database as the cache user. It takes as parameters the TimesTen host name and the TimesTen database name (with full path). For information, refer to "Dropping Oracle Database objects used by autorefresh cache groups" in *Oracle TimesTen Application-Tier Database Cache User's Guide*.

10. Uninstall the old and install the new TimesTen major release on `master1host`. For details, see "Installing TimesTen on UNIX systems" on page 1-18 or "Installing TimesTen on Windows systems" on page 1-23.

11. Create a new database in 11.2.2.*x.x* using `ttIsql` with DSN connection attribute setting `AutoCreate=1`. In this new database, create a cache user. The following example is a sequence of commands to execute in `ttIsql` to create this cache user and give it appropriate access privileges.

    The cache user requires `ADMIN` privilege to execute the next step, `ttMigrate -r`. Once migration is complete, you can revoke the `ADMIN` privilege from this user.

```
Command> CREATE USER cacheuser IDENTIFIED BY cachepassword;
Command> GRANT CREATE SESSION, CACHE_MANAGER, CREATE ANY TABLE, DROP ANY TABLE
 TO cacheuser;
Command> GRANT ADMIN TO cacheuser;
```

12. In the new TimesTen instance on `master1host`, use the `ttMigrate` utility as the cache user to restore `master1` from the binary file created earlier. (This example performs a checkpoint operation after every 20 MB of data has been restored, and assumes the password is the same in the Oracle database as in TimesTen.)

```
ttMigrate -r -cacheuid cacheuser -cachepwd cachepassword -C 20 -connstr
 "DSN=master1;uid=cacheuser;pwd=cachepassword;oraclepwd=cachepassword"
 master1.bak
```

13. On `master1host`, use `ttAdmin` to start the replication agent.

```
ttAdmin -repStart master1
```

> **Note:** This step also sets the database to the active state. You can then call the `ttRepStateGet` built-in procedure (which takes no parameters) to confirm the state.

14. On `master1host`, call the `ttCacheStart` built-in procedure or use `ttAdmin` to start the cache agent.

```
ttAdmin -cacheStart master1
```

Then you can use the `ttStatus` utility to confirm the replication and cache agents have started.

15. Put each automatic refresh cache group into the `AUTOREFRESH PAUSED` state. This example uses `ttIsql`:

```
Command> ALTER CACHE GROUP mycachegroup SET AUTOREFRESH STATE paused;
```

16. From `master1`, reload each cache group, specifying the name of the cache group and how often to commit during the operation. This example uses `ttIsql`:

```
Command> LOAD CACHE GROUP cachegroupname COMMIT EVERY n ROWS;
```

You can optionally specify parallel loading as well. For information about the `LOAD CACHE GROUP` SQL statement, refer to *Oracle TimesTen In-Memory Database SQL Reference*.

17. On `master2host`, use `ttDestroy` to destroy the standby database. You must either use the `-force` option or first drop all cache groups. If you use `-force`, run the script `cacheCleanup.sql` afterward (as discussed earlier).

```
ttDestroy -force /data_store_path/master2
```

18. Uninstall the old and install the new TimesTen major release on `master2host`.

19. In the new TimesTen instance on `master2host`, use the `ttRepAdmin` utility with the `-duplicate` option to create a duplicate of active database `master1` to use as standby database `master2`. Specify the appropriate administrative user on `master1`, the cache manager user and password, and to keep cache groups.

```
ttRepAdmin -duplicate -from master1 -host master1host -uid pat -pwd patpwd
-cacheUid orcluser -cachePwd orclpwd -keepCG master2
```

20. On `master2host`, use `ttAdmin` to start the replication agent. (You could optionally have used the `ttRepAdmin` option `-setMasterRepStart` in the previous step instead.)

```
ttAdmin -repStart master2
```

21. On `master2`, the replication state will automatically be set to `STANDBY`. You can call the `ttRepStateGet` built-in procedure to confirm this. (This occurs asynchronously and may take a little time.)

```
call ttRepStateGet();
```

22. On `master2host`, call the `ttCacheStart` built-in procedure or use `ttAdmin` to start the cache agent.

```
ttAdmin -cacheStart master2
```

After this, you can use the `ttStatus` utility to confirm the replication and cache agents have started.

If you want to create read-only subscriber databases, on each subscriber host you can create the subscriber by using the `ttRepAdmin` utility `-duplicate` option to duplicate the standby database. The following example creates `subscriber1`, using the same `ADMIN` user as above and the `-nokeepCG` option to convert the cache tables to normal TimesTen tables, as appropriate for a read-only subscriber.

```
ttRepAdmin -duplicate -from master2 -host master2host -nokeepCG
-uid pat -pwd patpwd subscriber1
```

For related information, refer to "Rolling out a disaster recovery subscriber" in *Oracle TimesTen In-Memory Database Replication Guide*.

Performing an offline TimesTen upgrade when using Oracle Clusterware

This section discusses the steps for an offline upgrade of TimesTen when using TimesTen with Oracle Clusterware. You have the option of also upgrading Oracle Clusterware, independently, while upgrading TimesTen. (For an online upgrade, see "Performing an online TimesTen upgrade when using Oracle Clusterware" on page 3-35.)

> **Notes:**
>
> - These instructions apply for either a TimesTen minor upgrade (11.2.2.*x.x* to higher 11.2.2.*x.x*) or a TimesTen major upgrade (11.2.1.*x.x* to 11.2.2.*x.x*).
>
> - TimesTen 11.2.2 supports Clusterware releases 11.2.0.2 and 11.2.0.3 only. (TimesTen 11.2.1 supports Clusterware releases 11.1.0.7, 11.2.0.2, and 11.2.0.3.)
>
> - TimesTen does not support Clusterware on Windows platforms.

For this procedure, except where noted, you can execute the `ttCWAdmin` commands from any host in the cluster. Each command affects all hosts.

1. Stop the replication agents on the databases in the active standby pair:

   ```
   ttCWAdmin -stop -dsn advancedDSN
   ```

2. Drop the active standby pair:

   ```
   ttCWAdmin -drop -dsn advancedDSN
   ```

3. Stop the TimesTen cluster agent. This removes the hosts from the cluster and stops the TimesTen daemon:

   ```
   ttCWAdmin -shutdown
   ```

4. Upgrade TimesTen on the desired hosts.

   - To perform a TimesTen minor upgrade, each node in the cluster must have TimesTen from the same major release line (such as all 11.2.2.*x.x* releases). You can do an in-place upgrade as discussed in "Performing an in-place upgrade" on page 3-9.

   - To perform a TimesTen major upgrade, you must use `ttMigrate` as discussed in "Performing an offline upgrade" on page 3-11.

5. Upgrade Oracle Clusterware if desired, according to instructions in Oracle Database documentation, which you can access as follows.

a. Go go `http://www.oracle.com/pls/db112/homepage`.

b. Select the "Installing and Upgrading" link.

c. Refer to the *Grid Infrastructure Installation Guide* for your platform.

Also see "Overview of Upgrading Oracle Clusterware" in *Oracle Clusterware Administration and Deployment Guide*.

For use with TimesTen, you can upgrade from Clusterware release 11.1.0.7 to 11.2.0.3 or from 11.2.0.2 to 11.2.0.3, but *not* from 11.1.0.7 to 11.2.0.2.

6. If you have upgraded Oracle Clusterware, use the `ttmodinstall` utility (documented in *Oracle TimesTen In-Memory Database Reference*) to configure TimesTen with Oracle Clusterware. On each host, enter:

```
ttmodinstall -crs
```

When `ttmodinstall` asks permission to overwrite the `ttcrsagent.options` file, choose `yes` (the default).

See "Install TimesTen on each host" and "Start the TimesTen cluster agent" in *Oracle TimesTen In-Memory Database Replication Guide* for information related to `ttcrsagent.options`.

7. Start the TimesTen cluster agent. This includes the hosts defined in the cluster as specified in `ttcrsagent.options`. This also starts the TimesTen daemon.

```
ttCWAdmin -init
```

8. Create the active standby pair replication scheme:

```
ttCWAdmin -create -dsn advancedDSN
```

Important: The host from which you run this command must have access to the `cluster.oracle.ini` file. (See "Configuring Oracle Clusterware management with the cluster.oracle.ini file" in *Oracle TimesTen In-Memory Database Replication Guide* for information about this file.)

9. Start the active standby pair replication scheme:

```
ttCWAdmin -start -dsn advancedDSN
```

Performing an online TimesTen upgrade when using Oracle Clusterware

This section discusses how to perform an online rolling in-place minor upgrade for TimesTen, from one TimesTen 11.2.2.*x.x* release to a higher 11.2.2.*x.x* release, in a configuration where Oracle Clusterware manages active standby pairs. (For an offline upgrade, see "Performing an offline TimesTen upgrade when using Oracle Clusterware" on page 3-34.)

The following topics are covered:

- Supported configurations

- Restrictions and assumptions

- Upgrade tasks for one active standby pair

- Upgrades for multiple active standby pairs on many pairs of hosts

- Upgrades for multiple active standby pairs on a pair of hosts

- Sample in-place upgrade

> **Notes:**
>
> - TimesTen 11.2.2 supports Clusterware releases 11.2.0.2 and 11.2.0.3 only. (TimesTen 11.2.1 supports Clusterware releases 11.1.0.7, 11.2.0.2, and 11.2.0.3.)
>
> - TimesTen does not support Clusterware on Windows platforms.

Supported configurations

The following basic configurations are supported for online rolling in-place upgrades for TimesTen. In all cases, Oracle Clusterware manages the hosts.

- One active standby pair on two hosts.

- Multiple active standby pairs with one TimesTen database on each host.

- Multiple active standby pairs with one or more TimesTen database on each host.

(Other scenarios, such as with additional spare systems, are effectively equivalent to one of these scenarios.)

Restrictions and assumptions

Note the following assumptions for upgrading TimesTen when using Oracle Clusterware:

- The existing active standby pairs are configured and operating properly.

- Oracle Clusterware commands are used correctly to stop and start the standby database.

- The in-place upgrade does not change the TimesTen environment for the active and standby databases.

- The in-place upgrade is from a TimesTen 11.2.2.*x.x* release to a later TimesTen 11.2.2.*x.x* release. These instructions are for TimesTen minor upgrades only. Online major upgrades are not currently supported in configurations where Oracle Clusterware manages active standby pairs.

- Oracle Clusterware is release 11.2.0.2 or 11.2.0.3.

- The in-place upgrade applies to one TimesTen installation on each host.

- There are at least two hosts managed by Oracle Clusterware.

  Multiple active or standby databases managed by Oracle Clusterware can exist on a host only if there are at least two hosts in the cluster.

> **Important:**
>
> - Upgrade Oracle Clusterware if desired, but *not* concurrently with an online TimesTen upgrade. When performing an online TimesTen minor upgrade in configurations where Oracle Clusterware manages active standby pairs, you must perform the Clusterware upgrade independently and separately, either before or after the TimesTen upgrade.
>
>   Be aware that for use with TimesTen, you can upgrade from Clusterware release 11.1.0.7 to 11.2.0.3 or from 11.2.0.2 to 11.2.0.3, but *not* from 11.1.0.7 to 11.2.0.2.
>
> - For upgrading TimesTen when using Oracle Clusterware, online rolling in-place upgrades are not supported when global cache groups are replicated by the active standby pairs.

> **Note:** Refer to Oracle Database documentation for information about upgrading Oracle Clusterware:
>
> 1. Go go `http://www.oracle.com/pls/db112/homepage`.
> 2. Select the "Installing and Upgrading" link.
> 3. Refer to the *Grid Infrastructure Installation Guide* for your platform.
>
> Also see "Overview of Upgrading Oracle Clusterware" in *Oracle Clusterware Administration and Deployment Guide*.

Upgrade tasks for one active standby pair

This section describes the following tasks:

- Verify that the active standby pair is operating properly
- Shut down the standby database
- Perform an in-place upgrade for the standby database
- Start the standby database
- Switch the roles of the active and standby databases
- Shut down the new standby database
- Perform an in-place upgrade of the new standby database
- Start the new standby database

> **Note:** In examples in the following subsections, the host name is host2, the DSN is myDSN, the TimesTen instance name is upgrade2, and the instance administrator is terry.

Verify that the active standby pair is operating properly

Complete these steps to confirm that the active standby pair is operating properly.

1. Verify the following.

   - The active and the standby databases run a TimesTen 11.2.2.*x.x* release.

- The active and standby databases are on separate hosts managed by Oracle Clusterware.

- Replication is working.

- If the active standby pair replication scheme includes cache groups, the following are true:

    - AWT and SWT writes from the standby TimesTen database to the Oracle database are working.

    - Refreshes from the Oracle database to the active TimesTen database are working.

2. Enter the `ttCWAdmin -status -dsn` *yourDSN* command to verify the following.

- The active database is on a different host than the standby database.

- The state of the active database is `'ACTIVE'` and the status is `'AVAILABLE'`.

- The state of the standby database is `'STANDBY'` and the status is `'AVAILABLE'`.

3. Enter the `ttStatus` command on the active database to verify the following.

- The `ttCRSactiveservice` and `ttCRSmaster` processes are running.

- The subdaemon and the replication agents are running.

- If the active standby pair replication scheme includes cache groups, the cache agent is running.

4. Enter the `ttStatus` command on the standby database to verify the following.

- The `ttCRSsubservice` and `ttCRSmaster` processes are running.

- The subdaemon and the replication agents are running.

- If the active standby pair replication scheme includes cache groups, the cache agent is running.

Shut down the standby database

Complete these steps to shut down the standby database.

1. Enter an Oracle Clusterware command similar to the following to obtain the names of the Oracle Clusterware Master, Daemon, and Agent processes on the host of the standby database. It is suggested to filter the output through `grep TT`:

```
crsctl status resource -n standbyHostName | grep TT
```

2. Use Oracle Clusterware commands to shut down the standby database. The Oracle Clusterware commands stop the Master processes for the standby database, the Daemon process for the TimesTen installation, and the Agent process for the TimesTen installation.

```
crsctl stop resource TT_Master_upgrade2_terry_myDSN_1
crsctl stop resource TT_Daemon_upgrade2_terry_host2
crsctl stop resource TT_Agent_upgrade2_terry_host2
```

3. Stop the TimesTen main daemon.

```
ttDaemonAdmin -stop
```

If the `ttDaemonAdmin -stop` command gives error 10028, retry the command.

Perform an in-place upgrade for the standby database

Complete these steps for an in-place upgrade of the TimesTen instance for the standby database.

1. Use the `setup.sh` script from the installation media of your current TimesTen release to do the in-place upgrade. The in-place upgrade uninstalls the previous 11.2.2.*x.x* installation for the standby database. The new installation for the current release must be installed into the same directory structure.

    The in-place upgrade must keep the following files on the standby database:

    - Standby database files (such as checkpoint files, transaction log files, and so on)

    - `sys.odbc.ini`

    - `ttendaemon.options`

    - `cluster.oracle.ini`

        See "Configuring Oracle Clusterware management with the cluster.oracle.ini file" in *Oracle TimesTen In-Memory Database Replication Guide* for information about this file.

    - `ttcrsagent.options`

        See "Install TimesTen on each host" and "Start the TimesTen cluster agent" in *Oracle TimesTen In-Memory Database Replication Guide* for information related to this file.

    - `tnsnames.ora`

    See .

2. Configure the new installation for Oracle Clusterware.

Start the standby database

Complete these steps to start the standby database.

1. Enter the following `ttCWAdmin` command to start the TimesTen main daemon, the TimesTen Oracle Clusterware agent process and the TimesTen Oracle Clusterware Daemon process:

    ```
    ttCWAdmin -init -hosts localhost
    ```

2. Start the Oracle Clusterware Master process for the standby database.

    ```
    crsctl start resource TT_Master_upgrade2_terry_MYDSN_1
    ```

Switch the roles of the active and standby databases

Use the `ttCWAdmin -switch` command to switch the roles of the active and standby databases to enable the in-place upgrade on the other master database.

```
ttCWAdmin -switch -dsn myDSN
```

Use the `ttCWAdmin -status` command to verify that the switch operation has completed before starting the next task.

Shut down the new standby database

Use the Oracle Clusterware `crsctl status resource` command to obtain the names of the Master, Daemon, and Agent processes on the host of the new standby database. The following example assumes the host `host1` and filters the output through `grep TT`.

```
crsctl status resource -n host1 | grep TT
```

Enter commands such as those in "Shut down the standby database" on page 3-38. Use the appropriate instance name, instance administrator, DSN, and host name. For example:

```
crsctl stop resource TT_Master_upgrade2_terry_MYDSN_0
crsctl stop resource TT_Daemon_upgrade2_terry_host1
crsctl stop resource TT_Agent_upgrade2_terry_host1
ttDaemonAdmin -stop
```

Perform an in-place upgrade of the new standby database

Enter commands such as those in "Perform an in-place upgrade for the standby database" on page 3-39.

Start the new standby database

See "Start the standby database" on page 3-39, but use the Master process name obtained by the `crsctl status resource` command from "Shut down the new standby database", above.

```
ttCWAdmin -init -hosts localhost
crsctl start resource TT_Master_upgrade2_terry_MYDSN_0
```

Upgrades for multiple active standby pairs on many pairs of hosts

The process to upgrade the TimesTen instances for multiple active standby pairs on multiple pairs of hosts is essentially the same as the process to upgrade the instances for a single active standby pair on two hosts, discussed in the preceding section, "Upgrade tasks for one active standby pair" on page 3-37. The best practice is to perform the upgrades for the active standby pairs one at a time.

Use the `ttCWAdmin -status` command to determine the state of the databases managed by Oracle Clusterware.

Upgrades for multiple active standby pairs on a pair of hosts

Multiple active standby pairs can be on multiple pairs of hosts, as discussed in the preceding section, "Upgrades for multiple active standby pairs on many pairs of hosts". Alternatively, multiple active standby pairs can be on a single pair of hosts. One scenario is for all the active databases to be on one host and all the standby databases to be on the other. A more typical scenario, to better balance the workload, is for each host to have some active databases and some standby databases.

Figure 3–1 shows two active standby pairs on two hosts managed by Oracle Clusterware. The active database called `active1` on `host1` replicates to `standby1` on `host2`. The active database called `active2` on `host2` replicates to `standby2` on `host1`. AWT updates from both standby databases are propagated to the Oracle database. Read-only updates from the Oracle database are propagated to the active databases.

Figure 3–1 Multiple active standby pairs on two hosts
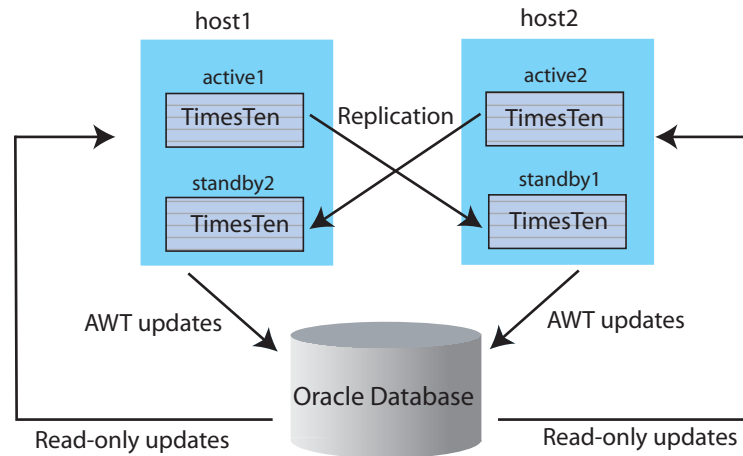
This configuration can result in greater write throughput for cache groups and more balanced resource usage. See the next section, "Sample configuration files: multiple active standby pairs on one pair of hosts", for sample `sys.odbc.ini` entries and a sample `cluster.oracle.ini` file for this kind of configuration. (See "Configuring Oracle Clusterware management with the cluster.oracle.ini file" in *Oracle TimesTen In-Memory Database Replication Guide* for information about that file.)

The in-place rolling upgrade process for multiple active standby pairs on a single pair of hosts is similar in nature to the process of upgrading multiple active standby pairs on multiple pairs of hosts, discussed in "Upgrades for multiple active standby pairs on many pairs of hosts" on page 3-40.

First, however, if the active and standby databases are mixed between the two hosts, switch all standby databases to one host and all active databases to the other host. Use the `ttCWAdmin -switch -dsn` *DSN* command to switch active and standby databases between hosts. Once all the active databases are on one host and all the standby databases are on the other host, follow the steps below to perform the upgrade for the entire "standby" host.

Be aware that in-place upgrades affect the entire TimesTen installation and associated databases on one host.

1. Verify that the standby databases run on the desired host. Use the `ttCWAdmin -status -dsn` *DSN* command and the `ttCWAdmin -status` command.

2. Modify the Oracle Clusterware `stop` commands to stop all Master processes on the host where all the standby databases reside.

3. Modify the Oracle Clusterware `start` commands to start all Master processes on the host where all the standby databases reside.

The following subsections contain related samples.

■   Sample configuration files: multiple active standby pairs on one pair of hosts

■   Sample scripts: stopping and starting multiple standby processes on one host

Sample configuration files: multiple active standby pairs on one pair of hosts

The following are sample `sys.odbc.ini` entries:

```
[databasea]
Driver=/scratch/terry/TimesTen/upgrade2/lib/libtten.so
```

```
DataStore=/scratch/terry/ds/databasea
PermSize=40
TempSize=32
PLSQL=1
DatabaseCharacterSet=WE8MSWIN1252
OracleNetServiceName=ORCL

[databaseb]
Driver=/scratch/terry/TimesTen/upgrade2/lib/libtten.so
DataStore=/scratch/terry/ds/databaseb
PermSize=40
TempSize=32
PLSQL=1
DatabaseCharacterSet=WE8MSWIN1252
OracleNetServiceName=ORCL

[databasec]
Driver=/scratch/terry/TimesTen/upgrade2/lib/libtten.so
DataStore=/scratch/terry/ds/databasec
PermSize=40
TempSize=32
PLSQL=1
DatabaseCharacterSet=WE8MSWIN1252
OracleNetServiceName=ORCL

[databased]
Driver=/scratch/terry/TimesTen/upgrade2/lib/libtten.so
DataStore=/scratch/terry/ds/databased
PermSize=40
TempSize=32
PLSQL=1
DatabaseCharacterSet=WE8MSWIN1252
OracleNetServiceName=ORCL
```

The following is a sample `cluster.oracle.ini` file:

```
[databasea]
MasterHosts=host1,host2
CacheConnect=Y

[databaseb]
MasterHosts=host2,host1
CacheConnect=Y

[databasec]
MasterHosts=host2,host1
CacheConnect=Y

[databased]
MasterHosts=host1,host2
CacheConnect=Y
```

The `cluster.oracle.ini` file places one active database and one standby database on each host. This is accomplished by reversing the order of the host names specified for the `MasterHost` attribute.

Sample scripts: stopping and starting multiple standby processes on one host

Enter an Oracle Clusterware command similar to the following to obtain the names of the Oracle Clusterware Master, Daemon and Agent processes on the host of the standby database. It is suggested to filter the output through `grep TT`:

```
crsctl status resource -n standbyHostName | grep TT
```

The following script is an example of a "stop standby" script for multiple databases on the same host that Oracle Clusterware manages. The TimesTen instance name is `upgrade2`. The instance administrator is `terry`. The host is `host2`. There are two standby databases: `databasea` and `databaseb`.

```
crsctl stop resource TT_Master_upgrade2_terry_DATABASEA_0
crsctl stop resource TT_Master_upgrade2_terry_DATABASEB_1
crsctl stop resource TT_Daemon_upgrade2_terry_HOST2
crsctl stop resource TT_Agent_upgrade2_terry_HOST2
ttDaemonAdmin -stop
```

The following script is an example of a "start standby" script for the same configuration.

```
ttCWAdmin -init -hosts localhost
crs start resource TT_Master_upgrade2_terry_DATABASEA_0
crs start resource TT_Master_upgrade2_terry_DATABASEB_1
```

Sample in-place upgrade

This section provides a sample in-place upgrade.

```
% ./setup.sh

There is 1 TimesTen instance installed locally :

1) sb1122 (TimesTen11.2.2.4)

Of the following options :

  [1] Install a new instance
  [2] Upgrade an existing instance
  [3] Display information about an existing instance
  [q] Quit the installation

Which would you like to perform? [ 1 ] 2

NOTE: There is only one instance which can be upgraded.

Instance Name        : sb1122
Product Installed     : TimesTen11.2.2.4
Installation Directory : /scratch/sboand/TimesTen/sb1122
BitLevel             : 64
Component Installed   : Client/Server and DataManager
Daemon Port          : 33333

NOTE: Upgrading will remove the selected instance (and any directories or files
      under its path) and then re-install into the same directory. You will
      have the option to retain configuration files in :
      /scratch/sboand/TimesTen/sb1122/info

Would you like to upgrade this instance? [ yes ]

NOTE: <install_dir>/info contains information related to the data
```

```
       stores that have been created with this release. If you remove
       /scratch/sboand/TimesTen/sb1122/info
       you will no longer be able to access your data stores,
       nor would you be able to restore nor migrate your data.

       Would you also like to remove all files in <install_dir>/info? [ no ]
       Would you like to remove the DemoDataStore directory in
/scratch/sboand/TimesTen/sb1122/info/DemoDataStore? [ yes ] no
       Would you also like to remove all files in <install_dir>/network/admin/samples?
[ no ]
       /scratch/sboand/TimesTen/sb1122 Removed (retained <install_dir>/info)
TimesTen uninstall completed.

Of the three components:
   [1] Client/Server and Data Manager
   [2] Data Manager Only
   [3] Client Only

Which would you like to install? [ 1 ]
Upgrading installation in /scratch/sboand/TimesTen/sb1122
Where would you like to create the daemon home directory?
[ /scratch/sboand/TimesTen/sb1122/info ]

The daemon logs will be located in /scratch/sboand/TimesTen/sb1122/info
Would you like to specify a different location for the daemon logs? [ no ]
Installing into /scratch/sboand/TimesTen/sb1122 ...
Uncompressing ...

NOTE: For security, we recommend that you restrict access to the
       TimesTen installation to members of a single OS group. Only members of
       that OS group will be allowed to perform direct mode connections to
       TimesTen, and only members of that OS group will be allowed to perform
       operations that access TimesTen data stores, TimesTen files and shared
       memory. The OS group defaults to the primary group of the instance
       administrator. You can default to this group, choose another OS group
       or you can make this instance world-accessible. If you choose to make
       this instance world-accessible, all database files and shared memory
       are readable and writable by all users.

Restrict access to the TimesTen installation to the group 'g900'? [ yes ]

NOTE: Enabling PL/SQL will increase the size of some TimesTen libraries.

Would you like to enable PL/SQL for this instance? [ yes ]

Do you want to replace the ttendaemon.options file in
/scratch/sboand/TimesTen/sb1122/info? [ yes ] no

NOTE: The existing daemon options file has been retained. The default options
       file was written as
       /scratch/sboand/TimesTen/sb1122/info/ttendaemon.options.sb1122.

In order to use the 'TimesTen Cache' feature in any databases
created within this installation, you must set a value for the TNS_ADMIN
environment variable. It can be left blank, and a value can be supplied later
using <install_dir>/bin/ttModInstall.

A value for TNS_ADMIN (/scratch/sboand/TimesTen/sb1122/network/admin/samples) was
found in the previous daemon options file.
```

```
Please enter a value for TNS_ADMIN (s=skip)?
[ /scratch/sboand/TimesTen/sb1122/network/admin/samples ] s

Installing server components ...
What is the TCP/IP port number that you want the TimesTen Server to listen on?
[ 33334 ]
Do you want to install QuickStart and the TimesTen Documentation? [ no ]
Would you like to install the documentation (without QuickStart)? [ yes ] no

An existing cluster.oracle.ini file has been detected in
/scratch/sboand/TimesTen/sb1122/info.

Would you like to replace the existing cluster.oracle.ini file? [ no ]

The existing cluster.oracle.ini file will be used.
The sample cluster.oracle.ini file will be saved as
'/scratch/sboand/TimesTen/sb1122/info/cluster.oracle.ini.tt1122'.

An existing sys.odbc.ini file has been detected in
/scratch/sboand/TimesTen/sb1122/info.

NOTE: You may not be able to successfully run the demos if you keep your
      existing sys.odbc.ini file. If you choose to replace the existing
      file, a backup will be made automatically.

Would you like to replace the existing
/scratch/sboand/TimesTen/sb1122/info/sys.odbc.ini file ? [ no ]

The existing sys.odbc.ini file will be used.
The sample sys.odbc.ini file will be saved as
'/scratch/sboand/TimesTen/sb1122/info/sys.odbc.ini.sb1122'.

An existing tnsnames.ora file has been detected in
/scratch/sboand/TimesTen/sb1122/network/admin/samples.

Would you like to replace the existing
/scratch/sboand/TimesTen/sb1122/network/admin/samples/tnsnames.ora file ? [ no ]

The existing tnsnames.ora file will be used.
The sample tnsnames.ora file will be saved as
'/scratch/sboand/TimesTen/sb1122/network/admin/samples/tnsnames.ora.tt1122'.

Installing client components ...

An existing sys.ttconnect.ini file has been detected in
/scratch/sboand/TimesTen/sb1122/info.

NOTE: You may not be able to successfully run the Client/Server demos if you keep
      your existing sys.ttconnect.ini file. If you choose to replace the existing
      file, a backup will be made automatically.

Would you like to replace the existing
/scratch/sboand/TimesTen/sb1122/info/sys.ttconnect.ini file ? [ no ]

The existing sys.ttconnect.ini file will be used.
The sample sys.ttconnect.ini file will be saved as
'/scratch/sboand/TimesTen/sb1122/info/sys.ttconnect.ini.sb1122'.

Would you like to use TimesTen Replication with Oracle Clusterware? [ no ] yes
```

```
A Clusterware installation was detected in /scratch/oracle/crs/app/11.2.0/grid

Please provide the path to the Oracle Clusterware installation on this machine
(s=skip)? [ /scratch/oracle/crs/app/11.2.0/grid ]

NOTE: The TimesTen Clusterware agent port must be the same on all nodes
      of the cluster. Please refer to the TimesTen documentation for
      additional information.

Please enter a port number for the TimesTen Clusterware agent? [ 33339 ]

Executing '/scratch/oracle/crs/app/11.2.0/grid/bin/olsnodes' ...
Oracle Clusterware is currently configured on the following nodes :

1. host1
2. host2
3. host3
4. host4
5. host5

NOTE: By default, all of the nodes listed above will be added to the TimesTen
      Replication with Oracle Clusterware configuration. You can also
      specify your own list of nodes based on the list above.

Would you like to specify a node list for TimesTen Replication with Oracle
Clusterware? [ no ] yes
From the nodes above, please provide a list of nodes that you would like to add
(ex: 1,2 or skip)? [  ] 4,5

TimesTen Replication with Oracle Clusterware will be configured for the following
nodes :

host4
host5

Are you sure? [ yes ]
Overwrite the existing TimesTen Clusterware options file? [ no ]
The new TimesTen Clusterware options file will be located here :
        /scratch/sboand/TimesTen/sb1122/info/ttcrsagent.options.tt1122.

NOTE: The TimesTen daemon startup/shutdown scripts have not been installed.

Run the 'setuproot' script :
        cd /scratch/sboand/TimesTen/sb1122/bin
        ./setuproot -install
This will move the TimesTen startup script into its appropriate location.

The startup script is currently located here :
  '/scratch/sboand/TimesTen/sb1122/startup/tt_sb1122'.

The documentation was not installed.
To manually install the documentation, run the command 'setup.sh -installDoc'

The 11.2.2.6 Release Notes are located here :
  '/scratch/sboand/TimesTen/sb1122/README.html'

Starting the daemon ...
TimesTen Daemon startup OK.
End of TimesTen installation.
```

Upgrades when using parallel replication

Automatic parallel replication is enabled by default beginning in TimesTen release 11.2.2.2.0. In previous releases of 11.2.2 and in releases of 11.2.1, user-specified replication was available, but automatic parallel replication was not available. Automatic parallel replication with disabled commit dependencies is available in TimesTen release 11.2.2.8.0.

> **Note:** The values for the `ReplicationApplyOrdering` attribute, documented in *Oracle TimesTen In-Memory Database Reference*, have changed. Beginning in release 11.2.2.2.0, a value of 0 enables automatic parallel replication. Before release 11.2.2.2.0, a value of 0 disabled user-specified parallel replication. Beginning in release 11.2.2.8.0, a value of 2 enables automatic parallel replication with disabled commit dependencies.

You can perform an online or offline upgrade from a database that has not enabled parallel replication to a database of this release that has enabled parallel replication (with or without disabled commit dependencies).

The rest of this section discusses additional considerations along with scenarios where an offline upgrade is required.

Considerations regarding parallel replication

Be aware of the following considerations when upgrading systems that use parallel replication:

- Consider an active standby pair without parallel replication enabled. To upgrade the TimesTen instances to release 11.2.2.3.0 or higher and use automatic parallel replication (default value of 0 for the `ReplicationApplyOrdering` attribute), simply use the appropriate procedure for an active standby pair upgrade. See "Performing an upgrade with active standby pair replication" on page 3-20.

- Consider an active standby pair with no cache groups and automatic parallel replication enabled (value of 0 for the `ReplicationApplyOrdering` attribute). To upgrade the TimesTen instances to release 11.2.2.8.0 or higher to use automatic parallel replication with disabled commit dependencies (value of 2 for the `ReplicationApplyOrdering` attribute), use the procedure for an active standby pair online major upgrade. See "Online major upgrade for active standby pair" on page 3-22. The value for the `ReplicationApplyOrdering` attribute must be changed from 0 to 2 before restoring any of the databases. For example:

```
ttMigrate -r "DSN=master2;ReplicationApplyOrdering=2;ReplicationParallelism=2;
  LogBufParallelism=4" master2.bak
```

> **Note:** You may upgrade a TimesTen database with a replication scheme with `ReplicationApplyOrdering=2` to a database with `ReplicationApplyOrdering=0` by using the same active standby pair online major upgrade procedure.
>
> Automatic parallel replication with disabled commit dependencies supports only asynchronous active standby pairs with no cache groups. For more information, see "Configuring parallel replication" in the *Oracle TimesTen In-Memory Database Replication Guide*.

- You cannot replicate between databases that have the `ReplicationParallelism` attribute set to greater than 1 but have different values for the `ReplicationApplyOrdering` attribute.

Scenarios that require an offline upgrade

You must use an offline upgrade for these scenarios:

- Moving from user-specified parallel replication to automatic parallel replication. For example, from a release preceding 11.2.2.3.0 to release 11.2.2.3.0 or higher with the `ReplicationApplyOrdering` attribute set to the default value (`0`).

- Moving from an automatic parallel replication environment to another automatic parallel replication environment with a different number of tracks, as indicated by the value of the `ReplicationParallelism` attribute.

- Moving between major releases (such as from 11.2.1.*x.x* to 11.2.2.*x.x*) and using asynchronous writethrough cache groups.

- Moving from regular replication with asynchronous writethrough in 11.2.1.*x.x* to automatic parallel replication with asynchronous writethrough in 11.2.2.*x.x*.

For offline upgrades, you can use the procedure described in "Performing an offline upgrade" on page 3-11. Alternatively, you can upgrade one side and use the `ttRepAdmin -duplicate -recreate` command to create the new database.

Record of upgrades with replication configured

If replication is configured on a database, a new set of tables with the prefix `ttrep_ schema_version-` is created each time the `ttMigrate -r` command is used. These tables provide a history of the replication scheme from release to release.

These tables do not take up much space and are helpful when debugging upgrade problems. However, you can drop these tables if you do not experience problems with replication after running `ttMigrate`.

For example, after two migrations, the database might contain tables like this:

```
TTREP_SCHEMA_VERSION_004.REPELEMENTS
TTREP_SCHEMA_VERSION_004.REPLICATIONS
TTREP_SCHEMA_VERSION_004.REPPEERS
TTREP_SCHEMA_VERSION_004.REPSTORES
TTREP_SCHEMA_VERSION_004.REPSUBSCRIPTIONS
TTREP_SCHEMA_VERSION_004.REPTABLES
TTREP_SCHEMA_VERSION_004.TTSTORES
TTREP_SCHEMA_VERSION_005.REPELEMENTS
TTREP_SCHEMA_VERSION_005.REPLICATIONS
TTREP_SCHEMA_VERSION_005.REPPEERS
TTREP_SCHEMA_VERSION_005.REPSTORES
TTREP_SCHEMA_VERSION_005.REPSUBSCRIPTIONS
TTREP_SCHEMA_VERSION_005.REPTABLES
TTREP_SCHEMA_VERSION_005.TTSTORES
```

Performing a Client/Server online upgrade

TimesTen Server can talk directly to all user applications linked with TimesTen Client ODBC driver release 6.0 or higher. There are at least the following possible ways to upgrade a TimesTen Client/Server installation, depending on your requirements for client access to the database.

- If you do not need the database in the TimesTen instance being upgraded to be available continuously to your client applications, you may simply stop the old server, perform the migration of the database using `ttMigrate`, and then start the new release of the server, which should be configured to listen on the same server port.

- If it is critical that the database be continuously available to the client applications, you may use the procedure outlined in "Performing an online upgrade with replication" on page 3-14 to keep a second copy of the database available during the migration of the first copy.

Client/Server online upgrade

To perform an online upgrade of a TimesTen Client/Server system to a new major release (such as from 11.2.1.*x.x* to 11.2.2.*x.x*) with minimal reconfiguration, perform the following steps:

1. Stop the TimesTen Server on the old release of TimesTen. From this point until the TimesTen Server for the new release is started, your client applications do not have access to the database. Any attempted updates to the database by the clients fail, and you should stop user applications if necessary.

2. Install the new release of TimesTen. At installation time, configure the server to listen on the same port as the old release of TimesTen.

3. Use `ttMigrate` to migrate the database from the old release to the new release. See "Moving to a different major release of TimesTen" on page 3-12 for an example of this procedure.

4. Start the new release of TimesTen Server. The client applications now have access to the post-upgrade databases.

> **Note:** Since both releases of TimesTen Server have been configured to listen on the same port in this procedure, to restart the server for the old release you must first configure it to listen on a different port.

Client/Server online upgrade with continuous access to the database

The migration of a database using `ttMigrate` can be a long process if your databases are very large. If it is critical that your client applications have nearly continuous access to the database during the Client/Server online upgrade procedure, you may incorporate the procedure for performing an online upgrade with replication by following these steps:

1. Install the new release of TimesTen, making sure to configure TimesTen Server to listen on the same port as the old release. The installation script asks whether you would like to start the new server. You must answer "no".

2. Follow the steps in "Performing an online upgrade with replication" on page 3-14 to upgrade the TimesTen instance for one copy of the database. The client applications remain connected to the other, non-upgraded copy of the database.

3. Disconnect all clients from the old release of the database.

4. Stop the old release of TimesTen Server.

5. Wait for all updates to finish replicating from the old release of the database to the new release.

6. Start the new release of TimesTen Server. It begins listening on the same port as the old release, and client applications may now connect to the new release of the database with no change to their configurations.

A

Oracle TimesTen In-Memory Database License Information

This appendix contains licensing information for the Oracle TimesTen In-Memory Database. It contains the following topics:

- Licensing Options
- Oracle TimesTen Product Components

Licensing Options

Oracle TimesTen In-Memory Database is available in four licensing options, referencing the TimesTen product components described later in this document:

Oracle TimesTen In-Memory Database

The *Oracle TimesTen In-Memory Database* license includes the features provided by the following product components:

- TimesTen Base Functionality
- TimesTen In-Memory Analytics
- TimesTen Replication

Oracle TimesTen Application-Tier Database Cache

The *Oracle TimesTen Application-Tier Database Cache* license includes the features provided by the following product components:

- TimesTen Base Functionality
- TimesTen Caching from Oracle Database
- TimesTen Replication

This license requires an Oracle Database Enterprise Edition license for the source database.

Oracle TimesTen Application-Tier Database Cache for Oracle Applications

The *TimesTen Application-Tier Database Cache for Oracle Applications* license provides the same set of functionality as that offered by the TimesTen Application-Tier Database Cache product.

This license is available for use only with the Oracle Billing and Revenue Management (BRM) server product.

Oracle TimesTen In-Memory Database for Exalytics

The *TimesTen In-Memory Database for Exalytics* license includes the features provided by the following product components:

- TimesTen Base Functionality
- TimesTen In-Memory Analytics
- Integration with Oracle Business Intelligence (OBI) server for in-memory reporting and analysis running on Oracle Exalytics In-Memory Machine

Oracle TimesTen Product Components

TimesTen Base Functionality

- Relational in-memory database with persistence on disk and flash storage
- SQL query processing and transaction processing with ACID properties
- Utilities and built-in procedures for database operations and administration
- Application programming interfaces including JDBC, ODBC, OCI (Oracle Call Interface), ODP.NET, and PL/SQL supporting direct-linked and client/server connections
- SQL Developer support for TimesTen database application development
- TimesTen Plug-in for Oracle Enterprise Manager
- All other functionality not explicitly defined in the sections below

TimesTen In-Memory Analytics

- In-Memory Columnar Compression
- OLAP Group Operators including Cube, Grouping Set, Rollup
- Star-Join query performance optimizations

TimesTen Replication

- Real-time transactional replication between TimesTen databases
- All replication topologies including Classic Active-Standby, Active-Standby Pair, and Active-Standby Pair with Read-only Subscribers
- Automated failover management of TimesTen databases, processes and applications using Oracle Clusterware.

The use of Oracle Clusterware to manage Active-Standby Pair replication is available without additional licensing cost, provided that the customer has an active Oracle Support agreement for any of the TimesTen product licenses.

TimesTen Caching from Oracle Database

- Caching Oracle Database tables into a TimesTen database with automatic data synchronization between the TimesTen database and the Oracle Database
- Cache Group features and functionality

- Read/write and read-only access with automatic data synchronization for cache tables
- Multi-node Cache Grid features and functionality

Index

W