Oracle® TimesTen In-Memory Database

Installation Guide

11*g* Release 2 (11.2.2)

E21632-09

September 2012

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

E21632-09

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

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

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

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

2 Migration, Backup, and Restoration

3 TimesTen Upgrades

Index

Preface

The Oracle TimesTen In-Memory Database provides application-tier database and transaction management built on a memory-optimized architecture accessed through industry-standard interfaces. Optional data replication and Oracle caching extend the product to enable multi-node and multitier configurations that exploit the full performance potential of today's networked, memory-rich computing platforms.

The TimesTen CD contains a `readme.html` file that is the latest version of the release notes. These notes list product information and late changes to the printed documentation.

Related documents

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

`http://www.oracle.com/technetwork/products/timesten/documentation/`

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 unless otherwise noted.

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).
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 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

Where features or procedures differ from other Windows platforms, this version of the document includes information for Windows 7 systems.

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 1-22.

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 version 1.0 server. For details on installing TimesTen on Exalogic, see "Installing TimesTen on Exalogic systems" on page 1-27.

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 on Exalogic systems

Note: This document does not discuss TimesTen installation on an Exalytics system, as TimesTen is preinstalled on Exalytics. Go to the following location for Exalytics documentation:

`http://www.oracle.com/technetwork/documentation/exalytics-doc-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
- Viewing the online documentation

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

- Installation problems

- Licensing Information

Installation instances

On UNIX, you can install multiple instances of TimesTen. 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 11.2.2.3.0 on the same Windows computer, but you cannot install both 11.2.2.0.0 and 11.2.2.3.0.

You can retrieve information about the TimesTen instance name, release number and port settings using the `ttVersion` utility.

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 is not a supported configuration.

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 that particular installation of TimesTen.

WINDOWS

On Windows, the TimesTen installation scripts do not prompt you to supply an instance name. The instance name on Windows is `tt1122_32` on 32-bit systems and `tt1122_64` on 64-bit systems.

UNIX

On UNIX systems, by default, the instance name for this release is `tt1122_32` on 32-bit systems and `tt1122_64` on 64-bit systems. The default location is the TimesTen directory in the home directory of the user installing TimesTen. The instance name is case-insensitive, must be 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, you must supply a unique instance name and port number. The TimesTen installation script can detect if an instance of the particular release of TimesTen exists on the computer and prompts you for a new instance name and port number for the main TimesTen daemon.

Instance port numbers

Any time that you install multiple instances of TimesTen on the same computer, specify a unique TCP/IP port number for each TimesTen daemon during the install.

However, all TimesTen databases that replicate to each other must use the same daemon port number unless the `-remoteDaemonPort` option is specified in duplicate operations. This port number is set at install time. You can use the `ttVersion` utility to verify the port number of your installation of TimesTen.

The port on which the TimesTen cache agent listens is determined by the operating system and cannot be configured separately.

Choosing the appropriate TimesTen components

TimesTen 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 install the TimesTen Quick Start and documentation.

Type	Description
TimesTen Client	Installs the TimesTen Client only. No other TimesTen components are installed on the computer. 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 the 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 which you would like to change.

Components available on Windows

On Windows you can install one or more of the following components by checking the appropriate boxes during installation.

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. Use this installation to: ■ Allow a Client on another computer to access the TimesTen Server on this computer. ■ Allow the TimesTen Clients on other computers to access the TimesTen Server on this computer.
TimesTen Client	Installs the TimesTen Client. Use this installation to allow the TimesTen Client 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, which includes demos.

Type	Description
TimesTen Documentation	Installs the TimesTen Documentation Library.

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 the appropriate version of the JDK installed on your computer to use JDBC.

For improved JDBC performance on TimesTen, when using a XenNet virtual device, configure the `LargeSendOffload` parameter to `FALSE`. By default, the `LargeSendOffload` parameter is set to `TRUE` for the XenNet virtual device. This parameter can be modified either in the Windows registry or in the **Advanced** tab of the XenNet properties dialog.

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*.)

This section also discusses the platform-specific prerequisites:

- General UNIX requirements
- AIX 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
- Java
- Other Client/Server settings

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
```

Using `vxtunefs` online option requires Advanced VxFS.

Semaphores

TimesTen consumes one `SEMMNI` per active database, plus one additional `SEMMNI` per TimesTen instance where Client/Server communication is done through shared memory. For each active database, TimesTen consumes 155 `SEMMSL`, plus one `SEMMSL` for each connection.

For example, if the value of the `Connections` attribute is set to `100`, then TimesTen requires a minimum of 255 `SEMMSL`. It is best to set parameter to a higher value, for example for Linux `x8664` to support 2000 connections you should set

```
kernel.sem = 2200 6400 200 25
```

> **Note:** You can use the following formula as a guide, although in practice, `SEMMNS` and `SEMMNU` can be much less than `SEMMNI * SEMMSL` because not every program in the system needs semaphores.
>
> `SEMMNS=SEMMNU = (SEMMNI * SEMMSL)`

For more information about the `Connections` attribute, see the "Connection Attributes" chapter in the *Oracle TimesTen In-Memory Database Reference*.

Java

If you are running 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 Sun 64-bit JVM, you may need to pass the `-d64` option on the `java` command line.

Other Client/Server settings

The maximum number of concurrent IPC connections to a TimesTen Server allowed by TimesTen is 9,999. However, system limits can take precedence on the number of connections to a single DSN. Client/Server users can increase the file descriptor limit to support a large number of connections and processes.

For example, on Solaris, you may change the file descriptor limit to have a maximum of 1024 simultaneous server connections by adding the following line to /etc/system:

```
set rlim_fd_max = 1080
```

In this case, 1080 is greater than the number of anticipated client/server connections and allows for a few extra connections.

AIX prerequisites

On AIX, before installation, set the kernel parameter sb_max to a minimum of 512 KB if you plan to use replication. The replication agent requests TCP send and receive buffers of a minimum size of 512 KB. The value may be changed using the following command.

```
# /usr/sbin/no -p -o sb_max=524288
```

To query the value, use the following command:

```
# /usr/sbin/no -o sb_max
```

Large pages

On AIX 5.3 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
```

Asynchronous I/O

If you plan to use PL/SQL with AIX versions preceding 6.1, enable the AIO (Asynchronous Input Output) device drivers on the system where the TimesTen software is installed. To manually enable asynchronous I/O:

1. Start smitty aio.
2. Run Change/Show Characteristics of Asynchronous I/O.
3. Set the STATE to be configured at system restart as **available**.

This procedure is neither applicable nor feasible in AIX Version 6.1, where AIO is enabled dynamically.

Note: The above procedure does not require a system restart.

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 the release notes for details.

The C development tools are required if you intend to do native development.

On Linux, before installation, the following sections describe steps you can perform to improve the performance of TimesTen on your system:

- Required Library Packages
- Semaphores
- Shared memory
- IPC Client/Server
- Replication
- IMDB Cache

Large pages

Large pages can be enabled only if the running Linux kernel supports large pages (also called "huge pages" in the Linux community).

If large pages are supported by the kernel, there should be special files in the `/proc` directory that indicate the number and size of the large pages.

On Linux 2.4.x systems, the `/proc/sys/vm/hugetlb_pool` indicates the total size of the large pages.

On 2.6.x systems, the `/proc/sys/vm/nr_hugepages` file indicates the total number of large pages.

You can change the total number and size of the large pages by changing the contents of those files. For example, you can use:

```
echo 32 > /proc/sys/vm/nr_hugepages
```

To see the number and size of the allocated large pages:

```
cat /proc/meminfo
```

The following output from this command would indicate that you have 16 large pages, each of the size 256 MB for a total of 4 GB:

```
HugePages_Total: 16
HugePages_Free: 16
Hugepagesize: 262144 kB
```

> **Note:** Since large pages must be allocated on a contiguous memory space, the actual large page size allocated may be smaller than requested. Also, the large page size itself is not configurable. The value of `Hugepagesize` in `/proc/meminfo` indicates the system's fixed large page size.

If PAM (Pluggable Authentication Modules) is enabled, you may need to modify the `/etc/security/limits.conf` file to increase the `memlock` limit. By default, the limit is 32 KB.

You must also set `/proc/sys/vm/hugetlb_shm_group` to the group ID of the user that is running the main TimesTen daemon.

The operating system now is ready for the large page support. To enable this feature on TimesTen, simply set `-linuxLargePageAlignment` *`Size_in_MB`* in the daemon options file (`ttendaemon.options`).

You should specify the large page alignment size in MB, which is the `Hugepagesize` value in `/proc/meminfo`.

After you set up large pages, TimesTen uses as many large pages as possible. If there are not enough pages, TimesTen uses the normal pages after consuming all available large pages.

When TimesTen uses large pages, the `HugePages_Free` file in `/proc/meminfo` changes.

Required Library Packages

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

Semaphores

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 a large number of connections.

For example, if you plan to use 200 connections, we recommend that you add the following line to the `/etc/sysctl.conf` file:

```
kernel.sem = 400 32000 100 128
```

The first parameter is the maximum number of semaphores per array (`SEMMSL`), 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 or run the following command:

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

Shared memory

To increase the shared memory size to 2048 MB, log in as `root` and edit the `/etc/sysctl.conf` file by adding the following line.

```
kernel.shmmax=2147483648
```

If your configuration is greater than 8 GB, you should also increase the value of the `shmall` parameter. The value is in KB and should be equal to `ceil(SHMMAX/PAGE_SIZE)`. Page size is generally 4K on x86 systems and 16K on Itanium. For example, for a 64 GB database on Itanium, you should specify the following parameters values:

```
kernel.shmmax=68719476736
kernel.shmall=4194304
```

To increase the shared memory size without restarting:

```
% /sbin/sysctl -w kernel.shmmax=2147483648
```

If you have your kernel configured with the `/proc` file system and it is mounted, then the current maximum shared memory segment size (in bytes) can be viewed by the following command:

```
% cat /proc/sys/kernel/shmmax
```

You can also change this value by the following command:

```
% echo 2147483648 > /proc/sys/kernel/shmmax
```

This command has the same effect as the `sysctl` command.

IPC Client/Server

On Red Hat Linux systems, to enable more than six ShmIpc Client/Server connections, add the following line to the `/etc/sysctl.conf` file:

```
kernel.sem = 270 32000 100 128
```

Then restart or run the following command.

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

Replication

For replication, TCP send and receive buffers should be increased to *a minimum* of 512 KB. To increase the buffers to 4 MB, add the following lines to the `/etc/sysctl.conf` file:

```
net.ipv4.tcp_rmem=4096 4194304 4194304
net.ipv4.tcp_wmem=98304 4194304 4194304
net.core.rmem_default=65535
net.core.wmem_default=65535
net.core.rmem_max=4194304
net.core.wmem_max=4194304
net.ipv4.tcp_window_scaling=1
```

Then restart or run this command:

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

IMDB Cache

For IMDB Cache, TCP send and receive buffers should be increased to even greater values. To make these changes, add the following lines to the `/etc/sysctl.conf` file:

```
net.ipv4.tcp_rmem=4096 4194304 4194304
net.ipv4.tcp_wmem=98304 4194304 4194304
net.core.rmem_default=262144
net.core.wmem_default=262144
net.core.rmem_max=4194304
net.core.wmem_max=4194304
net.ipv4.tcp_window_scaling=1
net.ipv4.ip_local_port_range="1024 65000"
```

Then restart or run this command:

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

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

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 user 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 group project for the group `timesten 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 "process.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 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)" 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 "process.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 have the following 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 proceeds the default entry.

  Or:

- Add the following line to a `.profile` or `.cshrc` script or execute it before restarting daemons. This uses the project and process settings. This is required if project created with `-G` only.

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

Windows requirements

The TimesTen debug libraries depend on Visual Studio 2003, 2005, 2008, or 2010. If you intend to use the debug libraries, ensure that one of these versions is installed.

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. If User Account Control is enabled, and you are logged in as the local Administrator, then you can successfully run these operations in the usual way. However, if you are logged in as a member of the TimesTen users group, then you must explicitly invoke these tasks with Windows Administrator privileges.

To start a command prompt window with Windows Administrator privileges, you can 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 will indicate "Administrator" in the title bar.

Default installation directories

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

- On Windows, `C:\TimesTen`
- On UNIX, `$HOME/TimesTen`

TimesTen creates temporary files when a transaction frees a large amount of space in a database. In addition, other TimesTen operations, such as large deletes, use the temporary directory when copying files.

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: On Windows, the complete temporary directory path must be less than 190 characters for the installation to complete successfully. In addition, 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.

Considerations for locations of database files and other user files

During installation, if you have elected to install the TimesTen Quick Start, the installer will prompt you for a location for the `DemoDataStore` directory. By default, this will be located under the `info` directory that is under the Timesten installation path. We strongly recommend that you choose an alternate location, outside of the TimesTen installation path, for this directory.

We also strongly recommend 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 uninstall 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 TimesTen users groups
- Directory and file permissions
- Creating UNIX TimesTen administrator and users groups

TimesTen instance administrators and TimesTen users groups

For security, we restrict access to the TimesTen installation to members of a single operating system group, under which TimesTen is installed. We refer to this group 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.

WINDOWS

- On Windows, which does not have the same concept of "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:

UNIX

  - 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 computer.
  - 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 administrator and users groups" on page 1-14.

Directory and file permissions

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

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.

WINDOWS

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 TimesTen users groups" on page 1-13.

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

UNIX

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 administrator's 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 "Creating UNIX TimesTen administrator and users groups" on page 1-14.

> **Note:** Checkpoint files and log files for databases should be installed on separate operating system devices. TimesTen returns a message to the daemon log if the transaction log files and checkpoint files for your databases are on the same operating system device.

Creating UNIX TimesTen administrator and users groups

UNIX

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

- Create the TimesTen users group
- Create the TimesTen instance registry and 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's 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 user group is to uninstall and reinstall the TimesTen instance, providing the new group name during reinstall.

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. Log in as `root`.

2. Create an operating system group under which the TimesTen instance can be installed. In creating this operating system group, we suggest using the name `timesten`, but you can choose any name that you prefer.

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

3. Add the user who is installing and any users who are administering TimesTen to the TimesTen users group that you just created.
4. 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 1-13 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 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. Log in as `root`.
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. Create an operating system group for the TimesTen instance administrators group. You can name this group as you want. For our examples, we use the name `ttadmin`.

   ```
   # groupadd ttadmin
   ```

4. Assign ownership permissions on the `/etc/TimesTen` directory to the TimesTen instance administrators group so that only the instance administrator may access and execute. At install 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/*
```

5. 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
- Working with the TimesTen daemon and server on UNIX systems
- 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, follow these 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 linux86
   ```

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.
`-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 install procedure fails. We recommend 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.
`-quickstart`	Installs the TimesTen Quick Start.
`-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. Batch files from older releases of TimesTen cannot be used to install the current release. All new prompts in the installation script for this release are assigned default answers and may produce unexpected results when batch files from different releases are used.

We recommend 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. (This option enables you to upgrade from a release preceding the TimesTen 11.2.2 release.)
  - 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 at least one alphanumeric string and no longer than 255 characters.

- 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 `$HOME/TimesTen`. The TimesTen documentation refers to the installation directory as *`install_dir`*.
- Prompts you for the location of the TimesTen daemon home directory.
- Prompts you for the location of TimesTen daemon log files. The default is *`install_dir`*`/info`.
- Prompts you to specify the daemon port number. (The default port number is shown.)

> **Note:** All installations that replicate to each other must use the same daemon port number that is set at installation time. Verify the daemon port number by running the `ttVersion` utility.

- 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 1-13 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 enabled for the instance. The default answer is "yes". If not enabled at install time, PL/SQL can be enabled for the instance at a later time using the `ttmodinstall` utility.

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

- Prompts you to set the location to be supplied for the `TNS_ADMIN` environment variable that specifies the directory where the `tnsnames.ora` file can be found.

  You can leave this field blank. If you do not specify the location at install time, you can set it later with the `ttmodinstall` utility. However, you must specify it before using the In-Memory Database Cache.
- Prompts you to specify the server port number. (The default port number is shown.) Installs the client and server components.
- Prompts you to install Quick Start and the TimesTen documentation. The TimesTen Quick Start applications can take up to 64 MB of disk space. The default directory is *`install_dir`*`/quickstart` and *`install_dir`*`/doc`.
- Prompts for the location of where to install the demo database. This indicates that when you install the Quick Start, the TimesTen demo database files are installed in the `DemoDataStore` directory that defaults to the *`install_dir`*`/info/DemoDataStore` location. See the section "Considerations for locations of database files and other user files" on page 1-12 for suggestions on selecting a location for the `DemoDataStore`.
- Installs the client components.
- 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.
- The install checks for any nodes where the Oracle Clusterware is currently configured and prompts you to specify a node list for TimesTen replication with Oracle Clusterware.
- 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.

The daemon writes a `timestend.pid` file into the directory from which the daemon was started. By default, this is *`install_dir`*`/info`. This file contains the daemon process ID. When the process terminates, the `timestend.pid` file is removed.

> **Note:** TimesTen writes a message to the daemon log if the transaction log files and checkpoint files for your databases are on the same device.

If you want the TimesTen instance to start each time the computer is restarted, run the `setuproot` script as `root`. The `setuproot` script is located in the *`install_dir`*`/bin` directory:

```
# cd install_dir/bin
# setuproot -install
```

Working with the TimesTen daemon and server on UNIX systems

The TimesTen main daemon (`timestend`) starts automatically when the operating system is booted and operates continually in the background. Application developers do not interact with `timestend` directly; no application code runs in the daemon and application developers do not, in general, have to be concerned with it. Application programs that use TimesTen databases communicate with the daemon transparently by using TimesTen internal routines.

There are situations, however, when you may have to start and stop the daemon manually, using the TimesTen main daemon startup script. This section explains how to start and stop the daemon. If you have installed the TimesTen Server, it starts automatically when the TimesTen daemon is started and stops automatically when the TimesTen daemon is stopped.

> **Note:** You must be the TimesTen instance administrator or have `root` privileges to interact with the TimesTen daemon.

To stop the daemon manually, use the utility command:

```
% ttDaemonAdmin -stop
```

To start the daemon manually, use the utility command:

```
% ttDaemonAdmin -start
```

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 take the following steps to use it if you have valid reason.

If messages are logged using the `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`.

The `syslog` facility allows messages to be routed in a variety of ways, including recording them to a file. The disposition of messages is under the control of the configuration file, `/etc/syslog.conf`.

Entries in the `syslog.conf` file contain two columns. The first column contains a list of the types of messages to log to a particular file. The second column contains the name of the log file. A tab appears between the message type and file name. Each entry in the `syslog.conf` file has the format: *`message_type file_name`*. Message types are specified in two parts:

```
subsystem-facility.severity-level
```

Depending on the configuration specified in that file, messages can be logged into various files. For the TimesTen daemon, specify the message types: `user.debug`, `user.info`, `user.warn` and `user.err`. You can also use the wildcard character * to represent the subsystem-facility. Since debug messages are ranked highest, specifying `*.debug` or `user.debug` is sufficient in preparing a file for the support or error log. In a message type list, delimit items by semi-colons. For example:

```
*.debug /var/adm/syslog/syslog.log
user.err; user.warn; user.info /var/adm/messages
```

To make changes to `/etc/syslog.conf`, you must have `root` privileges. Changes only take effect after the `syslog` daemon (`syslogd`) process is terminated (with the command `kill -1`) and restarted.

For further details, see your operating system documentation for `syslog.conf` or `syslogd` for information on configuring this file.

Note: If the `/etc/syslog.conf file` does not exist on your system, create one according to the `syslog.conf` manual page so the daemon can log its data to the `syslog` facility.

To determine if your `syslog` configuration file is set up correctly, run the TimesTen `ttSyslogCheck` utility. Finally, after `syslogd` has been set up correctly, you may use the TimesTen `ttDaemonLog` utility to view only those messages in the system log file that TimesTen logged.

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 the *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 the system 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.

Installing TimesTen on Windows systems

WINDOWS

This section discusses installation and related issues for Windows systems.

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, but you cannot do both a 64-bit installation and a 32-bit installation of the same release. (You could, however, have a 64-bit installation of the 11.2.2.2.0 release and a 32-bit installation of the 11.2.1.9.0 release, for example.)

Notes:

- You can install both the 64-bit client and the 32-bit client from a 64-bit installation.
- Before beginning installation, ensure that the prerequisites defined in "Installation prerequisites" on page 1-4 have been met.

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. A program takes you through the installation.

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 install program 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 11.2.1. If a previous version of the same TimesTen major release exists (such as 11.2.1.4.0 if you are trying to install 11.2.1.9.0), the installer returns an error message asking you to uninstall the previous release.

The TimesTen installer performs the following actions.

- Prompts you for the location of the installation. By default, TimesTen is installed under `C:\TimesTen\tt1122_32` (32-bit installation) or `tt1122_64` (64-bit installation).
- 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
    - TimesTen Documentation

  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, 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 1-12 for suggestions on selecting a location for the demo database files.

- Prompts you to set the location to be supplied for the `TNS_ADMIN` environment variable that specifies the directory where the `tnsnames.ora` file can be found.

  You can leave this field blank. If you do not specify the location at install time, you can set it later with the `ttmodinstall` utility. However, you must specify it before using the In-Memory Database Cache.

- Prompts you to select the Program Folder for the Start Menu. 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 1-13 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 specify whether PL/SQL should be enabled for the instance. By default, it is enabled. If not enabled at install time, PL/SQL can be enabled for the instance at a later time using the `ttmodinstall` utility.

- 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 script *install_dir*\bin\ttenv.bat. The ttenv script is described in "Setting environment variables for TimesTen" on page 1-30.

> **Note:** This is not advisable if you are installing multiple versions of TimesTen, in which case the path could cause unexpected behavior.

- 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 Guide. For information on the Quick Start, see "TimesTen Quick Start" on page 1-34.

> **Note:** TimesTen cannot be installed in a mapped network drive. Attempting to do so results in an error.

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 installation script 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** under the **Start** menu on the Windows Desktop.
2. On the Windows Desktop, from the **Start** menu, select **Control Panel**. Then select **Administrative Tools**. Then 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. On the Windows Desktop, from the **Start** menu, select **Control Panel**. Then select **Administrative Tools**. Then select **Services** and check that the **TimesTen Data Manager 11.2.2** service has status "Started".

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. On the Windows Desktop, from the **Start** menu, select **Control Panel**. Then select **Administrative Tools**. Then 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**.

Note:

- This step assumes you installed TimesTen Quick Start, which includes 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 the following:

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

Working with the Data Manager Service and the Server on Windows

When you install the TimesTen Data Manager, the TimesTen Data Manager Service starts automatically whenever the system restarts. In addition, if you installed the TimesTen Server, it is automatically started whenever the TimesTen Data Manager service is started. You can change the startup mode for the TimesTen Data Manager to require manual startup.

> **Note:** You must have administrative privileges to set the startup mode or to start and stop the TimesTen Data Manager service.

To change the startup mode:

1. On the Windows Desktop, from the **Start** menu, select **Control Panel**. Then select **Administrative Tools**. Then select **Services**. This displays all currently available services.
2. Double-click the **TimesTen Data Manager 11.2.2** service to examine its properties dialog.
3. In the properties dialog, the Startup type list should indicate **Automatic** (default). You can optionally change it to **Manual**. In either case, you can click **Stop** or **Start** (as applicable) in the properties dialog to stop or start the service. For typical usage, set the Startup type back to the default and click **OK** when you are through.

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 the *Oracle TimesTen In-Memory Database Reference*.

Uninstalling TimesTen on Windows systems

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

1. On the Windows Desktop, from the **Start** menu, select **Control Panel**.
2. Select the **Add/Remove Programs** control panel (or **Programs and Features** in Windows 7).
3. In the list of programs, select TimesTen—for example, select "Oracle TimesTen In-Memory Database 11.2.2 (64-bit)".
4. Click **Remove** (or **Uninstall** in Windows 7).

To verify that removal was successful, check the following:

- TimesTen 11.2.2 has been removed from **All Programs** in the **Start** menu.
- 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 Control Panel.

Notes:

- DSNs created by TimesTen installation are removed upon TimesTen uninstallation. DSNs created by users are not removed during TimesTen uninstallation.
- See "Verifying installation" on page 1-25 for information about the Services list and ODBC Data Source Administrator Control Panel.

Installing TimesTen on Exalogic systems

To Install TimesTen on an Exalogic system, you follow the prompts in the TimesTen installation scripts as described in the procedures in "Installing TimesTen on UNIX systems" on page 1-16. Also, ensure that you have met the prerequisites for Oracle Linux installations. See "Linux prerequisites" on page 1-7

We recommend that you install TimesTen on each of the compute nodes of an Exalogic system.

Note: Installing TimesTen binaries one time 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 the *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.

Specifically, we strongly recommend 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`*`/TT`*`instance`*

Installing Oracle Clusterware for use with TimesTen

Install Oracle Clusterware per instructions in the *Oracle Clusterware Administration and Deployment Guide*.

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 the *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

On Windows systems, the Windows driver manager supports anything up to Microsoft ODBC 3.5 SDK. TimesTen supports the Microsoft ODBC 2.5 SDK. For any ODBC applications that link with the latest version of the Microsoft ODBC Data Source Administrator (`ODBC32.LIB` file), the TimesTen driver manager handles the connection using ODBC 2.5.

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	What to include	For settings and other information, see:
`LIB`, `LIBPATH`, `LD_LIBRARY_PATH` or `SHLIB_PATH`	On UNIX systems, include the `lib` directory under the TimesTen installation directory.	"Shared library path environment variable" on page 1-32
`NLS_LANG`	If `NLS_LANG` is set to as `NA`, an OCI connection error or the ORA-12705 message is thrown.	On Windows, if an older version of Oracle has been installed, such as Oracle9*i*, the registry key `HKEY_LOCAL_computer\Software\ORACLE\NLS_LANG` may be set to an invalid value, such as NA. If this value is NA, the TimesTen installer replaces the value with `AMERICAN_AMERICA.US7ASCII`. This ensures that TimesTen OCI, Pro*C, and IMDB Cache can connect properly. TimesTen uses the Oracle Instant Client to make these connections. The Oracle Instant Client requires this value to be a valid `NLS_LANG` setting.

Environment variable	What to include	For settings and other information, see:
ODBCINI	Include the location where the odbc.ini file used by TimesTen databases is to be found.	"ODBCINI environment variable" on page 1-31
PATH	Include the bin directory under the TimesTen installation directory. On Windows, also include the path to the Oracle installation if you are using the IMDB Cache option.	"PATH environment variable" on page 1-31, "Shared library path environment variable" on page 1-32 and "Installing TimesTen on Windows systems" on page 1-22
SYSODBCINI	Set to the location where the sys.odbc.ini file used by TimesTen system databases is to be found. This environment variable should be set in the startup script.	"SYSODBCINI environment variable" on page 1-31
SYSTTCONNECTINI	Set to the location where the sys.ttconnect.ini file used by TimesTen Client applications to define logical server names.	"SYSTTCONNECTINI environment variable" on page 1-32
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 1-12
TNS_ADMIN	If using IMDB Cache, set to the location of the tnsnames.ora file. Required if you are using the IMDB Cache option.	"TNS_ADMIN environment variable" on page 1-32
Java	For Java applications, there are certain environment variables that must be set.	"Java environment variables" on page 1-32

The following sections describe environment variables in TimesTen:

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

Setting environment variables for TimesTen

If, after installation, you want to set the environment variables to standard TimesTen settings, use the ttenv script. This includes setting paths so that TimesTen utilities can be executed, among other things. You must invoke this script before starting TimesTen in order for any of the changes to take effect.

For UNIX platforms, use either of the following scripts depending on your shell:

```
install_dir/bin/ttenv.sh
install_dir/bin/ttenv.csh
```

WINDOWS

For a Windows platform, use the *install_dir*/bin/ttenv.bat script.

```
install_dir\bin\ttenv.bat
```

Execute the following for a description of the command-line options for ttenv:

```
source ttenv.csh -help
```

Or:

```
. ttenv.sh -help
```

PATH environment variable

TimesTen provides utilities for managing and debugging TimesTen applications. To make these utilities readily available, include the bin directory found in *install_dir* in the PATH environment variable.

> **Note:** *install_dir* is the directory where TimesTen is installed.

WINDOWS

On Windows, the PATH environment variable must also contain the bin directory of the Oracle installation, if you are using the IMDB Cache option.

ODBCINI environment variable

UNIX

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 the *Oracle TimesTen In-Memory Database Reference*.) By default on UNIX platforms, TimesTen first looks for the .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 the path name of an .odbc.ini file before launching the TimesTen application.

If TimesTen cannot locate a user DSN file, TimesTen also looks for the sys.odbc.ini file in *install_dir*/info. For more information, see "Overview of user and system DSNs" in *Oracle TimesTen In-Memory Database Operations Guide*.

SYSODBCINI environment variable

UNIX

TimesTen applications use the sys.odbc.ini file to define system data sources and their connection attributes. (For a description of connection attributes, see "Connection Attributes" in the *Oracle TimesTen In-Memory Database Reference*.) Any user on the computer 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.

If TimesTen cannot locate a user DSN file, TimesTen also looks for the sys.odbc.ini file in *install_dir*/info.

For more information, see "Overview of user and system DSNs" in *Oracle TimesTen In-Memory Database Operations Guide*.

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 the *Oracle TimesTen In-Memory Database Operations Guide*. By default on UNIX platforms, TimesTen looks in *`install_dir`*`/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.

TimesTen also looks for the `sys.ttconnect.ini` file under *`install_dir`*`/info`.

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

TNS_ADMIN environment variable

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

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

Shared library path environment variable

On Solaris and Linux systems, add *`install_dir`*`/lib` directory to the `LD_LIBRARY_PATH` environment variable.

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

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/ttjdbcjdk_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 6.0, *`jdk_ver`* is 6 and the file name would be `ttjdbc6.jar`.

Note: 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:

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

Or:

```
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`

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:

```
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 the option of installing TimesTen Quick Start, which includes a variety of tutorials, demo applications, and other resources. If you choose to install Quick Start, it is 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, demos, and sample code for administration, access control, application development, replication, and caching, including the following areas. The demo databases are first installed when you install the Quick Start.

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 In-Memory 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 demo
- 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 demos.

- Demo schema and setup: The `build_sampledb` script creates a sample database and demo schema. You must run this before you start using the demos.
- Demo 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 demos.
- Demos and setup: Quick Start demos are in subdirectories under the `quickstart/sample_code` directory. For instructions on compiling and running the demos, 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.

Viewing the online documentation

Online copies of TimesTen documentation are installed along with the TimesTen product unless you choose not to install the documentation. Documentation is provided in HTML and PDF format. The HTML can be viewed in your browser. The PDFs can be viewed with the Adobe Acrobat Reader. If you do not currently have the Adobe Acrobat Reader installed, it is available from the Adobe Systems web page, `http://www.adobe.com`.

Online documentation is installed in the *`install_dir`*`/doc` directory.

Note: The online documentation represents the most current release of the documentation.

Installation problems

To avoid problems during installation, ensure you have met all prerequisites. Using information in the installation guide and the release notes, check that:

- 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 administrator's group. See "TimesTen instance administrators and TimesTen users groups" on page 1-13.
- On Windows, you are installing as user `Administrator` who is a member of the local `Administrators` group.

Licensing Information

You can find TimesTen licensing information at:
`http://www.oracle.com/technetwork/database/timesten/documentation/licensing-info-1478940.pdf`

This document contains information on licensing:

- Oracle TimesTen In-Memory
- Oracle In-Memory Database Cache
- In-Memory Database Cache for Oracle Applications

  and
- TimesTen In-Memory Database for Exalytics

Note: Go to the following location for Exalytics documentation:

`http://www.oracle.com/technetwork/documentation/exalytics-doc-1645656.html`

2

Migration, Backup, and Restoration

This chapter describes migration, backup and restoration for the TimesTen database in these topics:

- Copying, migrating and restoring a database
- Backing up and restoring a database
- Globalization support during migration

Copying, migrating and restoring a database

The TimesTen utilities for copying, restoring and migrating a database enable you to perform the following:

- Migrate a database between releases of TimesTen. Use the `ttMigrate` utility. This utility saves tables and indexes from a TimesTen database into a binary file. The tables and indexes can then be restored into another TimesTen database. This enables you to migrate data between TimesTen releases.
- Migrate a database between different hardware platforms. Use the `ttBulkCp` utility to save and restore the data. The `ttBulkCp` utility copies data from TimesTen tables into ASCII files and back again, but only for those objects owned by the user executing the utility, and those objects for which the owner has `SELECT` privileges.

  The `ttBulkCp` utility enables you to copy a single table between databases, including between databases from different releases of TimesTen or between databases on different hardware platforms.

  The `ttSchema` utility saves the table definitions. Use `ttIsql` to re-create the tables from the saved table definitions.
- Take a snapshot of a database and then restore the database in the exact same state. Use the `ttBackup` and `ttRestore` utilities or the `ttBackup` and `ttRestore` C functions.

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*. These utilities are supported only where the TimesTen Data Manager is installed.

Every database backup contains the information needed to restore the database as it existed at a 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 will produce a backup that can be used 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 and 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 In-Memory 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 in order 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 referred to 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.

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

See also "Copying, migrating and restoring a database" on page 2-1 of this guide and 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).

Note: The ttMigrate utility issues a warning whenever there is an implicit or explicit character set conversion while saving or restoring data.

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.

Note: 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 upgrading from one release of TimesTen to a newer release of TimesTen. It includes the following topics:

Introduction

When a TimesTen database is loaded into shared memory, many of its attributes are fixed, including size, logging options, TimesTen software release number, and the location of its checkpoint and transaction log files on disk. This chapter describes the steps required to change these attributes and to upgrade TimesTen databases when you install a new release of TimesTen.

The following sections describe the compatibility and character set of the installed TimesTen release:

Database compatibility

In prior releases of TimesTen, release numbers used either five parts (7.0.0.0.0 and later), or three parts (releases before 7.0.0.0.0, such as 6.0.17) to represent the release. Starting with TimesTen release 11.2.1.0.0, five parts are used, with the first three parts, such as 11.2.2, indicating a major release of TimesTen. The fourth part indicates the patch release of a major release, and the fifth part indicates a port patch. For example, TimesTen release number 11.2.2.2.1 indicates the first port patch of the second patch release of TimesTen 11*g* Release 2 (11.2.2).

TimesTen databases are not compatible between major releases, but they are always compatible between patch releases. For example, a database created with TimesTen release 7.0.5.1.0 is not compatible with a TimesTen release 11.2.1.1.0 application, but a database created with TimesTen 11.2.1.0.0 is compatible with a TimesTen release 11.2.1.1.0 application.

Data type compatibility

TimesTen supports a selection of Oracle 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 the *Oracle TimesTen In-Memory Database SQL Reference*. Because some Oracle 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 that is set for the database. See "TypeMode" in the *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 IMDB Cache to Oracle— use only Oracle data types with IMDB Cache to Oracle. If you want to use IMDB Cache to Oracle, you must convert any original TimesTen data types to the new Oracle data types when performing a database upgrade with ttMigrate. See "Converting data types to Oracle data types" on page 3-4 for details.

Oracle 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 "Upgrading data types as TimesTen data types" on page 3-5 for more information.

Database character set

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`. The value of this attribute is used to determine 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 the *Oracle TimesTen In-Memory Database Operations Guide* for more information.

Before upgrading your database from a release before TimesTen 7.0, you must specify a database character set by adding the `DatabaseCharacterSet` attribute to your database's DSN. This attribute is ignored by releases of TimesTen before 7.0. In most cases, you want to choose a database character set that makes sense for your region and that matches the character data that is present in your database. Consider the following important restrictions, however.

- If you plan to use the database with TimesTen IMDB Cache to Oracle, you must specify a value for `DatabaseCharacterSet` that is the same as the character set specified for the Oracle database that the TimesTen database connects to.
- Replication is not possible between databases with different character sets. Because databases created with releases of TimesTen before 7.0 do not have a database character set specified, a special database character set, `TIMESTEN8`, has been 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 database upgrade as an online upgrade with replication (see "Performing an online upgrade with replication" on page 3-16), then you must specify a `DatabaseCharacterSet` of `TIMESTEN8` in your DSN.
- If you use TimesTen Client/Server and intend to connect to the upgraded database 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 from TimesTen release 6.0 and above" on page 3-44.

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` to convert your database to a database character set other than `TIMESTEN8`. See "Database character set conversion" on page 3-5 for details.

Utilities used when upgrading TimesTen databases

TimesTen includes three utilities that may be used when upgrading databases.

The `ttBackup` and `ttRestore` utilities can export and import, respectively, an image copy of the TimesTen database. This database image is only compatible between patch releases of TimesTen, and it must be backed up and restored using the same bit-level of TimesTen (32-bit or 64-bit). You may use the `ttBackup` and `ttRestore` utilities to:

- Move a database from one TimesTen instance to another TimesTen instance running the same major release version of TimesTen.
- Move a database from one computer to another computer running the same major release version of TimesTen.
- Move a database from one patch release of TimesTen (such as 11.2.2.1.0) to a different patch release of TimesTen (such as 11.2.2.2.0).

The `ttMigrate` utility, however, exports the TimesTen database to a release-neutral format that can be used to migrate the database across major releases and between 32-bit and 64-bit versions of TimesTen. You must use the `ttMigrate` utility to:

- Move a database from one major release of TimesTen (such as 7.0) to a different major release (such as 11.2.2).
- Move a database from a 32-bit version of TimesTen to a 64-bit version (or vice versa).
- Reduce the size of a TimesTen database.

Note: The ttMigrate utility cannot migrate databases across different hardware platforms. For example, you cannot migrate a Windows database to a Solaris database. The release of ttMigrate must also match the release of the database you are copying from or to. In the procedures detailed in the following sections, use the ttMigrate utility in the older version to save the tables of the original database a disk file and use the ttMigrate utility in the new version to restore the file into the new database.

On Windows, these utilities use the ODBC driver manager.

On UNIX platforms, the utilities are directly linked with the TimesTen Data Manager ODBC driver.

For a full description of the syntax and usage of these and other utilities, see "Utilities" in the *Oracle TimesTen In-Memory Database Reference*.

Access control

Access control is mandatory in TimesTen beginning with release 11.2.1. If you have been using previous versions 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.

Data type conversion

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 whether to convert them to Oracle data types. Your planned use for the database and your preferred upgrade method have an impact on this decision.

- Converting data types to Oracle data types
- Upgrading data types as TimesTen data types

Converting data types to Oracle data types

Note: If you intend to use your database with TimesTen IMDB Cache, you must convert your data types to Oracle data types. However, you cannot perform an online upgrade using replication in this case, as data types may only replicate to like data types.

This is not an issue if you are upgrading from TimesTen 7.0 or later and your data types have been converted to Oracle data types.

To convert the data types from a release before TimesTen 7.0 to Oracle data types, you must use the -convertTypesToOra option for ttMigrate 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 data types:

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

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

Note: Because the Oracle 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 data types before deploying them with the new release of TimesTen.

Upgrading data types as TimesTen data types

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

If you choose to upgrade 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 attribute for databases is 0, which indicates Oracle type mode. In Oracle type mode, a data type such as CHAR has the semantics of the Oracle 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 of 1 before restoring the database with ttMigrate as part of the upgrade procedure.

Database character set conversion

A character set must be specified for each TimesTen database using the DSN attribute DatabaseCharacterSet. In some cases, you may need to change the configured database character set as part of the upgrade process. There are two different cases in which a database character set conversion are required:

- You have specified the database character set as TIMESTEN8 to upgrade your database 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. See "Converting from the TIMESTEN8 character set" on page 3-6.

- You must change your database's character set from the one that you originally specified to a new one that fits your requirements more closely. See "Converting from a character set other than TIMESTEN8", following shortly.

Converting from the TIMESTEN8 character set

You may use ttMigrate to convert a database from TIMESTEN8 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 DSN=salesdata salesdata.mig
   ```

2. Destroy the database:

   ```
   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 instead of TIMESTEN8, use the following line in your ODBCINI file:

   ```
   DatabaseCharacterSet=WE8ISO8859P1
   ```

4. Load the database from the file using ttMigrate with the -noCharsetConversion command-line option. This option 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
   DSN=salesdata salesdata.mig
   ```

Note: If you find that you have accidentally converted your database from TIMESTEN8 to the wrong character set, you can use the same procedure to convert your database to the correct character set without any accidental modification of the character data.

Converting from a character set other than TIMESTEN8

You may 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 DSN=salesdata salesdata.mig
   ```

2. Destroy the database:

   ```
   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 used by the DSN. For example:

```
ttMigrate -r DSN=salesdata salesdata.mig
```

> **Note:** 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.

Upgrade modes

TimesTen enables you to perform the following types of upgrades:

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

Patch releases

TimesTen databases from different patch releases of the same major release are structurally equivalent or identical. For example, when upgrading from release 11.2.2.0.0 to 11.2.2.1.0, you are not required to migrate your existing databases using the `ttMigrate` utility, unless you change the instance name and or daemon home directory. If you are changing the instance name or the daemon home directory, keep the old installation installed. Install the new installation. Then use the `ttAdoptStores` utility to move your data stores to the new installation. Review your data stores, then remove the old instance of TimesTen.

However, during the installation of any new release, you should disconnect your application and stop the TimesTen daemon. If you have not explicitly performed these steps, the TimesTen daemon process from the prior release is stopped, effectively disconnecting all applications from the database. 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.

In-place upgrades

In-place upgrades are available (on UNIX systems only) for moving to a new patch release of TimesTen, such as moving from the first patch release of 11.2.2, release 11.2.2.1.0, to the second patch release of 11.2.2, release 11.2.2.2.0.

In a TimesTen in-place upgrade, software libraries and binaries are updated in place while the TimesTen directory structure and database files remain unchanged.

During the in-place upgrade process, all applications must disconnect from databases in the TimesTen instance being upgraded. This type of upgrade enables you to preserve the existing databases in the TimesTen instance without using the TimesTen backup and migration utilities.

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

Offline upgrades

During the time required to perform an offline upgrade, the database is not available to applications. Offline upgrades usually require enough disk space for an extra copy of the upgraded database.

Offline upgrades are used to do the following:

- Move to a new major or patch release of TimesTen.
- Move to a different directory or computer.
- Reduce database size.
- Move between 32-bit and 64-bit databases.

Offline upgrades require all applications to disconnect from the database during the upgrade procedure. The database must also be unloaded from shared memory. Offline upgrades require you to use TimesTen's `ttMigrate` or `ttBackup` utilities. (See "`ttMigrate`" and "`ttBackup`" in *Oracle TimesTen In-Memory Database Reference*.)

See "Performing an offline upgrade" on page 3-11 and "Performing an offline TimesTen upgrade when using Oracle Clusterware" on page 3-31.

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 other one is being upgraded. When the upgrade is finished, any updates that have been made on the active database are transmitted immediately to the upgraded database, and your applications can then be switched to the upgraded database with no data loss and no down time. For more information, see "Performing an online upgrade with replication" on page 3-16.

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 to be upgraded are required, you must have available twice the memory and disk space that the database usually requires, if performing the upgrade on a single system.

Note: Replication is not supported between 32-bit and 64-bit databases. Also, tables migrated using the `ttMigrate -inline` command should not be replicated with tables where the option is not supported, as inline columns cannot be replicated with not inline columns.

In addition to the reference above, also refer to the following topics related to replication:

- "Performing an online upgrade with active standby pair replication" on page 3-23
- "Performing an online TimesTen upgrade when using Oracle Clusterware" on page 3-32
- "Performing an upgrade when using parallel replication" on page 3-42
- "Record of upgrades with replication configured" on page 3-43

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 that has been upgraded to the new release. See "Performing a Client/Server online upgrade from TimesTen release 6.0 and above" on page 3-44.

The Client/Server online upgrade process minimizes, but does not eliminate, the interruption of client application access to the database being upgraded. To maintain nearly continuous availability of a database to all clients, you may use the approach outlined in "Online upgrades with replication" on page 3-8. This procedure keeps an identical copy of the database available to the old release of TimesTen Server, while you upgrade the original to the new release. Once the upgraded 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 UNIX systems only), which does not require exporting its databases to an external format. Prior to 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:

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

All steps must be completed for each database in the TimesTen instance.

Unloading a database from memory

A TimesTen database remains loaded in shared memory as long as any applications or TimesTen agents, such as the cache agent or replication agent, is connected to it. A database may also be kept in shared memory if the RAM policy of the database has been modified 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.

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 IMDB cache and belongs to the TimesTen instance being upgraded. (The example does not consider replication.)

   ```
   ttAdmin -cacheStop cachedb
   ```

4. If the RAM policy is set to manual, unload the database from memory using ttAdmin. This example references a database upgradedb that belongs to the TimesTen instance being upgraded (and does not consider replication or cache):

   ```
   ttAdmin -ramUnload upgradedb
   ```

   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
   ```

   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. For example:

   ```
   ttAdmin -ramGrace 0 upgradedb
   ```

5. Use the ttStatus utility to verify that the databases belonging to the TimesTen instance being upgraded have been unloaded from memory. (See "ttStatus" in *Oracle TimesTen In-Memory Database Reference*.)

Also see the next section, "Reloading a database into memory".

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

Performing the upgrade

When doing an in-place upgrade (applicable on UNIX systems only), you must 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 1-16.

Reloading a database into memory

After unloading the databases from memory (as shown in the preceding section, "Unloading a database from memory") 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.

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
```

To restart IMDB cache (assuming cachedb uses cache and was in the TimesTen instance being upgraded):

```
ttAdmin -cacheStart cachedb
```

To reload the database into memory (assuming RAM policy was set to `manual` for the database `upgradedb`, which was in the TimesTen instance being upgraded):

```
ttAdmin -ramLoad upgradedb
```

If you changed the RAM policy from `always` to `manual` when you unloaded the database from memory, change it back to `always`. For example:

```
ttAdmin -ramPolicy always upgradedb
```

If the RAM policy was `inUse` before you unloaded the database from memory and you changed the grace period to 0, change it back to its original setting. For example:

```
ttAdmin -ramGrace 300 upgradedb
```

Performing an offline upgrade

You can do an offline upgrade by exporting the database into an external file and then importing the database with the desired changes. These update 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 3-16.

> **Note:** If the database to be upgraded is replicated, you must use `ttMigrate` to move the database between releases.

For simple upgrades that do not require any changes to the size or structure of the database, such as moving between patch releases, you can use the `ttBackup` and `ttRestore` utilities. For upgrades that require structural changes to the database, such as major release upgrades, you must use the `ttMigrate` utility. The `ttMigrate` utility exports the database in a release-neutral format that is more flexible, while the `ttBackup` utility exports an image copy of the database that is only compatible across patch releases. The `ttBackup` utility may be used to move the database to a different computer or directory. The `ttMigrate` utility must be used if you want to do the following:

- Move to a new major or patch release of TimesTen.
- Reduce database size.
- Move between 32-bit and 64-bit databases.

The general steps in an offline upgrade include:

1. Disconnect all applications from the database and unload the database from memory. For more information, see "Unloading a database from memory" on page 3-9.
2. Use either `ttMigrate` with the `-c` and `-relaxedUpgrade` options or `ttBackup` to back up the database.
3. Install the new release of TimesTen. For more information, see Chapter 1, "TimesTen Installation".
4. Use either `ttMigrate` with the `-r` and `-relaxedUpgrade` options or `ttRestore` to restore the backed up database to the new TimesTen release.
5. Reconnect applications to the upgraded database.

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

Moving to a different directory

The TimesTen daemon identifies a database by the full path name of the database's 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. When you have verified that the database functions properly in the new location, free up the disk space by using `ttDestroy` to remove the old database.

For example, to move a database from `/old/SalesData/sales` with a database name `SalesData` (`"DSN=SalesData"`) to `/new/SalesData/sales` with database name `NewSalesData` (`"DSN=NewSalesData"`), using the `/tmp/dump` directory for temporary storage, use the following commands:

1. `mkdir /tmp/dump`
2. `ttBackup -dir /tmp/dump -fname salesdata "DSN=SalesData"`
3. Create a DSN definition for the `NewSalesData` database and specify the new database path: `/new/SalesData/sales/NewSalesData`.
4. `ttRestore -dir /tmp/dump -fname salesdata "DSN=NewSalesData"`

   (Verify that `NewSalesData` is operational.)
5. `rm -r /tmp/dump`
6. `ttDestroy /old/SalesData/sales/SalesData`
7. Remove the DSN definition for the `SalesData` database.

Note: You must reconfigure replication if the moved database has been configured to replicate.

Moving to a different computer

You can also use the `ttBackup` and `ttRestore` utilities to move a database between two computers that have the same CPU architecture and are running the same operating system.

Note: Before moving a replicated database to another computer, you should be experienced with TimesTen replication. We highly advise calling TimesTen customer support for help with this procedure.

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

1. Back up the database on the original system using `ttBackup`.
2. Move the backup to the new system.

3. Reconfigure the replication scheme on any databases the replicate to the moved database to specify the new host computer. (See "Reversing the roles of the master databases" in *Oracle TimesTen In-Memory Database Replication Guide* for information relevant in specifying a host in a replication scheme.)
4. Restore the backup using `ttRestore`.

The following example uses the `-o` flag of `ttBackup` to use standard output for the backup. Using the `-o` flag, the backup is stored in a single file, which is easily copied over the network to the other system. Once you have copied the database to the other system, you must create a data source name for access to the new database copy.

To move a database from `/ds/Sales/Data` with database name `salesdata` on the source system to `/data/Sales/View` with database name `salesview` on the destination system, use the following commands.

Step	On the source system	On the destination system
1.	`ttBackup -o "DSN=SalesData" >/tmp/salesbackup`	
2.	`ftp /tmp/salesbackup` to the destination system as `/tmp/salesbackup` **Note**: Use the `ftp` command in binary mode.	
3.		`ttRestore -i"DSN=SalesView" </tmp/salesbackup`
4.		`rm /tmp/salesbackup`
5.	`rm /tmp/salesbackup`	

Reducing database size

Once a database has been defined with a particular size for the permanent partition (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 database's permanent partition size embedded in it.

To reduce the allocated size of the permanent partition of a database, save a copy using the `ttMigrate` utility with the `-relaxedUpgrade` option. Then re-create the database with a smaller permanent partition size and restore the data.

> **Note:** The permanent partition size of a database cannot be reduced below the size that is actually required by the data currently stored in the database. This value can be determined by querying the `perm_in_use_size` column of the table `sys.monitor`.

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

1. Back up the old database, as follows.

   ```
   ttMigrate -c -relaxedUpgrade
   ```

2. Create a new DSN definition for the new copy of the database with a smaller `PermSize` value.
3. Restore the backup, as follows.

```
ttMigrate -r -relaxedUpgrade
```

Note: To modify the original DSN rather than create a new one in Step 2, then you must destroy the original database files using the `ttDestroy` utility before restoring from the backup.

Here are the steps to reduce a database's allocated size from 400 MB to 100 MB. The database is in `/ds/Sales/Data` and has the DSN `salesdata`.

1. `ttMigrate -c -relaxedUpgrade DSN=salesdata /tmp/salesbackup`
2. `ttDestroy salesdata`
3. Update the DSN `salesdata` to have a size of 100 MB.
4. `ttMigrate -r -relaxedUpgrade "DSN=salesdata;AutoCreate\=1" /tmp/salesbackup`

Note: The temporary partition size of a database may be changed by simply modifying the `TempSize` attribute of the DSN, unloading the database from memory, and then reconnecting to it. See "Unloading a database from memory" on page 3-9 for the steps to unload a database from memory.

Moving between 32-bit and 64-bit databases

The internal format of a 32-bit TimesTen database differs from that of a 64-bit database. To convert a 32-bit database to a 64-bit database:

1. Export the 32-bit database using the TimesTen 32-bit `ttMigrate` utility with the `-relaxedUpgrade` option.
2. Create a Data Source Name (DSN) for the 64-bit 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*.
3. Import the file created in Step 1 into the 64-bit database DSN using the 64-bit `ttMigrate` utility with the `-relaxedUpgrade` option.

For example, suppose that the 32-bit database DSN is `salesdata32` while the 64-bit database DSN is `salesdata64`. If a 32-bit instance of TimesTen is installed in `/opt/TimesTen/giraffe32` and a 64-bit instance is installed in `/opt/TimesTen/giraffe64`, the required steps are:

1. `/opt/TimesTen/giraffe32/bin/ttMigrate -c -relaxedUpgrade DSN=salesdata32/tmp/salesbackup`
2. `/opt/TimesTen/giraffe64/bin/ttMigrate -r -relaxedUpgrade "DSN=salesdata64;AutoCreate=1"/tmp/salesbackup`

Note: TimesTen does not support replication between 32-bit and 64-bit databases.

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 6.0 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. The procedure for this upgrade is similar to the steps outlined in "Moving between 32-bit and 64-bit databases" on page 3-14.

If you must upgrade two or more databases that are replicating to each other, you must perform a few extra steps to ensure that replication continues to operate during and after the upgrade. For example, to migrate two replicating databases, master1 on host computer masterhost and subscriber1 on host computer subscriberhost, from TimesTen release 6.0 to TimesTen release 11.2.2, perform the steps that follow.

Note: TimesTen 11.2.1 introduced the cache grid feature. By default, this feature is enabled and requires additional configuration before cache groups can be created on a database. If you are upgrading a database 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 the *Oracle In-Memory Database Cache User's Guide*.

1. Configure the replication scheme on both databases to use static TCP/IP ports for replication. This is necessary because the intermediate result of this procedure is that replication occurs between two different releases of TimesTen. Each release does not necessarily know how to find the main daemon of the other to have the replication ports assigned dynamically. See "Port assignments" in the *Oracle TimesTen In-Memory Database Replication Guide* for more information.

2. On the computer masterhost, use the ttAdmin utility of the 6.0 release to stop the replication daemon on the database:

   ```
   ttAdmin -repStop master1
   ```

3. Next, use the ttMigrate utility of the 6.0 release with the -c option to back up database master1 to a binary file:

   ```
   ttMigrate -c DSN=master1 master1.bak
   ```

4. Use the ttDestroy utility of the 6.0 release to destroy database master1, where the database's files are located in the directory *data_store_path*:

   ```
   ttDestroy /data_store_path/master1
   ```

5. Use the ttMigrate utility of the 11.2.2 release with the -r option to restore database master1 from the binary file. Restoring the database automatically upgrades it from release 6.0 to release 11.2.2. If you are restoring a very large database, you should use the -C option to tell ttMigrate to perform a checkpoint operation on the database periodically. This saves time if the restore fails at some point before the operation has been completed. For more information, see "Checkpoint operations" in the *Oracle TimesTen In-Memory Database Operations Guide*.

   ```
   ttMigrate -r -C 20 DSN=master1 master1.bak
   ```

6. Use the ttAdmin utility of the 11.2.2 release to start the replication daemon:

   ```
   ttAdmin -repStart master1
   ```

Replication is now occurring between the database `master1` on release 11.2.2 and the database `subscriber1` on release 6.0.

Now upgrade database `subscriber1` from release 6.0 to release 11.2.2. Perform the steps that follow.

7. On the computer `subscriberhost`, use the `ttAdmin` utility of the 6.0 release to stop the replication daemon:

   ```
   ttAdmin -repStop subscriber1
   ```

8. Use the `ttDestroy` utility of the 6.0 release to destroy database `subscriber1`, where the database files are located in the directory *`data_store_path`*:

   ```
   ttDestroy data_store_path/subscriber1
   ```

9. If you are upgrading from a release preceding 11.2.2, you must create a user with `ADMIN` privileges to duplicate a database using replication. 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;
   ```

10. Use the `ttRepAdmin` utility of the 11.2.2 release with the `-duplicate` option to duplicate database `subscriber1` from database `master1` using replication.

    ```
    ttRepAdmin -duplicate -from master1 -host masterhost -uid pat -pwd patpwd
     -setMasterRepStart subscriber1
    ```

11. Use the `ttAdmin` utility of the 11.2.2 release to start the replication daemon:

    ```
    ttAdmin -repStart subscriber1
    ```

The databases are now upgraded and replicating to each other.

Performing an online upgrade with replication

In "Performing an offline upgrade" on page 3-11 we showed 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. You can do an online upgrade when moving between major TimesTen releases. If moving to a patch release, you may instead perform an in-place or offline upgrade.

Normally, 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" *Oracle TimesTen In-Memory Database Replication Guide*.

Note: 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 the 5.1 release of TimesTen and a database created in the 7.0 release of TimesTen is not supported. However, if one database was created in the 5.1 release, and the peer database was created in the 5.1 release and then upgraded to the 7.0 release, replication between them is supported.

Note: For security reasons, replication is not normally allowed between databases in TimesTen 7.0 and databases in releases preceding TimesTen 7.0. To perform an online upgrade with replication, you must start the main TimesTen 7.0 or later daemon with the `-insecure-backwards-compat` option.

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

- Overview
- Limitations
- Requirements
- Online upgrade example

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 upgraded database.

The general steps in an online upgrade include:

1. Disconnect all applications from the database being upgraded.
2. Shut down replication on the system being upgraded.
3. Back up the database residing on the system 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 on the system being upgraded using the newer release's `ttMigrate` with the `-r` option.
6. Reconnect all applications to the upgraded database.
7. Restart replication on the upgraded system.

Note: After ttMigrate has been used, all autorefresh cache groups in an upgraded 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 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 upgraded database when replication is restarted. When the replicated updates have been completely applied, the applications may be reconnected to the upgraded database.

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 an 11.2.1 database that has the connection attribute ReplicationApplyOrdering set to 0, or from an 11.2.1 or 11.2.2 database that has ReplicationParallelism set to <2, requires that you re-create the database, even if the version is the same or only the minor version has changed.

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 Data Source Name for the upgraded 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.	

Step	Upgrade system	Active system
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.	
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 upgrade database.	

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

Limitations

Online upgrades can be performed only on 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 being upgraded. 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`:

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

Online upgrade example

This section describes how to perform an online upgrade of 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 a database 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 the *Oracle In-Memory Database Cache User's Guide*.

We'll refer to the two TimesTen systems being upgraded as the *upgrade* system, on which the TimesTen instance and the database are upgraded, and the *active* system, which remains operational and connected to the application for the duration of the upgrade. After this 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:
	`Command> call ttRepStop;`
	`Command> ALTER REPLICATION repscheme ALTER STORE upgrade ON upgradehost SET PORT 40000 ALTER STORE active ON activehost SET PORT 40001;`
	`Command> call ttRepStart;`
		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> ALTER REPLICATION repscheme ALTER STORE upgrade ON upgradehost SET PORT 40000 ALTER STORE active ON activehost SET PORT 40001;`
		`Command> call ttRepStart;`

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 -connStr DSN=active -receiver -name upgrade -state pause` 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*.

Step	Upgrade system	Active system
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 -connStr DSN=upgrade`
	`-receiver -name active`
	`-state stop`
	This step prevents active from accumulating updates to send to upgrade and resets some of the 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 DSN=upgrade`
	`/backup/upgrade.dat`	

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 -connStr DSN=active`
		`-receiver -name upgrade`
		`-start start`

Step	Upgrade system	Active system

9.	Use `ttMigrate` to load the backup created in Step 6. into the database `upgrade` for the new release of TimesTen:
	`ttMigrate -r`
	` "DSN=upgrade;AutoCreate=0"`
	` /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 -connStr DSN=upgrade`
	`   -receiver -name active -reset`
	`ttRepAdmin -connStr DSN=upgrade`
	`   -receiver -name active`
	`   -state stop`
	`sleep 10`
	`ttRepAdmin -connStr DSN=upgrade`
	`   -receiver -name active`
	`   -state start`
	`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`	
12.	Verify that the database `upgrade` is receiving updates from `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 the `active` to `upgrade`.

Step	Upgrade system	Active system
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 database `active` is ready to be upgraded. **Note**: You may choose to delay upgrading `active` to the new TimesTen release until sufficient testing has been performed with the database `upgrade` in the new TimesTen release.

Performing an online upgrade with active standby pair replication

You generally use active standby pair replication to provide high availability of your data to your applications. It is possible to maintain continuous availability of your data, even during a upgrade of TimesTen, your operating system, or your system hardware, by performing an online upgrade with active standby pair replication. This section describes the following procedures:

- Upgrading the databases in an active standby pair with no cache groups
- Upgrading the databases in an active standby pair with cache groups

Upgrading the databases in an active standby pair with no cache groups

This section includes the following topics:

- Upgrades for TimesTen patch releases on the standby master database and subscriber databases
- Upgrades for TimesTen patch releases on the active master database
- Upgrades for major TimesTen releases, application software and hardware

Upgrades for TimesTen patch releases on the standby master database and subscriber databases

To upgrade to a TimesTen patch release on the standby master database and subscriber databases, complete the following tasks on each database:

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 TimesTen patch. 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
   ```

Upgrades for TimesTen patch releases on the active master database

To upgrade the active master database, you must first reverse the roles of the active and standby master databases, and 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. 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`).
7. Upgrade 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 upgraded database using the `ttRepStart` built-in procedure or the `ttAdmin` utility:

   ```
   ttAdmin -repStart master2
   ```

9. To make the newly upgraded database the active master database again, see "Reversing the roles of the active and standby master databases" in *Oracle TimesTen In-Memory Database Replication Guide*.

Upgrades for major TimesTen releases, application software and hardware

When you upgrade an active standby pair across major releases 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 prepare for the upgrade by following these steps:

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*.

When you are ready to upgrade the active standby pair, the first database that you must upgrade is the standby master. While this node is being upgraded, there is no standby master database, and updates on the active master database are propagated directly to the subscriber databases.

1. Tell 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. For example, to stop the replication agent for the standby master database master2:

   ```
   ttAdmin -repStop master2
   ```

3. Upgrade the node where the standby master database resides. See "Performing an offline upgrade" on page 3-11.

4. Start the replication agent on the standby master database using the ttRepStart built-in procedure or the ttAdmin utility:

   ```
   ttAdmin -repStart master2
   ```

   When the upgraded standby master database has become synchronized with the active master database, the upgraded standby master database moves from the RECOVERING state to the STANDBY state. The upgraded 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 on the standby master database:

   ```
   call ttRepStateGet;
   ```

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

6. 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 );
   ```

7. 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
   ```

8. 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' );
   ```

9. Tell 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' );
   ```

10. Destroy the former active master database using the ttDestroy utility:

    ```
    ttDestroy master1
    ```

11. Perform the upgrade on the node where the master database was destroyed. There is no need to do anything to upgrade the database itself, since it was destroyed.
12. If you are upgrading from a release preceding 11.2.1, you must create a user with ADMIN privileges on the current active master database to duplicate that database. 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;
    ```

13. Duplicate the new standby master database from the current 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
    ```

14. Start the replication agent on the new standby master database using the ttRepStart built-in procedure or the ttAdmin utility:

    ```
    ttAdmin -repStart master1
    ```

15. 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
    ```

16. Destroy the subscriber database using the ttDestroy utility:

    ```
    ttDestroy subscriber1
    ```

17. Perform the upgrade on the node where the subscriber database was destroyed.
18. Duplicate the subscriber database from the standby master database using the ttRepAdmin utility, as follows.

```
ttRepAdmin -duplicate -from master1 -host master1host -uid pat -pwd patpwd
 -setMasterRepStart subscriber1
```

19. Start the replication agent for the duplicated subscriber database using the `ttRepStart` built-in procedure or the `ttAdmin` utility:

    ```
    ttAdmin -repStart subscriber1
    ```

20. Repeat Step 15 through Step 19 for each other subscriber database.

Upgrading the databases in an active standby pair with cache groups

This section includes the following topics:

- Upgrades for TimesTen patch releases on the standby master database and subscriber databases
- Upgrades for TimesTen patch releases on the active master database
- Upgrades for major TimesTen releases, application software and hardware

Upgrades for TimesTen patch releases on the standby master database and subscriber databases

To upgrade to a TimesTen patch release on the standby master database and subscriber databases, complete the following tasks on each database.

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 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 using the `ttCacheStart` built-in procedure or the `ttAdmin` utility:

    ```
    ttAdmin -cacheStop master2
    ```

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

    ```
    ttAdmin -repStart master2
    ```

Upgrades for TimesTen patch releases on the active master database

To upgrade the active master database, you must first reverse the roles of the active and standby master databases, and 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 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 upgraded database using the `ttCacheStart` built-in procedure or the `ttAdmin` utility:

   ```
   ttAdmin -cacheStart master1
   ```

10. Restart replication on the upgraded database using the `ttRepStart` built-in procedure or the `ttAdmin` utility:

    ```
    ttAdmin -repStart master1
    ```

11. To make the newly upgraded database the active master database again, see "Reversing the roles of the active and standby master databases" in *Oracle TimesTen In-Memory Database Replication Guide*.

Upgrades for major TimesTen releases, application software and hardware

When you upgrade an active standby pair across major releases 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 prepare for the upgrade by following these steps:

1. Tell 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 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
   ```

3. 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;
   ```

4. Destroy the standby master database and all of the subscribers using the ttDestroy utility. For example, to destroy the database subscriber1:

   ```
   ttDestroy subscriber1
   ```

5. 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*.

Begin major upgrades on the node with the standby master database. While this node is being upgraded, there is no standby master database, and updates on the active master database are propagated directly to the subscriber databases.

> **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 a database containing cache groups from a release preceding 11.2.1, you should set the CacheGridEnable attribute to 0 in each DSN definition before beginning, as discussed in *Oracle In-Memory Database Cache User's Guide*.

1. Tell 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. For example, to stop the replication agent for the standby master database master2:

   ```
   ttAdmin -repStop master2
   ```

3. Stop the cache agent on the standby master database using the ttCacheStop built-in procedure or the ttAdmin utility:

   ```
   ttAdmin -cacheStop master2
   ```

4. Upgrade the node where the standby master database resides. See "Performing an offline upgrade" on page 3-11.

5. Start the cache agent on the standby master database using the ttCacheStart built-in procedure or the ttAdmin utility:

   ```
   ttAdmin -cacheStart master2
   ```

6. Start the replication agent on the standby master database using the ttRepStart built-in procedure or the ttAdmin utility:

   ```
   ttAdmin -repStart master2
   ```

   When the upgraded standby master database has become synchronized with the active master database, the upgraded standby master database moves from the RECOVERING state to the STANDBY state. The upgraded 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 on the standby master database:

   ```
   call ttRepStateGet;
   ```

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

8. 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 );
   ```

9. 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, use the following command.

   ```
   ttAdmin -repStop master1
   ```

10. Stop the cache agent on the active master database using the ttCacheStop built-in procedure or the ttAdmin utility:

    ```
    ttAdmin -cacheStop 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. Tell 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. Perform the upgrade on the node where the master database was destroyed. There is no need to do anything to upgrade the database itself, since it was destroyed.

15. If you are upgrading from a release preceding 11.2.1, you must create a user with ADMIN privileges on the current active master database to duplicate that database. 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. Duplicate the new standby master database from the current active master database using the ttRepAdmin utility. For example, to duplicate the database master2 on the host master2host to the database master1, where the cache user ID is terry and the cache password is terrypwd, use the following on the host containing the database master1:

```
ttRepAdmin -duplicate -from master2 -host master2host -uid pat -pwd patpwd
 -cacheuid terry -cachepwd terrypwd -keepCG -setMasterRepStart
 "DSN=master1;UID=;PWD=;PWDCrypt="
```

> **Note:** Only the instance administrator can create a database, but the DSN for a database containing cache groups is often configured with any of the UID, PWD, and PWDCrypt attributes. To allow the ttRepAdmin utility to create the database when duplicating, you may specify blank values for any of UID, PWD, or PWDCrypt in the connection string to connect as the instance administrator.

17. Start the cache agent on the new standby master database using the ttCacheStart built-in procedure or the ttAdmin utility:

```
ttAdmin -cacheStart master1
```

18. Start the replication agent on the new standby master database using the ttRepStart built-in procedure or the ttAdmin utility:

```
ttAdmin -repStart master1
```

19. 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
```

20. Destroy the subscriber database using the ttDestroy utility:

```
ttDestroy subscriber1
```

21. Perform the upgrade on the node where the subscriber database was destroyed.

22. Duplicate the subscriber database from the standby master database using the ttRepAdmin utility:

```
ttRepAdmin -duplicate -from master1 -host master1host -uid scott -pwd tiger
 -noKeepCG -setMasterRepStart subscriber1
```

23. Start the replication agent for the duplicated subscriber database using the ttRepStart built-in procedure or the ttAdmin utility:

```
ttAdmin -repStart subscriber1
```

24. Repeat Step 15 through Step 19 for each other subscriber database.

Performing an offline TimesTen upgrade when using Oracle Clusterware

When using TimesTen with Oracle Clusterware, perform these tasks for an offline upgrade of TimesTen. You can also upgrade Oracle Clusterware while upgrading

TimesTen. (Also see the next section, "Performing an online TimesTen upgrade when using Oracle Clusterware".)

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 on the host. This removes the host from the cluster and stops the TimesTen daemon:

   ```
   ttCWAdmin -shutdown -hosts localhost
   ```

4. Upgrade TimesTen on the desired hosts. Every node in the cluster must have TimesTen from the same major release. If you are upgrading between major releases, use the `ttMigrate` utility.

5. Upgrade Oracle Clusterware if desired. See *Oracle Clusterware Administration and Deployment Guide*.

6. If you have upgraded Oracle Clusterware, use the `ttmodinstall` utility 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).

7. Start the TimesTen cluster agent. This includes the host in the cluster and starts the TimesTen daemon:

   ```
   ttCWAdmin -init
   ```

8. Create the active standby pair replication scheme:

   ```
   ttCWAdmin -create -dsn advancedDSN
   ```

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 a minor online rolling in-place upgrade for TimesTen in a configuration where active standby pairs are managed by Oracle Clusterware. (Also see the preceding section, "Performing an offline TimesTen upgrade when using Oracle Clusterware".)

The following topics are covered:

- Supported configurations
- Restrictions and assumptions
- Upgrade tasks for one active standby pair
- Upgrade multiple active standby pairs on many pairs of hosts
- Upgrade multiple active standby pairs on a pair of hosts
- Sample in-place upgrade

Note: These instructions are for minor upgrades only, from one TimesTen 11.2.2.*x.x* version to another 11.2.2.*x.x* version.

Supported configurations

The following basic configurations are supported for online rolling in-place upgrades for TimesTen. In all cases, the hosts are managed by Oracle Clusterware.

- 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 minor upgrades only.)
- 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 as long as there are at least two hosts in the cluster.

Note the following restrictions 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.
- Online rolling in-place grades are not supported on Windows in this TimesTen release.

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 of 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 are running a TimesTen 11.2.2.*x.x* release.
   - The active standby pair is on two 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 Oracle Database are working.
     - Refreshes from 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:

   ```
   crs_stat -c standbyHostName
   ```

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.

   ```
   crs_stop TT_Master_upgrade2_terry_myDSN_1
   crs_stop TT_Daemon_upgrade2_terry_host2
   crs_stop TT_Agent_upgrade2_terry_host2
   ```

3. Set the TimesTen RAM policy to `manual` and stop the TimesTen main daemon.

   ```
   ttAdmin -ramPolicy manual -dsn myDSN
   ttDaemonAdmin -stop
   ```

   If the `ttDaemonAdmin -stop` command gives error 10028, retry the command.

Perform an in-place upgrade of the standby database

Complete these steps for an in-place upgrade of 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`
   - `ttcrsagent.options`
   - `tnsnames.ora`

   See "Sample in-place upgrade" on page 3-39.

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. Set the RAM policy for the standby database to `inUse`.

   ```
   ttAdmin -ramPolicy inUse -dsn myDSN
   ```

3. Start the Oracle Clusterware Master process for the standby database.

```
crs_start 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 crs_stat command to obtain the names of the Master, Daemon and Agent processes on the host of the new standby database (host1 in this example):

```
crs_stat -c host1
```

Enter commands such as those in "Shut down the standby database" on page 3-34. Use the appropriate instance name, instance administrator, DSN, and host name. For example:

```
crs_stop TT_Master_upgrade2_terry_MYDSN_0
crs_stop TT_Daemon_upgrade2_terry_host1
crs_stop TT_Agent_upgrade2_terry_host1
ttAdmin -ramPolicy manual -dsn myDSN
ttDaemonAdmin -stop
```

Perform an in-place upgrade of the new standby database

Enter commands such as those in "Perform an in-place upgrade of the standby database" on page 3-35.

Start the new standby database

See "Start the standby database" on page 3-35. Use the Master process name obtained by the crs_stat -c command.

```
ttCWAdmin -init -hosts localhost
ttAdmin -ramPolicy inUse -dsn myDSN
crs_start TT_Master_upgrade2_terry_MYDSN_0
```

Upgrade multiple active standby pairs on many pairs of hosts

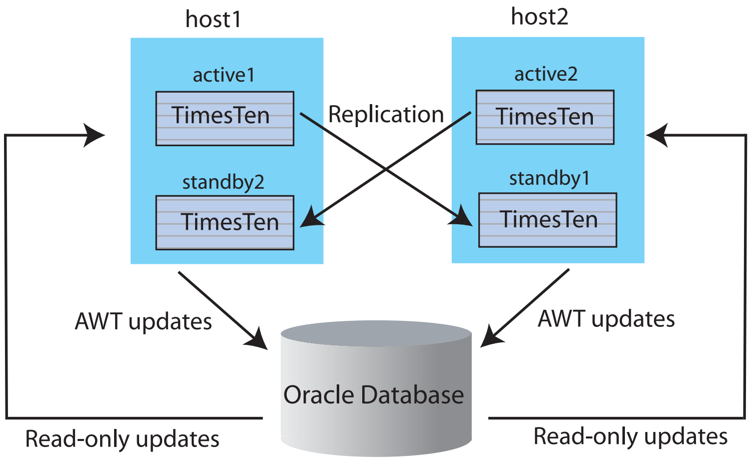
The process to upgrade a single active standby pair on two hosts is the same as the process to upgrade multiple active standby pairs on multiple pairs of hosts. The best practice is to upgrade the active standby pairs one at a time.

Use the ttCWAdmin -status command to determine the state of the databases managed by Oracle Clusterware.

Upgrade multiple active standby pairs on a pair of hosts

An active standby pair managed by Oracle Clusterware must use at least two hosts, one for the active database and one for the standby database, but more than one active standby pair can be configured on a pair of hosts. 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 "Sample configuration files: multiple active standby pairs on the same pair of hosts" on page 3-37 for sample `sys.odbc.ini` entries and a sample `cluster.oracle.ini` file for this kind of configuration.

The in-place rolling upgrade process is similar, with these additional modifications:

- Switch all standby databases to one host and do the upgrade for the entire "standby" host. In-place upgrades affect the entire TimesTen installation and associated databases on one host. Use the `ttCWAdmin -switch -dsn` *`DSN`* command to move the standby databases to one host.
- Verify that the standby databases are running on the desired host. Use the `ttCWAdmin -status -dsn` *`DSN`* command and the `ttCWAdmin -status` command.
- Modify the Oracle Clusterware `stop` commands to stop all Master processes on the host where the standby databases reside.
- Modify the Oracle Clusterware `start` commands to start all Master processes on the host where the standby databases reside.
- Modify the `ttAdmin ramPolicy` command to change the policy for all databases on the host where the standby databases reside.

The following subsections contain related samples.

- Sample configuration files: multiple active standby pairs on the same pair of hosts
- Sample scripts: stopping and starting multiple standby processes on one host

Sample configuration files: multiple active standby pairs on the same pair of hosts

The following are sample `sys.odbc.ini` entries for `databasea` and `databaseb`:

```
[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
```

Note that 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 hosts names specified for the `MasterHost` attribute.

Sample scripts: stopping and starting multiple standby processes on one host

Use the `crs_stat -c` *`hostName`* command to obtain the names of the Master, Daemon and Agent processes for the standby host.

The following script is an example of a "stop standby" script for multiple databases on the same host that is managed by Oracle Clusterware. The TimesTen instance name is upgrade2. The instance administrator is terry. The host is host2. There are two standby databases: databasea and databaseb.

```
crs_stop TT_Master_upgrade2_terry_DATABASEA_0
crs_stop TT_Master_upgrade2_terry_DATABASEB_1
crs_stop TT_Daemon_upgrade2_terry_HOST2
crs_stop TT_Agent_upgrade2_terry_HOST2
ttAdmin -ramPolicy manual -dsn databasea
ttAdmin -ramPolicy manual -dsn databaseb
ttDaemonAdmin -stop
```

The following script is an example of a "start standby" script for the same configuration.

```
ttCWAdmin -init -hosts localhost
ttAdmin -ramPolicy inUse -dsn databasea
ttAdmin -ramPolicy inUse -dsn databaseb
crs_start TT_Master_upgrade2_terry_DATABASEA_0
crs_start TT_Master_upgrade2_terry_DATABASEB_1
```

Sample in-place upgrade

This section provides a sample in-place upgrade.

```
$ ./setup.sh

There are 1 TimesTen instances installed locally :
1) upgrade2 (TimesTen11.2.2.3.0)

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          : upgrade2
Product Installed      : TimesTen11.2.2.3.0
Installation Directory : /scratch/terry/TimesTen/upgrade2
BitLevel               : 64
Component Installed    : Client/Server and DataManager
Daemon Port            : 12345


NOTE: Upgrading will remove the selected instance and re-install into
      the same directory. You will have the option to retain configuration
      files in : /scratch/terry/TimesTen/upgrade2/info

Would you like to upgrade this instance? [ yes ]
** WARNING **

   The uninstallation has been executed by a non-root user.
   If the TimesTen daemon startup scripts were installed,
   you must run <install_dir>/bin/setuproot -uninstall
   to remove them. If you proceed with this uninstallation, you
   will have to remove the startup scripts manually.
   Would you like to proceed with the uninstallation? [ yes ]
   The TimesTen daemon (timestend) is running
   Terminating subprocesses.

NOTE: /scratch/terry/TimesTen/upgrade2/info contains information related to the
data
   stores that have been created with this release. If you remove
   /scratch/terry/TimesTen/upgrade2/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
 /scratch/terry/TimesTen/upgrade2/info? [ no ]
   Would you also like to remove all files in
 /scratch/terry/TimesTen/upgrade2/network/admin/samples? [ no ]
   /scratch/terry/TimesTen/upgrade2 Removed
(retained /scratch/terry/TimesTen/upgrade2/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/terry/TimesTen/upgrade2
Where would you like to create the daemon home directory?
[ /scratch/terry/TimesTen/upgrade2/info ]

The daemon logs will be located in /scratch/terry/TimesTen/upgrade2/info
Would you like to specify a different location for the daemon logs? [ no ]
Installing into /scratch/terry/TimesTen/upgrade2 ...
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 's999'? [ 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/terry/TimesTen/upgrade2/info? [ yes ] no

NOTE: The existing daemon options file has been retained. The default options
      file was written as
/scratch/terry/TimesTen/upgrade2/info/ttendaemon.options.upgrade2.
      TNS_ADMIN exists in your environment and is set to :
      /scratch/terry/TimesTen/upgrade2/network/admin/samples

Would you like to use this TNS_ADMIN setting for the In-Memory Database Cache?
```

```
[ yes ]

TNS_ADMIN will be set to /scratch/terry/TimesTen/upgrade2/network/admin/samples
You can change TNS_ADMIN later by running <install_dir>/bin/ttmodinstall.
Installing server components ...
What is the TCP/IP port number that you want the TimesTen Server to listen on?
[ 12346 ]
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/terry/TimesTen/upgrade2/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/terry/TimesTen/upgrade2/info/cluster.oracle.ini.tt1122'.


An existing sys.odbc.ini file has been detected in
/scratch/terry/TimesTen/upgrade2/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/terry/TimesTen/upgrade2/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/terry/TimesTen/upgrade2/info/sys.odbc.ini.upgrade2'.

Installing client components ...

Would you like to use TimesTen Replication with Oracle Clusterware? [ no ] yes

A Clusterware installation was detected in /opt/oracle/product/11.2.0/crs

Please provide the path to the Oracle Clusterware installation on this machine
 (s=skip)? [ /opt/oracle/product/11.2.0/crs ]
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? [ 12351 ]

Executing '/opt/oracle/product/11.2.0/crs/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 ] y
From the nodes above, please provide a list of nodes
that you would like to add (ex: 1,2 or skip)? [  ] 1,2,3,4

TimesTen Replication with Oracle Clusterware will be configured
for the following nodes :

host1
host2
host3
host4

Are you sure? [ yes ] y
Overwrite the existing TimesTen Clusterware options file? [ no ]
The new TimesTen Clusterware options file will be located here :
        /scratch/terry/TimesTen/upgrade2/info/ttcrsagent.options.tt1122.
NOTE: The TimesTen daemon startup/shutdown scripts have not been installed.

Run the 'setuproot' script :
        cd /scratch/terry/TimesTen/upgrade2/bin
        ./setuproot -install
This will move the TimesTen startup script into its appropriate location.

The startup script is currently located here :
  '/scratch/terry/TimesTen/upgrade2/startup/tt_upgrade2'.

The documentation was not installed.
To manually install the documentation, run the command 'setup.sh -installDoc'

The 11.2.2.4.0 Release Notes are located here :
  '/scratch/terry/TimesTen/upgrade2/README.html'

Starting the daemon ...
TimesTen Daemon startup OK.
End of TimesTen installation.
```

Performing an upgrade 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.

> **Note:** The values for the `ReplicationApplyOrdering` attribute have changed. Beginning in release 11.2.2.2.0, a value of 0 enables automatic parallel replication. Prior to release 11.2.2.2.0, a value of 0 disabled user-specified parallel replication.

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.

Parallel replication with an active standby pair

To upgrade an active standby pair without parallel replication enabled to release 11.2.2.3.0 or higher with the `ReplicationApplyOrdering` attribute set to the default value (0), you must use the steps for doing an active standby upgrade. See "Performing an online upgrade with active standby pair replication" on page 3-23.

Scenarios that require an offline upgrade

You must use an offline upgrade for these scenarios:

- 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).
- From automatic parallel replication to another automatic parallel replication with different number of tracks as indicated by the value of the `ReplicationParallelism` attribute.
- If you are moving between major and minor releases and are using asynchronous writethrough parallelism.
- 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 the section "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.

Considerations regarding parallel replication

An upgrade from a database with parallel replication to a release with no parallel replication is possible only if the master has not been running with more than one track. Once multiple transmitters have been in operation for a specified source database, replication to a database without parallel replication enabled is not possible.

You cannot replicate between a database that has the `ReplicationParallelism` attribute set to greater than 1 and that has different values for the `ReplicationApplyOrdering` attribute.

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 that the `ttMigrate -r` command is used. These tables provide a history of the database's 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 from TimesTen release 6.0 and above

TimesTen Server can talk directly to all user applications linked with TimesTen Client ODBC driver release 6.0 or higher. There are at least two 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 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 3-16 to keep a second copy of the database available during the migration of the first copy.

> **Note:** For security reasons, client/server communication is not normally allowed between TimesTen 7.0 and previous releases. To perform an online upgrade with Client/Server, you must start the main TimesTen 7.0 daemon with the `-insecure-backwards-compat` option.

Client/Server online upgrade

To perform an online upgrade of a TimesTen Client/Server system to a new major release 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 install 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 between 32-bit and 64-bit databases" on page 3-14 for an example of this procedure.
4. Start the new release of TimesTen Server. The client applications now have access to the upgraded databases.

> **Note:** Since both releases of TimesTen Server have been configured to listen on the same port in this procedure, if you want 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 3-16 to upgrade 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.

Index