Oracle® TimesTen In-Memory Database

Installation, Migration, and Upgrade Guide

Release 22.1
F35388-02
February 2023

ORACLE

Oracle TimesTen In-Memory Database Installation, Migration, and Upgrade Guide, Release 22.1

F35388-02

Contents

3 Using systemd to Manage a TimesTen Service

4 Installation and Management of TimesTen on Windows

5 Client-only Installations and Instances

6 Back Up, Restore, and Migrate Data in TimesTen Classic

7 Upgrades in TimesTen Classic

Index

Preface

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

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

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

Audience

This guide is for customers who will be installing TimesTen Classic. See *Oracle TimesTen In-Memory Database Scaleout User's Guide* for information about the installation requirements for TimesTen Scaleout

Related documents

TimesTen documentation is available on the TimesTen documentation website.

Oracle Database documentation is also available on the Oracle documentation website. This may be especially useful for Oracle Database features that TimesTen supports but does not attempt to fully document.

Conventions

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

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

This document uses the following text conventions:

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

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

Convention	Meaning
installation_dir	The path that represents the directory where the current release of TimesTen is installed.
timesten_home	The path that represents the home directory of a TimesTen instance.
release or *rr*	The first two parts in a release number, with or without dots. The first two parts of a release number represent a major TimesTen release. For example, 221 or 22.1 represents TimesTen Release 22.1.
DSN	The data source name.

Documentation Accessibility

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

Access to Oracle Support

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

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

What's New

This section summarizes the new features of Oracle TimesTen In-Memory Database Release 22.1 that are documented in this guide. It provides links to more information.

This book is for TimesTen Classic only. For information on TimesTen Scaleout, see "Prerequisites and Installation of TimesTen Scaleout" in the *Oracle TimesTen In-Memory Database Scaleout User's Guide*.

New features in Release 22.1.1.1.0

- On Linux platforms that use systemd, you can use systemd to automatically manage the TimesTen daemon. See "About creating an instance on Linux/UNIX" and see "Using systemd to Manage a TimesTen Service" for details.
- You can perform a fast patch upgrade. This involves performing operations to modify a TimesTen instance to point to a new installation while ensuring the shared memory segment remains in memory. See "About performing a fast patch upgrade" for details.

1

Overview of the Installation Process in TimesTen Classic

This chapter provides an overview of the topics you should be familiar with prior to installing TimesTen Classic. For information on TimesTen Scaleout, see "Prerequisites and Installation of TimesTen Scaleout" in the *Oracle TimesTen In-Memory Database Scaleout User's Guide*.

Topics include:

- Overview of installations and instances
- Understanding the TimesTen users group
- Installation and instance management on Linux, UNIX or macOS
- Installation and instance management on Windows
- Operating system prerequisites
- Planning the installation and its deployment
- Environment variables

Overview of installations and instances

This section discusses these topics:

- Distribution media and the distribution
- Instance administrator
- TimesTen installations
- TimesTen instances
- Instance home
- Instance configuration file (timesten.conf)

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

Distribution media and the distribution

The TimesTen product is packaged into distribution media that you download. For each supported platform, TimesTen is packaged into one distribution. A distribution consists of a single ZIP file.

The distribution differs depending on the platform:

- On a Linux/UNIX 64-bit host: The distribution file name indicates the release number, the type of distribution, and the platform. For example, for release `22.1.1.10.0` on a Linux 64-bit host, the distribution file name is `timesten2211100.server.linux8664.zip`. Use this file for installing either the full product or for installing just the client.
- On a Linux 32-bit host:
  - The distribution file name indicates the release number, the type of distribution, and the platform. For example, for release `22.1.1.10.0`, the distribution file name is `timesten.2211100.client.linux86.zip`.
  - There is one distribution that contains the TimesTen client. Only the TimesTen client is supported on a Linux 32-bit host.
- On a macOS host:
  - The distribution file name indicates the release number, the type of distribution, and the platform. For example, for release `22.1.1.10.0`, the distribution file name is `timesten.2211100.client.macos64.zip`.
  - There is one distribution that contains the TimesTen client. Only the TimesTen client is supported on a macOS host.
- On a Windows host:
  - The distribution file name indicates the release number and platform. For example, `timesten2211100.win64.zip`.
  - There is one distribution that contains the TimesTen client. Only the TimesTen client is supported on Windows.

Instance administrator

On a Linux, UNIX or macOS host, the *instance administrator* is the operating system user who extracts the distribution. When the instance administrator extracts the distribution, a TimesTen installation is created. See "TimesTen installations" for information on TimesTen installations. The instance administrator also plays a role in instances. See "TimesTen instances" for information.

On a Windows host, the instance administrator is the operating system user who extracts the distribution and runs the installer.

Note that the instance administrator:

- Cannot be the root user
- Has the operating system permissions to read all files and to execute all executable files in the installation
- Must be a member of the TimesTen users group. (See "Understanding the TimesTen users group " for information.)

TimesTen installations

An *installation* is the set of files installed on the host from the distribution. The *installation directory* is the directory under which the installation is created.

The instance administrator is the only user who can delete the installation.

> **Note:**
>
> - Installations are read-only. Do not add, alter, or remove files or directories within the installation.
> - TimesTen does not maintain any inventory of installations.
> - File path names containing multibyte characters are not supported.

These sections provide additional information:

- Installations on Linux or UNIX
- Installations on macOS
- Installations on Windows

Installations on Linux or UNIX

For installations on a Linux/UNIX 64-bit host:

- On a Linux host, a full installation or a client installation is supported for TimesTen Scaleout and for TimesTen Classic.
- On a UNIX host, a full installation or a client installation is supported for TimesTen Classic only.
- Multiple instances may share a single installation.

See "Creating an installation on Linux/UNIX" for information.

For installations on a Linux 32-bit host: A client-only installation is supported. The TimesTen client can connect to a database in either TimesTen Scaleout or in TimesTen Classic. See "Creating a TimesTen client installation" for information.

Installations on macOS

A client-only installation is supported on a macOS host. The TimesTen client can connect to a database in either TimesTen Scaleout or in TimesTen Classic.

See "Creating a TimesTen client installation" for information.

Installations on Windows

On Windows, after you extract the ZIP file, the instance administrator must run the `setup.exe` installer from the `WIN64` subdirectory. This process creates a single installation and a single instance. No additional instances can be created.

The TimesTen client can connect to databases in either TimesTen Scaleout or in TimesTen Classic running on a separate Linux or UNIX server.

See "Creating an installation on Windows" for information.

TimesTen instances

An instance refers to either:

- A running TimesTen daemon (`timestend`) and its children and associated processes, along with the configuration files and other supporting files required for its operation (full instance)
- A set of configuration files and other supporting files required for the use of TimesTen clients (client-only instance)

Each instance has an *instance home*. This is the top level of directory structure associated with the instance and represented in this document as *`timesten_home`*. See "Instance home" for information. A full instance can manage one or more databases. A client instance cannot have a database itself. Multiple instances can run from a single installation. TimesTen does not maintain an inventory of instances on a host and does not maintain an inventory of all instances associated with a particular installation.

On a Windows host, there is one instance in an installation that is created automatically during installation. The instance name is `instance`.

The role of the instance administrator for instances is as follows:

- The instance administrator for a full instance creates and manages databases, loads databases into memory and from memory, modifies and destroys the instance, performs all management activities, and performs backup and restore operations.
- The instance administrator for a client instance creates, modifies and destroys the instance.
- On a Linux, UNIX, or macOS host:
  - The instance administrator is also the only user who can create the instance (by running the `ttInstanceCreate` utility) and is the instance administrator for all instances created from this installation.
  - You cannot change the instance administrator after that administrator creates the installation or the instance.
  - The `ttInstanceCreate` utility enforces that the instance administrator cannot create the instance within the TimesTen installation tree. See "ttInstanceCreate" in the *Oracle TimesTen In-Memory Database Reference* for information on the `ttInstanceCreate` utility.
- On a Windows host:
  - The instance administrator is the operating system user who extracts the distribution and runs the installer. There is no `ttInstanceCreate` utility on Windows. This instance administrator is the instance administrator for the instance.
  - The one installation and the single instance must have the same owner (the instance administrator).

- An instance has a single instance administrator, who is the user who created the instance.
- You cannot change the instance administrator after that administrator runs the installer.

Instance home

On a Linux, UNIX, or macOS host: The *instance home* is a directory that is created when the instance administrator runs the `ttInstanceCreate` utility.

On a Windows host: The *instance home* is a directory that is created as a result of the instance administrator running the installer.

This directory is owned by the instance administrator.

The instance home contains all the files that are configured specifically for the instance. It is indicated by *`timesten_home`* in the TimesTen documentation.

There are two types of instance home directories.

Either:

- Full instance home: Supports the full use of TimesTen, including the server and direct mode. It must be a local directory to the host on which the instance runs.
- Client-only instance home: Provides the files required to run TimesTen clients and is created when TimesTen is configured for client-only use. It must be a local directory to the host on which the instance runs.

On a Linux, UNIX, or macOS host: Users of a particular TimesTen instance must set their environment by sourcing `ttenv.sh` or `ttenv.csh` (where which you use depends on your shell) provided in each instance. See "Environment variables" for more information.

On a Windows host: You can either register the environment variables during the installation process for a persistent setting or you can execute the `ttenv.bat` file. See "Create an installation on Windows" and "Environment variables" for details.

> **Note:**
>
> - A single instance home cannot be shared by more than one instance.
> - The instance home includes symbolic links to the associated installation.

Instance configuration file (timesten.conf)

The instance configuration file defines the attributes of the TimesTen instance. It resides in the *`timesten_home`*`/conf` directory and is named `timesten.conf`. The file is an ASCII text file, consisting of *`name`*`=`*`value`* pairs, one pair per line.

Here is a sample configuration file for a full instance. Comments are indicated by "`#`".

```
# TimesTen Instance Configuration File
# Created by ttInstanceCreate
hostname=host1
timesten_release=22.1
```

```
instance_name=instance1
daemon_port=6624
server_port=6625
tns_admin=
admin_user=myadmin
admin_uid=12345
group_name=ttgroup
instance_guid=39734D8C-E59A-4164-A77D-FC4327FF9496
verbose=1
```

Some of these values are known or provided by TimesTen, others are according to your choices or specifications during installation or instance creation or modification.

For complete information about this file, see "TimesTen Instance Configuration File" in the *Oracle TimesTen In-Memory Database Reference*.

Understanding the TimesTen users group

On a Linux, UNIX, or macOS host:

- TimesTen restricts access to the installation and the instances created from that installation to members of a single operating system group. This group, called the *TimesTen users group*, owns the installation and the instances created from the installation. The TimesTen users group must be the primary group for the instance administrator. Create this group (for example, `timesten`) and add the desired operating system users prior to installation. Once you create the TimesTen users group, you cannot change the name of the group or the group ID.
- Users who wish to access databases through TimesTen utilities or direct mode applications must be members of the TimesTen users group. This group can be the user's primary or secondary group.
- Users who connect to a database through a client connection do not have to be members of the TimesTen users group.

On a Windows host:

- TimesTen is installed by the instance administrator. This instance administrator must be a member of the TimesTen users group.
- Information about the TimesTen installation is contained in the Windows operating system registry.

Installation and instance management on Linux, UNIX or macOS

These topics provide an overview of installation and instance management on a Linux, UNIX or macOS host:

Installation management:

- Installation creation on Linux, UNIX, or macOS
- Installation deletion on Linux, UNIX, or macOS
- Installation copying on Linux or UNIX

Instance management

- Instance creation on Linux, UNIX, or macOS
- Instance modification on Linux, UNIX, or macOS
- Upgrading or downgrading the instance on Linux, UNIX, or macOS
- Instance removal on Linux, UNIX, or macOS

Installation creation on Linux, UNIX, or macOS

The instance administrator creates the installation by extracting the distribution. See "Distribution media and the distribution" for information on the distribution. For a Linux/UNIX 64-bit host, see "Creating an installation on Linux/UNIX" for information. For a macOS or a Linux 32-bit host, see "Creating a TimesTen client installation" for information.

The instance administrator can run the `ttInstallationCheck` utility after installation to verify the installation has the expected contents and permissions. For a Linux/UNIX 64-bit host, see "Verify an installation on Linux/UNIX" for more information. For a macOS or a Linux 32-bit host, see "Verify a client installation" for information.

Installation deletion on Linux, UNIX, or macOS

The instance administrator who created the installation is the only user who can delete the installation. Deleting the installation involves manually deleting the installation tree (the files and the directories within the installation).

For a Linux/UNIX 64-bit host, see "Deleting an installation on Linux/UNIX" for information.

For a macOS or a Linux 32-bit host, see "Deleting a TimesTen client installation" for information.

Installation copying on Linux or UNIX

Since installations are read only and immutable, you can pack the installation (using a tool like ZIP), copy it to another host, and unpack it. As long as the file permissions are maintained and the files are copied, the copied installation is valid. You can use the `ttInstallationCheck` utility to verify the installation. See "Copying an installation on Linux/UNIX" for information.

Instance creation on Linux, UNIX, or macOS

The instance administrator who created the installation (by extracting the distribution) is the only user who can create the instance. The instance administrator creates the instance by running the `ttInstanceCreate` utility located in the `/bin` area of the installation directory tree (*`installation_dir`*`/tt22.1.1.10.0/bin`).

The instance administrator creates a client-only instance by running `ttInstanceCreate` with the `-clientonly` option. (On a macOS or a Linux 32-bit host, the `-clientonly` option is not required.) See "TimesTen instances" for information on TimesTen instances.

The `ttInstanceCreate` utility creates the instance, creates the instance home directory, sets the permissions on the instance home directory, and populates the directory with the appropriate files. See "Instance home" for information on the instance home directory.

For a Linux/UNIX 64-bit host, see "Creating an instance on Linux/UNIX: Basics" for more information on the `ttInstanceCreate` utility and the procedure for creating an instance.

For a macOS or a Linux 32-bit host, see "Creating a TimesTen client instance" for information.

Instance modification on Linux, UNIX, or macOS

The instance administrator who created the installation and the instance is the only user who can modify the instance. The instance administrator modifies the instance by running the `ttInstanceModify` utility located in the `/bin` area of the *`timesten_home`* directory. See "Instance home" for information on this directory.

The instance administrator can run the `ttInstanceModify` utility either interactively or by specifying a supported option. For a Linux/UNIX 64-bit host, see "Modifying an instance on Linux/UNIX" for information on the `ttInstanceModify` utility and the procedure for modifying an instance. For a macOS or a Linux 32-bit host, see "Modifying a TimesTen client instance" for information.

The instance administrator can also change the attributes of the instance by modifying the instance configuration file. See "Instance configuration file (timesten.conf)" for information on this file. Also see "TimesTen Instance Configuration File" in the *Oracle TimesTen In-Memory Database Reference*.

Upgrading or downgrading the instance on Linux, UNIX, or macOS

An instance can be upgraded from one patch release of TimesTen to a later patch release. Instances can also be downgraded from one patch release to an earlier one. Upgrades and downgrades are only possible within a single major release (for example, from `22.1.w.x.0` to `22.1.y.z.0`.

The instance administrator who created the installation and the instance is the only user who can upgrade or downgrade the instance. The instance administrator upgrades or downgrades the instance by running the `ttInstanceModify` utility located in the `/bin` area of the *`timesten_home`* directory. See "Instance home" for information on this directory.

The procedure for upgrading or downgrading the instance involves associating the instance with a different installation. The instance administrator runs the `ttInstanceModify` utility with the `-install` option to accomplish this.

For a Linux/UNIX 64-bit host, see "Modifying an instance on Linux/UNIX" for information on the `ttInstanceModify` utility and see "Associate an instance with a different installation (upgrade or downgrade)" for the procedure to associate an instance with a different installation.

For a macOS or a Linux 32-bit host, see "Modifying a TimesTen client instance" for information.

Instance removal on Linux, UNIX, or macOS

The instance administrator who created the installation and the instance is the only user who can remove (destroy) the instance. The instance administrator destroys the instance by running the `ttInstanceDestroy` utility located in the `/bin` area of the installation directory tree (*`installation_dir`*`/tt22.1.1.10.0/bin`).

The instance to be destroyed is determined by the setting of the `TIMESTEN_HOME` environment variable. See "Environment variables" for information on this environment variable and how to set it.

For a Linux/UNIX 64-bit host, see "Destroying an instance on Linux/UNIX" for information on the `ttInstanceDestroy` utility and the procedure for destroying an instance.

For a macOS or Linux 32-bit host, see "Destroying a TimesTen client instance" for information.

Installation and instance management on Windows

These topics provide an overview of installation and instance management on Windows:

Installation and instance management:

- Installation and instance creation on Windows
- One installation on Windows
- Installation and instance deletion on Windows

Installation and instance creation on Windows

The instance administrator extracts the distribution and then runs the TimesTen installer to create the installation and the instance. See "Distribution media and the distribution" for information on the distribution.

The TimesTen installer creates a single TimesTen client-only installation (and instance). No additional installations (or instances) can be created without first uninstalling the existing one. Thus, there can be only one single `22.1` installation at a time. See "Overview of the installation process on Windows" for details of the installation process.

One installation on Windows

A Windows host does not support multiple installations from the same TimesTen major release, such as `22.1`. For example, the host cannot have both an `22.1.1.`*`x`*`.0` installation and an `22.1.2.`*`x`*`.0` installation.

If there is a `22.1` release of Windows installed and you wish to install a different patch release of `22.1`:

- The instance administrator runs the installer to install the new release.
- The installer asks if the previous installation can be overwritten with the new one.

  If the instance administrator answers `yes`, the one provided instance can make use of the new installation.

Installation and instance deletion on Windows

The instance administrator deletes the installation by using the Control Panel or System Settings (depending on your version of Windows). Deleting the installation also deletes the instance. See "Deleting an installation on Windows" for information on the procedure for deleting an installation. Also see "Verify the uninstallation is successful on Windows" for the procedure to verify the success of the uninstallation.

Operating system prerequisites

Ensure you review (and perform) these operating system prerequisites before you install TimesTen Classic.

- Linux prerequisites
- AIX prerequisites
- Solaris prerequisites

Linux prerequisites

Perform these prerequisites on Linux:

- Complete general Linux prerequisites
- Create the TimesTen Users Group

Complete general Linux prerequisites

On Oracle Linux 7 and 8 systems, TimesTen requires the `libaio` library. To install this library, run:

```
sudo yum install libaio
```

On Oracle Linux 8.x and Red Hat Enterprise Linux 8.x systems, TimesTen depends on two libraries:

- The `ncurses-compat-libs` package: Enables cursor-based command recall in `ttIsql`
- The `/usr/lib64/libnsl.so.1` library: Enables OCI, TimesTen Cache and TimesTen passthrough

On SUSE Linux Enterprise Server, you need to install `libncurses5`. To do this, run:

```
zypper -n install libncurses
```

It is recommended that you enable stack traces for TimesTen. On Linux systems, use `pstack` or `gdb` to get a stack trace.

Create the TimesTen Users Group

This section summarizes the steps for creating the TimesTen users group:

- Create a TimesTen users group and add desired users.
- Determine the operating system user that will be the instance administrator. That user must be a member of the TimesTen users group. This user creates the installation.

> **Note:**
>
> Do not create a TimesTen installation as an operating system user whose name matches any of the TimesTen predefined internal users: `GRID`, `PUBLIC`, `SYS`, `SYSTEM`, or `TTREP`.

As an example, `instanceadmin` is the name of the operating system user and `timesten` is the name of the TimesTen users group.

1. Create the TimesTen users group. Name the group `timesten` with group ID `10000`. This information is needed when configuring HugePages. See "Configure HugePages" for more information.

   ```
   sudo groupadd -g 10000 timesten
   ```

2. Create the `instanceadmin` user with UID `55000` and assign this user to the `timesten` primary group. Then, create a password for the `instanceadmin` user.

   ```
   sudo useradd -u 55000 -g timesten instanceadmin
   sudo passwd instanceadmin
   ```

AIX prerequisites

On UNIX, semaphores are configured dynamically by the kernel.

On UNIX hosts with the required patch levels, TimesTen Classic 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. Use the `vmo` command to configure the size of the large page physical memory pool. This example allocates 4 GB to the large page physical memory pool:

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

> **Note:**
>
> There is some benefit in using `vmo` to set `vmm_mpsize_support` to a value of 3 (if available) or 2 to optimize memory page usage.

It is recommended that you enable stack traces for TimesTen. On AIX systems, use `procstack` to get a stack trace.

Solaris prerequisites

It is recommended that you enable stack traces for TimesTen. On Solaris systems, use `pstack` to get a stack trace.

In addition, before installation, use the information in these sections to improve the performance of TimesTen Classic on your Solaris system:

File system options (Solaris)

On a Solaris UFS file host, mount the file system with the `-forcedirectio` option if you plan to have applications that use `DurableCommits=1`.

Create a project (Solaris)

Create a project to manage system resources, such as shared memory, file descriptors and semaphores.

You can create a group project or a user project.

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

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

1. Log in as user `root`.
2. Add the group project.

   ```
   # projadd -G timesten timestenproj
   ```

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

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

4. Modify the maximum number of semaphores to 4096.

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

   > **Note:**
   >
   > For each active database, TimesTen Classic consumes a minimum of 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)" timestenproj
   ```

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

   ```
   # newtask -p timestenproj -c $$
   ```

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

1. Log in as user `root`.
2. Add the user project.

   ```
   # projadd -U timestenproj user.timestenuser
   ```

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

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

4. Modify the maximum number of semaphores to 4096.

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

   > **Note:**
   >
   > For each active database, TimesTen Classic 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.timestenuser
   ```

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 project settings at the top of the `project` file above the default settings.

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

- Edit the `/etc/project` file to move the `timestenproj` project entry so that it precedes the default entry.
- Execute the following before restarting daemons. This is required if the project was created with `-G` only.

  ```
  # newtask -p timestenproj -c $$
  ```

> **Note:**
>
> On a Solaris host, use `MemoryLock` with a setting of 3 or 4. A `MemoryLock` setting of 1 or 2 requires TimesTen to have been installed as `root`, which is not advisable.

Planning the installation and its deployment

This section is applicable for full installations and full instances. Client-only installations and instances are irrelevant. For planning purposes, consider the information in these sections:

- Locations of database files and user files
- Locations of the databases and the applications

Locations of database files and user files

These are the TimesTen requirements and recommendations regarding locations of databases and other user files:

- Storing database files (checkpoint and log files) or any other user files anywhere under the TimesTen installation path is not supported. The installation is immutable—do not add, change, or remove anything.
- It is strongly advised to not store database files or other user files under the instance home. Anything in or under the instance home will be removed if the instance is destroyed.
- For performance reasons, it is advisable to store TimesTen checkpoint files (the `DataStore` location in the database definition) on a different device from TimesTen transaction log files (the `LogDir` location).

Once you have a TimesTen installation, you can estimate the size of your database and the disk space required. Refer to "Storage provisioning for TimesTen" in *Oracle TimesTen In-Memory Database Operations Guide*.

Locations of the databases and the applications

Consider:

- Unless there are concerns about resource contention between your application and TimesTen Classic, it is best to have your application on the same host as the databases in TimesTen Classic. This allows the application to use direct connections, which offer much better response time and throughput than client/server connections, due primarily to avoidance of network round trips.
- To use TimesTen Cache, it is best to have the TimesTen Classic and Oracle databases on different hosts, to avoid resource contention between them.

> **Note:**
>
> These are general guidelines only, not necessarily suitable for every particular situation.

Environment variables

These sections discuss environment variables and are specific to TimesTen Classic. For specifics on environment variables in TimesTen Scaleout, see "Environment variables" in the *Oracle TimesTen In-Memory Database Scaleout User's Guide*.

Setting environment variables for TimesTen

You set environment variables for a terminal window, which enables the window to run commands for a particular instance. Here is a list of situations where you should set your environment variables:

- After you create the instance
- Before using any TimesTen utility
- Before executing a direct mode application on a host running an instance
- Before executing a client server application on a host running a client

On a Linux, UNIX, or macOS host, you set the environment variables by sourcing `ttenv.sh` or `ttenv.csh` (where which you use depends on your shell). On a Windows host, you set the environment variables by running the `ttenv.bat` batch file. TimesTen creates the scripts after you create an instance.

In TimesTen Classic, these scripts are located in the `/bin` directory of the instance home.

After sourcing these scripts, the environment variables required to use an instance are set.

The environment variables include `TIMESTEN_HOME`, `PATH`, `LD_LIBRARY_PATH` (or equivalent) and `TNS_ADMIN`.

For example:

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

```
% cd timesten_home/bin
% . ttenv.sh
```

For a `csh` or `tcsh` shell:

```
% cd timesten_home/bin
% source ttenv.csh
```

Once the `TIMESTEN_HOME` variable is set so that the instance home is known, TimesTen makes additional settings, such as the daemon port, according to the *timesten_home*`/conf/timesten.conf` instance configuration file. See "Instance configuration file (timesten.conf)" for information on the instance configuration file.

> **Note:**
>
> The output after sourcing indicates the path and the `TIMESTEN_HOME` settings, but may not indicate all settings.

Alternatively, you can use `ttenv` in command-line mode to fork a new shell, to set the environment, and to execute the specified command. For example, to execute `ttIsql` to connect to `database1`:

```
% cd timesten_home/bin
% ./ttenv ttIsql database1
```

Environment variables are set inside your `ttIsql` session and the `ttIsql` prompt displays. When you exit `ttIsql`, your shell will have its original environment variable settings.

On Windows, execute the `ttenv.bat` batch file from a DOS window, which changes the environment for your DOS session. For example:

```
C:\TimesTen\tt221_64\instance\bin>ttenv
```

> **Note:**
>
> - On Windows, the instance home, path, classpath, library path, and path are set persistently during installation if "Register environment variables" is enabled, which is the case by default. These settings are reflected in the System control panel and persist between sessions. It is not necessary to run `ttenv.bat`.
> - The `ttenv` command-line mode does not apply to Windows.

Environment variable descriptions

These sections provide more details on the environment variables:

- TIMESTEN_HOME environment variable
- NLS_LANG environment variable
- Shared library path environment variable
- PATH environment variable
- Temporary directory environment variable
- TNS_ADMIN environment variable
- Java environment variables
- SYSODBCINI environment variable
- ODBCINI environment variable
- SYSTTCONNECTINI environment variable

TIMESTEN_HOME environment variable

The TIMESTEN_HOME environment variable specifies the home directory of the instance. You explicitly set this variable when sourcing either ttenv.sh or ttenv.csh (where which you use depends on your shell).

On Windows, the TIMESTEN_HOME environment variable is set persistently if you register the environment variables during installation (the default), or is set for your session if you execute ttenv.bat.

NLS_LANG environment variable

This environment variable is relevant for OCI, Pro*C/C++, and ODP.NET. It is ignored for ODBC and JDBC. The character set specified in the sys.odbc.ini or user odbc.ini is used by default for the connection, if not overridden by NLS_LANG. While setting the character set explicitly is recommended, the default is normally AMERICAN_AMERICA.US7ASCII. To use the environment variable to set the character set, do the following:

```
% setenv NLS_LANG AMERICAN_AMERICA.WE8ISO8859P1
```

On Windows, the NLS_LANG setting is searched for in the registry, HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\NLS_LANG, if it is not in the environment. If your program has trouble connecting to the database, confirm the NLS_LANG setting is valid and that it indicates a character set supported by TimesTen.

For more information, see:

- "Specifying a character set" in the *Oracle TimesTen In-Memory Database C Developer's Guide*.
- "Supported character sets" in the *Oracle TimesTen In-Memory Database Reference*.

Shared library path environment variable

The shared library path environment variable is set when sourcing ttenv.sh or ttenv.csh (where which you use depends on your shell). This environment variable specifies the path for shared libraries.

The shared library path environment variable is set as follows:

- On Linux, ttenv.sh or ttenv.csh (where which you use depends on your shell) adds *$TIMESTEN_HOME*/install/lib to LD_LIBRARY_PATH.
- On UNIX, ttenv.sh or ttenv.csh (where which you use depends on your shell) adds *$TIMESTEN_HOME*/install/lib to LIBPATH.
- On macOS, ttenv.sh or ttenv.csh (where which you use depends on your shell) adds *$TIMESTEN_HOME*/install/lib:*$TIMESTEN_HOME*/install/ttoracle_home/ to DYLD_LIBRARY_PATH.
- On Windows, ttenv.sh (or ttenv.csh) adds tt221_64\lib to LIB (or the lib directory under the top level of the installation, if some other directory name was chosen).
- On Solaris systems, *timesten_home*/install/lib in LD_LIBRARY_PATH or LD_LIBRARY_PATH_64, as appropriate.

PATH environment variable

TimesTen provides utilities for managing and debugging your applications. For these utilities to be available, the path for executables in *$TIMESTEN_HOME*/bin and *$TIMESTEN_HOME*/install/bin must be designated in the PATH setting. The path is updated to include these directories when you source ttenv.sh or ttenv.csh (where which you use depends on your shell).

In addition, to compile programs, be sure the location of the compiler for your programming language is in the PATH setting.

Temporary directory environment variable

The temporary directory environment variable specifies the location of the temporary directory. TimesTen uses this directory during recovery and other operations. The ttenv.sh or the ttenv.csh script does not set this environment variable. You must explicitly set it to avoid the operating system default.

- On a Linux, UNIX, or macOS host, TMPDIR is the environment variable.
- On a Windows host, TMP is the environment variable.

TNS_ADMIN environment variable

The TNS_ADMIN environment variable specifies the full path to the directory where the tnsnames.ora file is located.

- For TimesTen OCI, Pro*C/C++, or ODP.NET, set the TNS_ADMIN environment variable to indicate the full path to the directory where the tnsnames.ora file is located.
- For TimesTen Cache in TimesTen Classic, set the TNS_ADMIN environment variable to indicate the full path to the directory where the tnsnames.ora file is located. This is for access to Oracle Database data.
- On a Linux or a UNIX host, also specify the -tnsadmin option for ttInstanceCreate or for ttInstanceModify to ensure that both TimesTen and user applications read the TNS_ADMIN setting.

Java environment variables

For Java applications, ensure that the locations of the java and javac executables are in the PATH setting.

In addition, Java classes and class libraries are found in the class path, as specified by the CLASSPATH environment variable. Before executing a Java program that loads any of the TimesTen JDBC drivers, the CLASSPATH setting must include the class library file and path:

$TIMESTEN_HOME/install/lib/ttjdbc*jdk_ver*.jar

where *jdk_ver* indicates the JDK version. (For example, ttjdbc17.jar.)

> **Note:**
>
> - This variable is set for your session by sourcing `ttenv.sh` or `ttenv.csh` (where which you use depends on your shell) or, on Windows, is set persistently during installation if environment variables are registered (default).
> - If multiple JAR files are listed in the `CLASSPATH`, ensure the TimesTen JAR file is listed first.

To check the JDK version:

```
% java -version
```

To use the JMS/XLA interface, these entries must also be in your `CLASSPATH`:

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

For example, the `CLASSPATH` would look like this example (replacing *timesten_home*`/install` as appropriate):

```
.:timesten_home/install/lib/ttjdbc17.jar:timesten_home/install
/lib/timestenjmsxla.jar:timesten_home/install/3rdparty/jms1.1/lib
/jms.jar:timesten_home/install/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, specify it as part of the environment variable in the `InitialContext` class and add the location to the `CLASSPATH` setting. See "JMS/XLA configuration file and topics" in *Oracle TimesTen In-Memory Database Java Developer's Guide* for more information.

SYSODBCINI environment variable

On a Linux or UNIX host, system DSNs and their connection attributes are defined in the `sys.odbc.ini` file. The default location, *$TIMESTEN_HOME*`/conf` is usually sufficient.

To override the name and location of this file at runtime, set the `SYSODBCINI` environment variable to the path name of a `sys.odbc.ini` file before starting the TimesTen daemon. Any user can use a system data source.

TimesTen first looks for a DSN in the user `odbc.ini` file. If the DSN is not found in this file, TimesTen looks in the `sys.odbc.ini` file.

Use of this environment variable is discouraged. For more information, see "Overview of user and system DSNs" in the *Oracle TimesTen In-Memory Database Operations Guide*.

ODBCINI environment variable

On a Linux or UNIX host, applications can use the `odbc.ini` file to define DSNs and their connection attribute settings. By default, TimesTen first looks for the user `odbc.ini` file in the home directory of the user running the application. To override the name and location of this file at run-time, set the `ODBCINI` environment variable to indicate a desired path and file name before launching the application.

If TimesTen cannot locate a user DSN file, on a Linux or UNIX host, it looks for the `sys.odbc.ini` file in *`$TIMESTEN_HOME`*`/conf`.

Use of this environment variable is discouraged. For more information, see "Overview of user and system DSNs" in the *Oracle TimesTen In-Memory Database Operations Guide*.

SYSTTCONNECTINI environment variable

On a Linux, UNIX, or macOS host, client applications can use the `sys.ttconnect.ini` file to define logical server names. For a description of logical server names, see "Working with the TimesTen Client and Server" in *Oracle TimesTen In-Memory Database Operations Guide*.

The default location, *`$TIMESTEN_HOME`*`/conf/sys.ttconnect.ini`, is usually sufficient. To override the name and location of this file at runtime, set the `SYSTTCONNECTINI` environment variable appropriately before starting the TimesTen daemon.

On a Windows host, configure logical server names using the **ODBC Data Source Administrator**.

Use of this environment variable is discouraged.

2

Installation of TimesTen Classic on Linux or UNIX

This chapter is for TimesTen Classic only. It discusses how to create and manage installations and instances for TimesTen Classic on Linux and UNIX hosts.

Before installing TimesTen Classic, ensure you have reviewed the terminology and completed the prerequisites. See Overview of the Installation Process in TimesTen Classic for more information.

For information on TimesTen Scaleout, see "Overview of TimesTen Scaleout" in the *Oracle TimesTen In-Memory Database Scaleout User's Guide*.

This chapter focuses primarily on full installations and full instances. See "Client-only Installations and Instances " for information on client-only installations and instances.

The examples in this chapter are performed on Linux, but are applicable to the platforms that support full installations and full instances.

Topics include:

- Creating an installation on Linux/UNIX
- About creating an instance on Linux/UNIX
- Creating an instance on Linux/UNIX: Basics
- Creating a TimesTen client instance
- Modifying an instance on Linux/UNIX
- Destroying an instance on Linux/UNIX
- Copying an installation on Linux/UNIX
- Deleting an installation on Linux/UNIX

There is an example at the end of the chapter that demonstrates the steps for creating an installation, creating an instance, modifying an instance, destroying an instance, and deleting the installation. This example ties all the concepts of the installation process together. See:

"Complete example for installing and uninstalling TimesTen Classic" for details.

There is also information on installing Oracle Clusterware. See "Installing Oracle Clusterware for use with TimesTen" for information.

Creating an installation on Linux/UNIX

The first step in creating the installation is to decide the operating system user who will install TimesTen Classic. This user installs TimesTen Classic by unzipping the distribution. See "Distribution media and the distribution" for information. This user is named the instance administrator. See "Instance administrator" for details.

> **Note:**
>
> The operating system user that unzips the distribution (and thereby installs TimesTen Classic) is named the instance administrator. This instance administrator is the only user that can create instances. See "TimesTen instances" and "Creating an instance on Linux/UNIX: Basics" for details.
>
> There is one distribution file:
>
> - On Linux 64-bit, there is not a separate client only installation file. For example, for release `22.1.1.10.0`, the distribution file name is `timesten2211100.server.linux8664.zip`. Unzip this file to install either the full product or the client. When you create the instance, there is a qualifier that you specify to create a client-only instance. See "Creating an instance on Linux/UNIX: Basics" for details.
> - On Linux 32-bit, there is only a client installation file.

The instance administrator can create an installation that is accessible by the instance administrator's primary group, by the instance administrator's secondary group, or by only the instance administrator. In addition, the instance administrator can and should verify the installation is successful. These sections cover these topics:

- Create an installation accessible by the instance administrator's primary group
- Create an installation accessible by the instance administrator's secondary group
- Create an installation accessible by only the instance administrator
- Verify an installation on Linux/UNIX

Create an installation accessible by the instance administrator's primary group

To create the installation that is accessible by the instance administrator's primary group:

1. Place the appropriate distribution file in some desired, accessible location.
2. Create the desired directory for the installation.
3. Change directories to the desired directory for the installation.
4. Extract the distribution into that directory.

In the following example on Linux, `ttuser1`, with primary group `timesten`, creates a full installation into the directory `fullinstall` (a subdirectory of the current directory) from a distribution file in `/swdir/TimesTen/ttinstallers`. This installation is accessible to members of `timesten`.

```
% mkdir fullinstall
% cd fullinstall
% unzip /swdir/TimesTen/ttinstallers/timesten2211100.server.linux8664.zip
[...UNZIP OUTPUT...]
```

The top level directory of the installed files is of the form `tt22.1.1.10.0`.

For example, this directory is created under `fullinstall`:

```
dr-xr-x--- 19 ttuser1 timesten 4096 Mar  2 22:07 tt22.1.1.10.0
```

The `tt22.1.1.10.0` directory includes such files as:

```
dr-xr-x---  3 ttuser1 timesten    4096 Jun 15 22:07 3rdparty
dr-xr-x---  2 ttuser1 timesten    4096 Jun 15 22:07 bin
dr-xr-x---  3 ttuser1 timesten    4096 Jun 15 22:07 include
dr-xr-x---  2 ttuser1 timesten    4096 Jun 15 22:07 info
dr-xr-x---  2 ttuser1 timesten    4096 Jun 15 23:33 kubernetes
dr-xr-x---  3 ttuser1 timesten    4096 Jun 15 22:07 lib
dr-xr-x---  7 ttuser1 timesten    4096 Jun 15 22:07 plsql
dr-xr-x---  3 ttuser1 timesten    4096 Jun 15 22:07 ttoracle_home
```

A `timesten` member can access the instance. A user who is not a member of `timesten` does not have access to this instance or any instance created from the installation.

Create an installation accessible by the instance administrator's secondary group

To create an installation that is accessible by a secondary group of the instance administrator:

1. Place the appropriate distribution file in some desired, accessible location.
2. Create the desired directory for the installation.
3. Change directories to the desired directory for the installation.
4. Extract the distribution into that directory.
5. Use the `chgrp` Linux command to change ownership of the installation to the desired group.

In the following example, `ttuser2`, with primary group `users` and secondary group `timesten`, creates a full installation into the directory `fullinstall_secgrp` from a distribution file in `/swdir/TimesTen/ttinstallers`. After `ttuser2` executes the `chgrp` command, this installation is accessible to members of `timesten`.

```
% mkdir fullinstall_secgrp
% cd fullinstall_secgrp
% unzip /swdir/TimesTen/ttinstallers/timesten2211100.server.linux8664.zip
[...UNZIP OUTPUT...]
```

The top level directory of the installed files is of the form `tt22.1.1.10.0`.

For example, under the installation directory, before the `chgrp` command:

```
dr-xr-x--- 19 ttuser2 users    4096 Mar  2 22:07 tt22.1.1.10.0
```

The `tt22.1.1.10.0` directory, shown before the `chgrp` command, includes such files as:

```
dr-xr-x---  3 ttuser2 users    4096 Jun 15 22:07 3rdparty
dr-xr-x---  2 ttuser2 users    4096 Jun 15 22:07 bin
dr-xr-x---  3 ttuser2 users    4096 Jun 15 22:07 include
dr-xr-x---  2 ttuser2 users    4096 Jun 15 22:07 info
dr-xr-x---  2 ttuser2 users    4096 Jun 15 23:33 kubernetes
dr-xr-x---  3 ttuser2 users    4096 Jun 15 22:07 lib
dr-xr-x---  7 ttuser2 users    4096 Jun 15 22:07 plsql
dr-xr-x---  3 ttuser2 users    4096 Jun 15 22:07 ttoracle_home
```

User ttuser2 executes chgrp from the fullinstall_secgrp installation directory:

```
% chgrp -R timesten .
```

Under the installation directory, after the chgrp command:

```
dr-xr-x--- 19 ttuser2 timesten 4096 Mar  2 22:07 tt22.1.1.10.0
```

The tt22.1.1.10.0 directory, shown after the chgrp command, includes such files as:

```
dr-xr-x---  3 ttuser2 timesten    4096 Jun 15 22:07 3rdparty
dr-xr-x---  2 ttuser2 timesten    4096 Jun 15 22:07 bin
dr-xr-x---  3 ttuser2 timesten    4096 Jun 15 22:07 include
dr-xr-x---  2 ttuser2 timesten    4096 Jun 15 22:07 info
dr-xr-x---  2 ttuser2 timesten    4096 Jun 15 23:33 kubernetes
dr-xr-x---  3 ttuser2 timesten    4096 Jun 15 22:07 lib
dr-xr-x---  7 ttuser2 timesten    4096 Jun 15 22:07 plsql
dr-xr-x---  3 ttuser2 timesten    4096 Jun 15 22:07 ttoracle_home
```

After the chgrp command, a timesten member can access the instance. A user who is not a member of timesten does not have access to this installation or any instance created from the installation.

Create an installation accessible by only the instance administrator

An installation can be created that is accessible by only the instance administrator, although this is not typical. To accomplish this:

1. Place the appropriate distribution file in the desired directory.
2. Create the desired directory for the installation.
3. Change directories to the desired directory for the installation.
4. Extract the distribution into that directory.
5. Change the permissions on the installation directory tree.

In the following example, ttuser1 creates a full installation into the directory fullinstall_installeronly from a distribution file in /swdir/TimesTen/ttinstallers. This installation is accessible to ttuser1 only.

```
% mkdir fullinstall_installeronly
% cd fullinstall_installeronly
% unzip /swdir/TimesTen/ttinstallers/timesten2211100.server.linux8664.zip
[...UNZIP OUTPUT...]
% chmod -R go-rwx tt22.1.1.10.0
```

The top level directory of the installed files is of the form tt22.1.1.10.0.

For example, under the installation directory:

```
dr-x------ 19 ttuser1 timesten 4096 Jun  2 22:07 tt22.1.1.10.0
```

Contents of tt22.1.1.10.0 includes such files as:

```
dr-x------  5 ttuser1 timesten    4096 Jun  2 22:07 3rdparty
dr-x------  2 ttuser1 timesten    4096 Jun  2 22:07 bin
dr-x------  3 ttuser1 timesten    4096 Jun  2 22:07 include
dr-x------  2 ttuser1 timesten    4096 Jun  2 22:07 info
dr-x------  2 ttuser1 timesten    4096 Jun  2 23:33 kubernetes
dr-x------  2 ttuser1 timesten    4096 Mar  2 22:07 lib
```

```
dr-x------  8 ttuser1 timesten    4096 Mar  2 22:07 plsql
dr-x------  3 ttuser1 timesten    4096 Mar  2 22:07 ttoracle_home
```

Only `ttuser1` has access to this instance.

Verify an installation on Linux/UNIX

To verify the installation, run the `ttInstallationCheck` utility. In addition, you can review the installation directory and subdirectories.

- The ttInstallationCheck utility
- The installation directory and subdirectories

The ttInstallationCheck utility

The `ttInstallationCheck` utility, located in the *`installation_dir`*`/tt22.1.1.10.0/bin` directory, verifies the success or failure of the installation. This utility generates an error if the checksum value for the installation differs from the original checksum value. Checksum values are different if there are any of these changes to the installation directory or files:

- Contents of a file
- Name of a file
- Addition of a file to a directory
- Removal of a file from a directory
- Changes to the permissions of a file or directory

In this example, the installation is verified:

```
% installation_dir/tt22.1.1.10.0/bin/ttInstallationCheck
This installation has been verified.
```

In this example, permissions on a file were changed, and `ttInstallationCheck` generates an error:

```
% installation_dir/tt22.1.1.10.0/bin/ttInstallationCheck
Cannot validate the installation in /installation_dir/tt22.1.1.10.0.
```

See "ttInstallationCheck" in the *Oracle TimesTen In-Memory Database Reference* for detailed information on the `ttInstallationCheck` utility.

The installation directory and subdirectories

A TimesTen full installation includes these subdirectories located under the top-level *`installation_dir`*`/tt22.1.1.10.0` directory (Not all of the subdirectories are included in this list).

- `3rdparty`: Includes resources for:
  - Apache ZooKeeper for TimesTen Scaleout (Unused by TimesTen Classic)
  - Java Message Service (JMS)
- `bin`: TimesTen utilities and executables
- `grid`: Files and resources for TimesTen Scaleout (Unused by TimesTen Classic)

- `include`: TimesTen include files, among them `timesten.h` (for TimesTen ODBC features) and `tt_errCode.h` (for information about TimesTen error codes)
- `kubernetes`: The directory containing the `operator.zip` file needed for the TimesTen Kubernetes Operator. See the *Oracle TimesTen In-Memory Database Kubernetes Operator User's Guide* for information on the TimesTen Kubernetes Operator.
- `lib`: TimesTen libraries
- `plsql`: Files and resources for TimesTen PL/SQL
- `ttoracle_home`: Oracle Database Instant Client files and resources, for OCI, Pro*C/C++, and ODP.NET

About creating an instance on Linux/UNIX

You can create a TimesTen full instance or a TimesTen client instance:

- A TimesTen full instance refers to a running TimesTen daemon (`timestend`) and its children and associated processes, along with the configuration files and other supporting files required for its operation. A full instance can manage one or more databases.
- A TimesTen client instance refers to a set of configuration files and other supporting files required for the use of TimesTen clients. A client instance cannot have a database itself.

The information in the following section is relevant for TimesTen full instances. See "Creating a TimesTen client instance" and "Client-only Installations and Instances " for information on creating TimesTen client instances.

In TimesTen Classic, when you create a TimesTen full instance on Linux/UNIX, you can choose how you want the TimesTen daemon to be managed at system startup.

A Linux system can support both the System V init mechanism and systemd for system startup. This presents you with choices depending on if and how you want the TimesTen daemon to be managed at system startup:

- If you want the TimesTen daemon automatically managed, choose systemd. The instance administrator does not control the TimesTen daemon.
- If you want the TimesTen daemon to be started at system startup and stopped at system shutdown, then you can choose either the System V init mechanism or systemd. You cannot choose both. Note that if you choose systemd, the TimesTen daemon will also be automatically managed by systemd and the instance administrator will have no control over the TimesTen daemon. If you choose the System V init mechanism, the instance administrator controls the TimesTen daemon at all times, except at system startup and system shutdown.
- If you want the instance administrator to control the TimesTen daemon, do not use either the System V init mechanism or systemd to control the TimesTen daemon. See "Creating an instance on Linux/UNIX: Basics" for details.

The way in which you create a TimesTen full instance depends on what choice you make.

On a UNIX system or on a Linux system that does not support systemd, see "Creating an instance on Linux/UNIX: Basics" for information.

On a Linux system that supports both the System V init mechanism and systemd, you can choose one of these options:

- System V init mechanism: See "Creating an instance on Linux/UNIX: Basics" for information.
- systemd: See "Using systemd to Manage a TimesTen Service" for information.
- Neither System V init mechanism nor systemd: See "Creating an instance on Linux/UNIX: Basics" for information.

Creating an instance on Linux/UNIX: Basics

The information in the following section is relevant for TimesTen full instances. See "Creating a TimesTen client instance" and "Client-only Installations and Instances " for information on creating TimesTen client instances.

This section discusses creating a full instance on:

- A Linux/UNIX system that does not support systemd
- A Linux system that supports both the System V init mechanism and systemd, but you have chosen either the SystemV init mechanism or have chosen to have the instance administrator control the TimesTen daemon at all times.

See "Using systemd to Manage a TimesTen Service" for information on creating a full instance when your choice is systemd.

To create a full instance:

- Use the `ttInstanceCreate` utility located in the *`installation_dir`*`/tt22.1.1.10.0/bin` directory to create the instance. You can run the `ttInstanceCreate` utility on the command line, in a file, or interactively.
- Configure Linux kernel parameters.
- Optionally, run the `timesten_home/bin/setuproot` script with the `-install` option to have the TimeTen daemon automatically started at system startup and automatically stopped at system shutdown. This script copies the System V init scripts to the proper location (`/etc/init.d`). If you want the instance administrator to control the TimesTen daemon at system startup and at system shutdown, do not run this script.

> **Note:**
>
> - Only the user who unpacked the distribution (the creator of the installation) can create an instance from the installation.
> - This user must be a member of the TimesTen users group.
> - This user creates the instance by running the `ttInstanceCreate` utility.
> - After creating the instance, this user becomes the instance administrator of the instance.
> - This user is the only user that can be the instance administrator of this instance and all other instances associated with the installation.

Perform the procedures in these sections:

- Create a TimesTen full instance on Linux/UNIX
- Review the instance home directory and subdirectories
- Configure Linux kernel parameters
- Start an instance automatically at system startup with System V init scripts

Create a TimesTen full instance on Linux/UNIX

This section discusses the options you have to create a TimesTen full instance.

- Create a full instance interactively
- Create an instance by specifying options on the command line
- Create an instance interactively with the -record option
- Create an instance from information provided in a batch file
- Create an instance interactively for Oracle Clusterware

Create a full instance interactively

If you are using systemd as a service, see "Using systemd to Manage a TimesTen Service" for details.

This example uses the `ttInstanceCreate` utility to create a full instance. No options are specified on the command line.

Navigate to the *installation_dir*`/tt22.1.1.10.0/bin` area of the installation directory and then run the `ttInstanceCreate` utility located in that directory. The `ttInstanceCreate` utility must be run from the installation directory. User input is shown in bold.

```
% installation_dir/tt22.1.1.10.0/bin/ttInstanceCreate

NOTE: Each TimesTen instance is identified by a unique name.
      The instance name must be a non-null alphanumeric string, not
longer
      than 255 characters.

Please choose an instance name for this installation? [ tt221 ]
myinstance
Instance name will be 'myinstance'.
Is this correct? [ yes ]
Where would you like to install the myinstance instance of TimesTen?
[ /home/ttuser ] /scratch/ttuser
Creating instance in /scratch/ttuser/myinstance ...

NOTE: If you are configuring TimesTen for use with Oracle Clusterware,
the
      daemon port number must be the same across all TimesTen
installations
      managed within the same Oracle Clusterware cluster.

NOTE: All installations that replicate to each other must use the same
daemon
```

```
      port number that is set at installation time. The daemon port number
can
      be verified by running 'ttVersion'.

The default port number is 6624.

Do you want to use the default port number for the TimesTen daemon? [ yes ]
The daemon will run on the default port number (6624).

In order to use the cache features in any TimesTen databases
created within this instance, you must set a value for the TNS_ADMIN
environment variable. It can be left blank, and a value can be supplied later
using <install_dir>/bin/ttInstanceModify.

Please enter a value for TNS_ADMIN (s=skip)? [  ] s
What is the TCP/IP port number that you want the TimesTen Server to listen
on? [ 6625 ]

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

Would you like to use systemd to manage TimesTen? [ no ]

NOTE: The TimesTen daemon startup/shutdown scripts have not been installed.

The startup script is located here :
        '/scratch/ttuser/myinstance/startup/tt_myinstance'

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

The 22.1 Release Notes are located here :
  'installation_dir/tt22.1.1.1.0/README.html'

Starting the daemon ...
TimesTen Daemon (PID: 9020, port: 6624) startup OK.
Instance created successfully.
```

You have successfully created the TimesTen full instance. Proceed to "Review the instance home directory and subdirectories" for next steps.

Create an instance by specifying options on the command line

This example runs the `ttInstanceCreate` utility and specifies the name, the location, and the daemon port number on the command line. (The name of the directory specified in the `-location` option must exist prior to running the `ttInstanceCreate` utility.) See "ttInstanceCreate" in the *Oracle TimesTen In-Memory Database Reference* for the supported options.

```
% installation_dir/tt22.1.1.10.0/bin/ttInstanceCreate -name fullinstance1
-location /scratch/ttuser/fullinstance1 -daemonport 6824
Creating instance in /scratch/ttuser/fullinstance1 ...
INFO: Mapping files from the installation to
/scratch/ttuser/fullinstance1/install
```

```
NOTE: The TimesTen daemon startup/shutdown scripts have not been installed.

The startup script is located here :
        '/scratch/ttuser/fullinstance1/startup/tt_fullinstance1

Run the 'setuproot' script :
        /scratch/ttuser/fullinstance1/bin/setuproot -install
This will move the TimesTen startup script into its appropriate location.

The 22.1 Release Notes are located here :
  '/installation_dir/tt22.1.1.10.0/README.html'
```

You have successfully created the TimesTen full instance. Proceed to "Review the instance home directory and subdirectories" for next steps.

Create an instance interactively with the -record option

When creating an instance interactively, you can use the `-record` option to record the interactive prompts and responses into a batch file, which you can then use as a template for creating additional instances in batch mode. See "Create an instance from information provided in a batch file " for details.

This example illustrates how to use the `ttInstanceCreate` utility with the `-record` option.

```
% installation_dir/tt22.1.1.10.0/bin/ttInstanceCreate -record
/swdir/TimesTen/ttinstances/instancecreatebatch
```

The batch file that is created, `instancecreatebatch`, contains this input for `ttInstanceCreate`:

```
Please choose an instance name for this installation:ttuser1instanceint
Is this correct:y
Where would you like to install the ttuser1instanceint instance of
TimesTen:/swdir/TimesTen/ttinstances
Please enter a unique port number for the TimesTen daemon (<CR>=list):27100
Please enter a value for TNS_ADMIN (s=skip):s
What is the TCP/IP port number that you want the TimesTen Server to listen
on:27101
Would you like to use TimesTen Replication with Oracle Clusterware:n
```

Create an instance from information provided in a batch file

You can use the `ttInstanceCreate -batch` option to use a batch file with instructions for creation of an instance. You created such a file by running `ttInstanceCreate` with the `-record` option. See "Create an instance interactively with the -record option " for information.

This example first uses a sample batch file that was created with the `-record` option. It then uses this sample batch file (`instancecreatebatch`) to create the instance.

```
Please choose an instance name for this installation:ttuser1instancebat
Is this correct:y
Where would you like to install the ttuser1instancebat instance of
TimesTen:/swdir/TimesTen/ttinstances
Please enter a unique port number for the TimesTen daemon (<CR>=list):29100
Please enter a value for TNS_ADMIN (s=skip):s
What is the TCP/IP port number that you want the TimesTen Server to listen
```

```
on:29101
Would you like to use TimesTen Replication with Oracle Clusterware:n
```

Then to create the instance:

```
% installation_dir/tt22.1.1.10.0/bin/ttInstanceCreate -batch /swdir/TimesTen/
ttinstances/instancecreatebatch
```

This results in the following output from `ttInstanceCreate`:

```
NOTE: Each TimesTen instance is identified by a unique name.
      The instance name must be a non-null alphanumeric string, not longer
      than 255 characters.

Instance name will be 'ttuser1instancebat'.
Creating instance in /swdir/TimesTen/ttinstances/ttuser1instancebat ...
TCP port 6624 is in use!

NOTE: If you are configuring TimesTen for use with Oracle Clusterware, the
      daemon port number must be the same across all TimesTen installations
      managed within the same Oracle Clusterware cluster.

** The default daemon port (6624) is already in use or within a range of 8
ports of an existing TimesTen instance. You must assign a unique daemon port
number for this instance. This installer will not allow you to assign another
instance a port number within a range of 8 ports of the port you assign below.

NOTE: All installations that replicate to each other must use the same daemon
      port number that is set at installation time. The daemon port number can
      be verified by running 'ttVersion'.

INFO: installation group ownership (ttVersion) is 'timesten'

In order to use the cache features in any TimesTen databases
created within this instance, you must set a value for the TNS_ADMIN
environment variable. It can be left blank, and a value can be supplied later
using <install_dir>/bin/ttInstanceModify.

NOTE: The TimesTen daemon startup/shutdown scripts have not been installed.

The startup script is located here :
    '/swdir/TimesTen/ttinstances/ttuser1instancebat/startup/tt_ttuser1instancebat'

Run the 'setuproot' script :
        /swdir/TimesTen/ttinstances/ttuser1instancebat/bin/setuproot -install
This will move the TimesTen startup script into its appropriate location.

The 22.1 Release Notes are located here :
  '/swdir/TimesTen/ttinstallations/fullinstall/tt22.1.1.10.0/README.html'

Starting the daemon ...
TimesTen Daemon (PID: 7725, port: 29100) startup OK.
```

You have successfully created the TimesTen full instance. Proceed to "Review the instance home directory and subdirectories" for next steps.

Create an instance interactively for Oracle Clusterware

You can run the `ttInstanceCreate` utility interactively to create an instance and enable TimesTen Replication with Oracle Clusterware.

When prompted for the path to the Oracle Clusterware installation, you can either provide it or skip it. If you skip it, you can specify the path later using the `ttInstanceModify -crs` option. See "Change the Oracle Clusterware configuration for an instance" for information.

This example provides the path to the Oracle Cluster installation as `/u01/app/`*`crs_releasedir`*`/grid`, where *`crs_releasedir`* is a variable for the CRS release (for example, if the CRS release is `19.1.0`, you could substitute, `19.1.0`, such that the path would be `/u01/app/19.1.0/grid`).

There must be a valid Oracle Clusterware installation in the specified directory.

In this example, either the values entered or the defaults chosen are shown in bold.

```
% installation_dir/tt22.1.1.10.0/bin/ttInstanceCreate

NOTE: Each TimesTen instance is identified by a unique name.
      The instance name must be a non-null alphanumeric string, not longer
      than 255 characters.

Please choose an instance name for this installation? [ tt221 ] crsinstance
Instance name will be 'crsinstance'.
Is this correct? [ yes ] yes
Where would you like to install the crsinstance instance of TimesTen?
[ /home/oracle ] /u02/ttinstances
Creating instance in /u02/ttinstances/crsinstance ...

NOTE: If you are configuring TimesTen for use with Oracle Clusterware, the
      daemon port number must be the same across all TimesTen installations
      managed within the same Oracle Clusterware cluster.

NOTE: All installations that replicate to each other must use the same daemon
      port number that is set at installation time. The daemon port number can
      be verified by running 'ttVersion'.

The default port number is 6624.

Do you want to use the default port number for the TimesTen daemon? [ yes ]
The daemon will run on the default port number (6624).
INFO: installation group ownership (ttVersion) is 'oinstall'

In order to use the cache features in any TimesTen databases
created within this instance, you must set a value for the TNS_ADMIN
environment variable. It can be left blank, and a value can be supplied later
using <install_dir>/bin/ttInstanceModify.

Please enter a value for TNS_ADMIN (s=skip)? [  ] s
What is the TCP/IP port number that you want the TimesTen Server to listen on?
[ 6625 ]

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

A Clusterware installation was detected in /u01/app/crs_releasedir/grid

Please provide the path to the Oracle Clusterware installation on this machine
(s=skip)? [ /u01/app/crs_releasedir/grid ]

NOTE: The TimesTen Clusterware agent port must be the same on all nodes
      of the cluster. Please refer to the TimesTen documentation for
      additional information.
```

```
Please enter a port number for the TimesTen Clusterware agent? [ 3574 ]

Executing '/u01/app/crs_releasedir/grid/bin/olsnodes' ...
Oracle Clusterware is currently configured on the following nodes :

1. tthost1
2. tthost2

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 ]

NOTE: The TimesTen daemon startup/shutdown scripts have not been installed.

The startup script is located here :
        '/u02/ttinstances/crsinstance/startup/tt_crsinstance'

Run the 'setuproot' script :
        /u02/ttinstances/crsinstance/bin/setuproot -install
This will move the TimesTen startup script into its appropriate location.

The 22.1 Release Notes are located here :
  '/u02/tt22.1.1.10.0/README.html'

Starting the daemon ...
TimesTen Daemon (PID: 11839, port: 6624) startup OK.
```

> **Note:**
>
> The Oracle Clusterware installation was detected by TimesTen. The location merely had to be confirmed, not entered.

Review the instance home directory and subdirectories

You can review the instance home directory and subdirectories for informational purposes. When you create an instance, each instance includes these subdirectories under *$TIMESTEN_HOME* (Not all of the subdirectories are included in this list):

- `bin`: TimesTen utilities and executables tailored and specific to the instance

  This includes `ttenv.sh` (or `ttenv.csh`), which sets environment variables appropriately for the TimesTen environment for your session, and `setuproot.sh`, which can be run as root to cause data instances to be automatically started whenever the operating system reboots.

  Note that `ttenv` also puts the `bin` directory in your path.
- `conf`: Contains the `timesten.conf` file, which is the TimesTen instance configuration file
- `diag`: Diagnostic output, including the daemon log and error log
- `info`: Working directory of the TimesTen daemon, containing persistent state about the TimesTen instance
- `install`: Symbolic link referencing the installation associated with this instance.

- `plsql`: Contains this subdirectory:
  - `utl_file_dir`: The only directory that can be read from or written to by PL/SQL blocks using the `UTL_FILE` package
- `startup`: Contains a script that can be added to `/etc/init.d` to cause the instance to be automatically started at system startup and stopped at system shutdown.

Configure Linux kernel parameters

This section discusses the Linux kernel parameters that you must configure after creating your TimesTen instance:

- Configure shmmax and shmall
- Configure HugePages
- Modify the memlock settings
- Set the semaphore values
- Set the SHMMNI parameter

> **Note:**
>
> If you have chosen systemd to automatically manage the TimesTen daemon, see "Using systemd to Manage a TimesTen Service" for details on configuring kernel parameters for systemd.

Configure shmmax and shmall

You must configure Linux shared memory so that the maximum size of a shared memory segment (the `shmmax` memory kernel parameter) is large enough to contain the size of the total shared memory segment for the database. In TimesTen Classic, the entire database resides in a single shared memory segment. There is also a second memory segment used for PL/SQL.

On Linux, a shared memory segment consists of pages, where the default page size is normally 4 kB (4096 bytes). You can verify the default page size by running the `getconf PAGESIZE` command:

```
% getconf PAGESIZE
4096
```

Configure these shared memory kernel parameters to control the size of the shared memory segment:

- `shmmax`: The maximum size of a single shared memory segment expressed in bytes. The value must be large enough to accommodate the size of the total shared memory segment for the database.
- `shmall`: The total size of shared memory segments system wide expressed in pages. The value is expressed in multiples of the page size (4 kB) and `shmall * pagesize` must be greater or equal to the value of `shmmax`. It is recommended that you set the value of `shmall` to less than or equal to the total amount of physical

RAM. To display the total amount of physical memory, run the Linux `cat /proc/meminfo` command.

Use the `ttShmSize` utility to determine the size of the shared memory segment. The `ttShmSize` utility uses the values of the `PermSize`, the `TempSize`, the `LogBufMB` and the `Connections` connection attributes (for a specified database) to determine this size. See "ttShmSize" in *Oracle TimesTen In-Memory Database Reference* for details on the `ttShmSize` utility and see "PermSize", "TempSize", "LogBufMB", and "Connections" in *Oracle TimesTen In-Memory Database Reference* for details on each connection attribute.

For example, use the `ttShmSize` utility with the `-connStr` option to determine the size of the shared memory segment using the `database1` DSN. Supply a `PermSize` value of 32GB (32768 MB), a `TempSize` value of 4 GB (4096 MB), a `LogBufMB` value of 1 GB (1024 MB) and a `Connections` value of 2048.

```
% ttShmSize -connstr
"DSN=database1;PermSize=32768;TempSize=4096;LogBufMB=1024;Connections=2048"
The required shared memory size is 39991547720 bytes.
```

Note:

- The `-connStr` option of the `ttShmSize` utility requires that you have defined a DSN in either the user `.odbc.ini` or the system `sys.odbc.ini` file. You may use any DSN. Note that for any connection attribute not specified in the `-connStr` option, `ttShmSize` uses the setting defined in either the user `.odbc.ini` file or the system `sys.odbc.ini` file for the specified DSN. If the connection attribute is missing from both the `-connStr` option and either the user `.odbc.ini` file or the `sys.odbc.ini` file, `ttShmSize` uses the default value for the connection attribute.
- You can add a DSN to either the user `.odbc.ini` file of the user or the system `sys.odbc.ini` file. For example, to add the `database1` DSN to the user `.odbc.ini` file of the current operating system user:

  ```
  % vi ~/.odbc.ini
  ...
  [database1]
  ```

To size `shmmax` and `shmall`:

1. As the `root` user, edit the `/etc/sysctl.conf` file, modifying `kernel.shmmax` and `kernel.shmall`. Set `shmmax` to 39,991,547,720 bytes and `shmall` to 9,763,561 pages, which is `shmmax/pagesize`.

   ```
   % sudo vi /etc/sysctl.conf
   ...
   kernel.shmmax=39991547720
   kernel.shmall=9763561
   ```

2. To reload the settings from the modified `/etc/sysctl.conf` file:

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

3. Run the Linux `ipcs lm` command to display the current `shmmax` and `shmall` settings. The `max seg size (kbytes)` is the `shmmax` value and the `max total shared memory (kbytes)` is the value of `shmall` times `max number of segments`.

```
% ipcs -lm

------ Shared Memory Limits --------
max number of segments = 4096
max seg size (kbytes) = 39054246
max total shared memory (kbytes) = 39054246
min seg size (bytes) = 1
```

> **Note:**
>
> - The settings for `shmmax` and `shmall` in these examples can be increased if there are other applications that require them to be greater.
> - If you are unsure of the size of your database, you can set `shmmax` and `shmall` to correspond to a percentage of the size of physical memory (such as 80%).

Configure HugePages

You can configure `HugePages` for more efficient memory management.

Once configured, the memory allocated for `HugePages` is taken from the total RAM on the Linux host and is not available for any other use. In addition, the `HugePages` memory segment is automatically locked and cannot be swapped to disk.

To configure `HugePages`, you need to know:

- The maximum size of the shared memory segment for the database
- The `HugePages` page size on your Linux host
- The group ID of the instance administrator

Using the examples in the "Configure shmmax and shmall" section, where the value of `shmmax` value is 39,054,246 kB, and the "Create the TimesTen Users Group " section, where the group ID of the `instanceadmin` user is 10000:

- The size of the total shared memory segment is is 39,054,246 kB.
- The `HugePages` page size is 2048 KB. (This value is fixed for each platform and is not configurable.)

  To determine the HugePages page size, run the Linux `cat /proc/meminfo|grep Hugepagesize` command:

  ```
  % cat /proc/meminfo | grep Hugepagesize
  Hugepagesize:       2048 kB
  ```

- The group ID is 10000.

  To determine the group ID of the instance administrator, log in as the `instanceadmin` user, and run the Linux `id` command:

```
% id
uid=55000(instanceadmin) gid=10000(g10000)groups=10000(g10000)
```

To configure `HugePages`:

1. Determine the number of `HugePages` by dividing the size of the total shared memory segment (expressed in MB) by the value of `Hugepagesize` (expressed in MB). In this example, the total shared memory segment is 39,054,246 kB (~38,138 MB) and the `Hugepagesize` value is 2,048 kB (2 MB):

   ```
   38138 MB/ 2 MB = 19069
   ```

2. As the root user, edit the `/etc/sysctl.conf` file, and set `vm.nr_hugepages` to the number of `HugePages` and set `vm.hugetlb_shm_group` to the group ID of the instance administrator. The latter setting restricts access to `HugePages` to members of the group.

   ```
   % sudo vi /etc/sysctl.conf
   ...
   vm.nr_hugepages=19069
   vm.hugetlb_shm_group=10000
   ```

3. Reload the settings from the modified `/etc/sysctl.conf` file:

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

4. To verify that you have configured `HugePages` correctly, run:

   ```
   % cat /proc/meminfo | grep HugePages
   HugePages_Total:   19069
   HugePages_Free:    19069
   ...
   ```

Note:

- Because `HugePages` must be allocated in contiguous available memory space, the requested allocation may not be granted, or may be only partially granted, until after the host is restarted. Check the `HugePages_Total` and `HugePages_Free` values from `/proc/meminfo`. Restarting grants the full allocation, assuming enough memory is available in the host.
- The TimesTen PL/SQL shared memory segment consumes some of the configured HugePages allocation, determined by the value of the `PLSQL_MEMORY_SIZE` connection attribute. See "PLSQL_MEMORY_SIZE" in the *Oracle TimesTen In-Memory Database Reference* for more information.
- On Linux, the `HugePages` segment is automatically locked such that the memory segment is not a candidate to be swapped to disk. Therefore, if you configure `HugePages`, you do not need to set the `MemoryLock` connection attribute.

Modify the memlock settings

The `memlock` entries in the `/etc/security/limits.conf` file control the amount of memory a user can lock. These entries are set at the system level and are different than the `MemoryLock` connection attribute setting.

> **Note:**
>
> If you are using systemd to manage a TimesTen service, you must set the `LimitMEMLOCK` setting in the systemd service unit configuration file. systemd ignores the values in the `/etc/security/limits.conf` file. See "Using systemd to Manage a TimesTen Service" for details.

If `HugePages` are configured, the `memlock` values must be large enough to accommodate the size of the shared memory segment or the database will not be loaded into memory.

For example, for the `instanceadmin` user, assuming a total shared memory segment size of 39,054,246 kB, set the `memlock` entries to `39054246`:

1. As the `root` user, edit the `/etc/security/limits.conf` file, and set the `memlock` entries to 39,054,246 kB for the `instanceadmin` user. This value indicates the total amount of memory the `instanceadmin` user can lock.

   ```
   % sudo vi /etc/security/limits.conf
   ...
   instanceadmin soft    memlock 39054246
   instanceadmin hard    memlock 39054246
   ```

2. As the `instanceadmin` user, log out and log in again for the changes to take effect.

Set the semaphore values

TimesTen has an upper bound on the maximum number of connections to the database. The database connections consist of:

- User connections: established by user applications
- System connections: established internally by TimesTen (set at 48 connections)
- Other required connections (set at 107 connections)

Each of these connections is assigned one semaphore, such that the total semaphores for a database are:

```
Total semaphores = user connections (N) + system connections (48) +
                   other required connections (107)

Total semaphores = N + 155
```

The semaphore settings are located in the `kernel.sem` configuration directive in `/etc/sysctl.conf`:

```
kernel.sem = SEMMSL SEMMNS SEMOPM SEMMNI
```

where:

- `SEMMSL` is the maximum number of semaphores per array. This value is related to the maximum number of connections. Configure this value to be `155` plus the number of simultaneous user connections.
- `SEMMNS` is the maximum number of semaphores system wide. Use the formula `SEMMNS` = (`SEMMNI * SEMMSL`) as a guideline. However, in practice, `SEMMNS` can be much less than `SEMMNI * SEMMSL`.

- SEMOPM is the maximum number of operations for each semop call.
- SEMMNI is the maximum number of arrays.

Follow these steps to configure the SEMMSL and the SEMMNI settings. Ensure that the user is root:

1. View the existing kernel parameter settings:

```
# /sbin/sysctl -a | grep kernel.sem
kernel.sem = 2500 320000 1000 1280
```

2. Edit the /etc/sysctl.conf file, changing SEMMSL (the first of the four values in kernel.sem) to 155 plus the number of simultaneous user connections.

   In this example, assume the number of simultaneous user connections is 4000. Set the SEMMSL value to 4155 (=4000+155) or greater.

   In addition, change SEMMNI (the last of the four values in kernel.sem) to the value of SEMMSL plus the number of TimesTen-specific shared memory segments. (These shared memory segments include the TimesTen database and PL/SQL.) In this example, set SEMMNI to 4157 (=4155+2) or greater.

   > **Note:**
   >
   > TimesTen uses the value of the Connections first connection attribute to determine the maximum number of simultaneous user connections that can be connected to the TimesTen database. TimesTen returns an error if the number of simultaneous user connections exceeds this value. The default value is the lesser of 2000 or (SEMMSL - 155). See "Defining Server DSNs for TimesTen Server on a Linux or UNIX system" in the *Oracle TimesTen In-Memory Database Operations Guide* for information on setting the Connections attribute. Also see "Connections" in the *Oracle TimesTen In-Memory Database Reference* for information on the Connections first connection attribute.

```
# vi /etc/sysctl.conf
...
...
kernel.sem = 4155 400000 2000 4157
```

3. Reload the settings from the modified /etc/sysctl.conf file.

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

> **Note:**
>
> If you are using replication, the Linux platform for each host on which the master databases reside must have similar kernel settings for shared memory and semaphores. Specifically, the SEMMSL and SEMMNI settings must be large enough on all hosts that participate in an active standby replication scheme before the duplication is performed. In the event of a failover, the standby must be able to accommodate the active.

Set the SHMMNI parameter

The SHMMNI value controls the number of shared memory segments that the host can create simultaneously. TimesTen creates a shared memory segment for the TimesTen database and a shared memory segment for PL/SQL. In addition, there is a small shared memory segment that is allocated for the duration of each client/server connection. This shared memory segment is created at connect time and is destroyed when the client/server connection is disconnected from the TimesTen database. There is one shared memory segment per client/server connection.

You must configure the SHMMNI parameter setting to account for the number of client/server connections. Set SHMMNI to a value that is greater than number of expected client/server connections. (Ensure to also take into account the TimesTen shared memory segment, the PL/SQL shared memory segment, and other programs that use shared memory.) As an example, if you expect there to be 8000 client/server connections, an appropriate value is 9000 or greater. A value of 9000 or greater is appropriate as TimesTen has system connections that are not included in the client/server connections count.

Follow these steps to configure the SHMMNI setting. Ensure that the user is root:

1. View the existing SHMMNI parameter setting.

```
# /sbin/sysctl -a | grep shmmni
kernel.shmmni = 4096
```

2. Edit the /etc/sysctl.conf file, changing kernel.shmmni to a value that is greater than the number of client/server connections. This example sets kernel.shmmni to 9000.

```
 vi /etc/sysctl.conf
...
...
kernel.shmmni = 9000
```

3. Reload the settings from the modified /etc/sysctl.conf file.

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

> **Note:**
>
> If you are using replication, the Linux platform for each host on which the master databases reside must have a similar SHMMNI kernel setting. Specifically, the SHMMNI setting must be large enough on all hosts that participate in an active standby replication scheme before the duplication is performed. In the event of a failover, the standby must be able to accommodate the active.

Start an instance automatically at system startup with System V init scripts

You have the option of having the TimesTen daemon started automatically at system startup and stopped automatically at system shutdown.

> **Note:**
>
> If you do not choose this option, the instance administrator controls the TimesTen daemon at all times.

To enable this option, the `root` user runs the TimesTen `setuproot` script with the `-install` option. The script is located in the *timesten_home*`/bin` directory. This script copies a set of scripts to the `/etc/init.d` directory.

Recall that there is information about this in the `ttInstanceCreate` output when it creates an instance. For example, for an instance named `myinstance`:

```
NOTE: The TimesTen daemon startup/shutdown scripts have not been installed.

The startup script is located here :
        '/swdir/TimesTen/ttinstances/myinstance/startup/tt_myinstance'

Run the 'setuproot' script :
        /swdir/TimesTen/ttinstances/myinstance/bin/setuproot -install
This will move the TimesTen startup script into its appropriate location.
```

The `setuproot` script operates according to the current setting of the `TIMESTEN_HOME` environment variable, which indicates the instance home directory.

In this example, *timesten_home*`/bin` is the current directory of the `root` user:

```
# echo $TIMESTEN_HOME
/swdir/TimesTen/ttinstances/myinstance
# $TIMESTEN_HOME/bin/setuproot -install
Would you like to install the TimesTen daemon startup scripts into /etc/init.d?
[ yes ]
Copying /swdir/TimesTen/ttinstances/myinstance/startup/tt_myinstance to
 /etc/init.d

Successfully installed the following scripts :
/etc/init.d/tt_myinstance
/etc/rc.d/rc0.d/K45tt_myinstance
/etc/rc.d/rc1.d/K45tt_myinstance
/etc/rc.d/rc2.d/S90tt_myinstance
/etc/rc.d/rc3.d/S90tt_myinstance
/etc/rc.d/rc5.d/S90tt_myinstance
/etc/rc.d/rc6.d/K45tt_myinstance
```

Before you destroy the instance, `root` should run `setuproot -uninstall`.

```
# $TIMESTEN_HOME/bin/setuproot -uninstall
Would you like to uninstall the TimesTen daemon startup scripts in /etc/init.d?
[ yes ]

Successfully deleted the following scripts :
/etc/rc.d/rc0.d/K45tt_myinstance
```

```
/etc/rc.d/rc1.d/K45tt_myinstance
/etc/rc.d/rc2.d/S90tt_myinstance
/etc/rc.d/rc3.d/S90tt_myinstance
/etc/rc.d/rc5.d/S90tt_myinstance
/etc/rc.d/rc6.d/K45tt_myinstance
/etc/init.d/tt_myinstance
```

> **Note:**
>
> If you do not run `setuproot -uninstall` before you destroy the instance, you must manually delete the files that were placed by `setuproot -install`.

Creating a TimesTen client instance

A TimesTen client instance can be used to connect to a database in either TimesTen Scaleout or in TimesTen Classic.

This is an example of creating a client instance. See "Client-only Installations and Instances " for detailed information about client-only installations and instance, including multiple examples.

Navigate to the *installation_dir*`/tt22.1.1.10.0/bin` area and run the `ttInstanceCreate` utility, specifying the `-clientonly` option. User input is shown in bold.

```
% installation_dir/tt22.1.1.10.0/bin/ttInstanceCreate -clientonly

NOTE: Each TimesTen instance is identified by a unique name.
      The instance name must be a non-null alphanumeric string, not longer
      than 255 characters.

Please choose an instance name for this installation? [ tt221 ] clientinstance1
Instance name will be 'clientinstance1'.
Is this correct? [ yes ]
Where would you like to install the clientinstance1 instance of TimesTen?
 [ /home/ttuser ] /scratch/ttuser
Creating instance in /scratch/ttuser/clientinstance1 ...
INFO: Mapping files from the installation to
/scratch/ttuser/clientinstance1/install

In order to use the cache features in any TimesTen databases
created within this instance,
you must set a value for the TNS_ADMIN environment variable.
It can be left blank, and a value can be supplied later
using <install_dir>/bin/ttInstanceModify.

Please enter a value for TNS_ADMIN (s=skip)? [  ] s
The 22.1 Release Notes are located here :
  '/installation_dir/tt22.1.1.10.0/README.html'
```

Modifying an instance on Linux/UNIX

If you are using systemd as a TimesTen service, and you want to modify the instance, see "Using systemd to Manage a TimesTen Service" and "About modifying a TimesTen instance to enable or disable management by systemd" for details.

These sections discuss how to modify instances that do not use systemd as a TimesTen service:

- The ttInstanceModify utility
- Modify an instance from information provided interactively
- Change the daemon port for an instance
- Set or change the path to tnsnames.ora for an instance
- Change the Oracle Clusterware configuration for an instance
- Associate an instance with a different installation (upgrade or downgrade)

For information on modifying an instance with systemd, see Using systemd to Manage a TimesTen Service

The ttInstanceModify utility

The `ttInstanceModify` utility modifies the attributes of an instance. The instance that is modified is the instance that the `$TIMESTEN_HOME` environment variable references. Run the `ttInstanceModify` utility from the *`$TIMESTEN_HOME`*`/bin` directory.

For a full instance, you can modify all of these attributes. For a client-only instance, you can only modify the installation with which the instance is associated (`-install` option):

- TimesTen daemon port
- TimesTen Server port for client/server
- `TNS_ADMIN` location for `tnsnames.ora`
- Oracle Clusterware configuration
- TimesTen installation with which the instance is associated

The utility then updates the `timesten.conf` file accordingly.

> **Note:**
>
> The instance administrator cannot be modified.

Run the `ttInstanceModify` with the `-help` option to list the available options. See "ttInstanceModify" in the *Oracle TimesTen In-Memory Database Reference* for detailed descriptions.

Modify an instance from information provided interactively

You can run `ttInstanceModify` interactively if you do not specify any command-line options. In this case, the utility asks one by one if you want to change any of the supported options.

This example sets a new daemon port and server port. The values entered, or defaults chosen, are shown in bold:

```
% $TIMESTEN_HOME/bin/ttInstanceModify

Instance Info
-------------

Name:           myttinstance
Version:        22.1.1.10.0
Location:       /swdir/TimesTen/ttinstances/myttinstance
Installation:   /swdir/TimesTen/ttinstallations/fullinstall/tt22.1.1.10.0
Daemon Port:    28000
Server Port:    28001

Would you like to change the installation that this instance points to? [ no ]

The daemon for instance 'myttinstance' is currently configured to use port 28000.
Would you like to change this port? [ no ] yes

NOTE: If you are configuring TimesTen for use with Oracle Clusterware, the
      daemon port number must be the same across all TimesTen installations
      managed within the same Oracle Clusterware cluster.

Please enter a unique port number for the TimesTen daemon (<CR>=list)? [ ] 28050
Are you sure you want to change the daemon port number from 28000 to 28050?
[ yes ]

The server for instance 'myttinstance' is currently configured to use port 28001.
Would you like to change this port? [ no ] yes
Please enter a unique port number for the TimesTen server (<CR>=list)? [ ] 28060
Are you sure you want to change the server port number from 28001 to 28060?
[ yes ]

TNS_ADMIN for the instance 'myttinstance' is currently not set.
Would you like to change TNS_ADMIN for this instance? [ no ]
Would you like to configure TimesTen Replication with Oracle Clusterware? [ no ]
NOTE: The daemon will be stopped before changing port numbers.
Would you like to proceed with modifying the instance? [ yes ]
ttDaemonAdmin: daemon is not running
Changing the daemon port number ...
The daemon will now run on port 28050 ...
Changing the server port number ...
The server will now run on port 28060 ...

NOTE: You may need to modify your sys.ttconnect.ini file to
      reflect the new server port. The sys.ttconnect.ini file is usually
      located in your instance's conf directory, $TIMESTEN_HOME/conf.

Restarting the daemon ...
ttDaemonAdmin: daemon is not running
TimesTen Daemon (PID: 3072, port: 28050) startup OK.
```

The `timesten.conf` file is updated accordingly:

```
# TimesTen Instance Configuration File
...
daemon_port=28050
server_port=28060
...
```

> **Note:**
>
> In interactive mode, the daemon is restarted automatically if you set the daemon port or server port, and you are prompted to restart the daemon if you set `TNS_ADMIN`.

Change the daemon port for an instance

You can change the daemon port for an instance.

This example first runs the Linux/UNIX `echo` command to display the instance in which *`$TIMESTEN_HOME`* references. It then reviews the *`$TIMESTEN_HOME`*`/conf/timesten.conf` file to display the current port number. It then runs `ttInstanceModify` to change the port number to `6524`. As a final step, the example shows the port number has been changed in the *`$TIMESTEN_HOME`*`/conf/timesten.conf` file.

```
% cat $TIMESTEN_HOME/conf/timesten.conf
# TimesTen Instance Configuration File
# Created by ttInstanceCreate
# Commented values are default values
hostname=host1
timesten_release=22.1
instance_name=fullinstance1
daemon_port=6624
server_port=6625
admin_user=myadminuser
admin_uid=4738
group_name=timesten
instance_guid=9EEF0277-21C0-45F1-AB63-F0C5F48B6FE0
verbose=1
```

Run `ttInstanceModify` to change the port number to `6524`.

```
% $TIMESTEN_HOME/bin/ttInstanceModify -port 6524
The daemon will now run on port 6524 ...
You must restart the daemon for these changes to take effect.
```

Review the `$TIMESTEN_HOME/conf/timesten.conf` file to ensure the port number is changed.

```
% cat $TIMESTEN_HOME/conf/timesten.conf
# TimesTen Instance Configuration File
# Created by ttInstanceCreate
# Commented values are default values
hostname=host1
timesten_release=22.1
instance_name=fullinstance1
daemon_port=6524
server_port=6625
```

```
admin_user=myadminuser
admin_uid=4738
group_name=timesten
instance_guid=9D37C711-DF86-4007-A959-2AB52DA46035
verbose=1
```

Restart the daemon

```
% ttDaemonAdmin -start
TimesTen Daemon (PID: 11635, port: 6524) startup OK.
```

Set or change the path to tnsnames.ora for an instance

You can use the `ttInstanceModify` utility with the `-tnsadmin` option to set or change the path to the `tnsnames.ora` file. Specify the full path to the directory where the file is located. This is relevant for an application using TimesTen Cache, OCI, Pro*C/C++, or ODP.NET. For TimesTen Cache, this is for access to the Oracle Database. For TimesTen OCI, Pro*C/C++, and

To ensure that the `TNS_ADMIN` setting is read by TimesTen as well as by user applications, also set the `TNS_ADMIN` environment variable.

This example shows how to use the `-tnsadmin` option. The current directory is *`timesten_home`*:

```
% $TIMESTEN_HOME/bin/ttInstanceModify -tnsadmin
/swdir/TimesTen/ttinstances/ttuser1instancelatest/instantclient
The value for TNS_ADMIN in
/swdir/TimesTen/ttinstances/ttuser1instancelatest/conf/timesten.conf
was changed to /swdir/TimesTen/ttinstances/ttuser1instancelatest/instantclient
```

Before you ran `ttInstanceModify`, the *`$TIMESTEN_HOME`*`/conf/timesten.conf` file had no value for `tns_admin`:

```
# TimesTen Instance Configuration File
...
tns_admin=
...
```

After running `ttInstanceModify` with the `-tnsadmin` option, the *`$TIMESTEN_HOME`*`/conf/timesten.conf` file is updated:

```
# TimesTen Instance Configuration File
...
tns_admin=/swdir/TimesTen/ttinstances/ttuser1instancelatest/instantclient
...
```

Change the Oracle Clusterware configuration for an instance

After creating an instance, you can use the `ttInstanceModify -crs` option to change the TimesTen configuration to use Oracle Clusterware for TimesTen Replication and specify the full path to the Oracle Clusterware installation. As shown in the example, there is a prompt to confirm whether you want to configure TimesTen Replication with Oracle Clusterware and to enter or confirm the directory path to the installation. There must be a valid Oracle Clusterware installation in the specified directory.

This example illustrates how to use `ttInstanceModify` with the `-crs` option. The current directory is *timesten_home*. User entries or defaults that are chosen are shown in bold.

This example provides the path to the Oracle Cluster installation as `/u01/app/`*crs_releasedir*`/grid`, where *crs_releasedir* is a variable for the CRS release (for example, if the CRS release is `19.1.0`, you could substitute, `19.1.0`, such that the path would be `/u01/app/19.1.0/grid`).

```
% $TIMESTEN_HOME/bin/ttInstanceModify -crs
Would you like to configure TimesTen Replication with Oracle Clusterware? [ no ]
yes

A Clusterware installation was detected in /u01/app/crs_releasedir/grid

Please provide the path to the Oracle Clusterware installation on this machine
(a=abort)?
[ /u01/app/crs_releasedir/grid ]

NOTE: The daemon port number must be the same across all TimesTen installations
      part of the this Clusterware configuration.

Please enter a port number for the TimesTen Clusterware agent? [ 31006 ]

Executing '/u01/app/crs_releasedir/grid/bin/olsnodes' ...
Oracle Clusterware is currently configured on the following nodes :

1. tthost1
2. tthost2

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 ]
INFO: Modifying /u02/ttinstances/myinstance/conf/cluster.oracle.ini ...

Do you want to restart the daemon using the new configuration? [ yes ]
Restarting the daemon ...
ttDaemonAdmin: daemon is not running
TimesTen Daemon (PID: 30345, port: 31000) startup OK.
TimesTen Replication with Oracle Clusterware has been configured.

NOTE: Please run ttInstanceModify for all other TimesTen instances which are
      part of the this Clusterware configuration.
```

> **Note:**
>
> The Oracle Clusterware installation was detected by TimesTen. The location merely had to be confirmed, not entered.

Associate an instance with a different installation (upgrade or downgrade)

You can use the `ttInstanceModify -install` option to associate the instance with a different TimesTen installation. This is typically used to upgrade the instance to a new maintenance or

patch release. It *cannot* be used to upgrade to a new major release (such as going from 18.1 to 22.1).

This example upgrades from the `ttinstall` installation to the `ttinstalllatest` installation. The current directory is *`timesten_home`* for the `ttuser1instance` instance.

Optional: Before running *`$TIMESTEN_HOME`*`/bin/ttInstanceModify`, you can review the *`timesten_home`*`/install` directories and files to gain an understanding of the symbolic links that are defined.

Navigate to the *`timesten_home`*`/install` directory and run the Linux `ls -l` command to list the directories and the files within this directory. Note that symbolic links to the installation directory are defined for most of the files and the directories within this *`timesten_home`*`/install` directory. (In this example, `swdir/TimesTen/ttinstallations/ttinstall/tt22.1.1.10.0` is the installation directory.)

```
% cd timesten_home/install
% ls -l
...
lrwxrwxrwx 1 ttuser1 timesten   56 Jun 28 11:13 3rdparty ->
/swdir/TimesTen/ttinstallations/ttinstall/tt22.1.1.10.0/3rdparty
lrwxrwxrwx 1 ttuser1 timesten   52 Jun 28 11:13 PERL ->
/swdir/TimesTen/ttinstallations/ttinstall/tt22.1.1.10.0/PERL
lrwxrwxrwx 1 ttuser1 timesten   59 Jun 28 11:13 README.html ->
/swdir/TimesTen/ttinstallations/ttinstall/tt22.1.1.10.0/README.html
lrwxrwxrwx 1 ttuser1 timesten   51 Jun 28 11:13 bin ->
/swdir/TimesTen/ttinstallations/ttinstall/tt22.1.1.10.0/bin
lrwxrwxrwx 1 ttuser1 timesten   52 Jun 28 11:13 grid ->
/swdir/TimesTen/ttinstallations/ttinstall/tt22.1.1.10.0/grid
lrwxrwxrwx 1 ttuser1 timestten   55 Jun 28 11:13 include ->
/swdir/TimesTen/ttinstallations/ttinstall/tt22.1.1.10.0/include
lrwxrwxrwx 1 ttuser1 timesten   52 Jun 28 11:13 info ->
/swdir/TimesTen/ttinstallations/ttinstall/tt22.1.1.10.0/info
drwxr-x--- 2 ttuser1 timesten 4096 Jun 28 11:13 lib
lrwxrwxrwx 1 ttuser1 timesten   55 Jun 28 11:13 network ->
/swdir/TimesTen/ttinstallations/ttinstall/tt22.1.1.10.0/network
lrwxrwxrwx 1 ttuser1 timesten   51 Jun 28 11:13 nls ->
/swdir/TimesTen/ttinstallations/ttinstall/tt22.1.1.10.0/nls
lrwxrwxrwx 1 ttuser1 timesten   61 Jun 28 11:13 oraclescripts ->
/swdir/TimesTen/ttinstallations/ttinstall/tt22.1.1.10.0/oraclescripts
lrwxrwxrwx 1 ttuser1 timesten   53 Jun 28 11:13 plsql ->
/swdir/TimesTen/ttinstallations/ttinstall/tt22.1.1.10.0/plsql
lrwxrwxrwx 1 ttuser1 timesten   55 Jun 28 11:13 startup ->
/swdir/TimesTen/ttinstallations/ttinstall/tt22.1.1.10.0/startup
lrwxrwxrwx 1 ttuser1 timesten   55 Jun 28 11:13 support ->
/swdir/TimesTen/ttinstallations/ttinstall/tt22.1.1.10.0/support
drwxr-x--- 3 ttuser1 timesten 4096 Jun 28 11:13 ttoracle_home
```

Now run *`$TIMESTEN_HOME`*`/bin/ttInstanceModify -install` to associate the `ttuser1instance` instance with the new installation (`/swdir/TimesTen/ttinstallations/ttinstalllatest/22.1.1.11.0`, in this example).

```
% $TIMESTEN_HOME/bin/ttInstanceModify
/swdir/TimesTen/ttinstallations/ttinstalllatest/22.1.1.11.0

Instance Info (UPDATED)
-----------------------

Name:           ttuser1instance
Version:        22.1.1.11.0
```

```
Location:       /swdir/TimesTen/ttinstances/ttuser1instance
Installation:   /swdir/TimesTen/ttinstallations/ttinstalllatest/22.1.1.11.0
Daemon Port:    21000
Server Port:    21001

The instance ttuser1instance now points to the installation in
/swdir/TimesTen/ttinstallations/ttinstalllatest/22.1.1.11.0
```

After running `ttInstanceModify`, note the symbolic links within the *$TIMESTEN_HOME*`/install` directory point to the new installation directory.

```
% cd $TIMESTEN_HOME/install
% ls -l
...
lrwxrwxrwx 1 ttuser1 timesten   55 Jun 28 13:07 3rdparty ->
/swdir/TimesTen/ttinstallations/ttinstalllatest/tt/3rdparty
lrwxrwxrwx 1 ttuser1 timesten   51 Jun 28 13:07 PERL ->
/swdir/TimesTen/ttinstallations/ttinstalllatest/tt22.1.1.11.0/PERL
lrwxrwxrwx 1 ttuser1 timesten   58 Jun 28 13:07 README.html ->
/swdir/TimesTen/ttinstallations/ttinstalllatest/tt22.1.1.11.0/README.html
lrwxrwxrwx 1 ttuser1 timesten   50 Jun 28 13:07 bin ->
/swdir/TimesTen/ttinstallations/ttinstalllatest/tt22.1.1.11.0/bin
lrwxrwxrwx 1 ttuser1 timesten   51 Jun 28 13:07 grid ->
/swdir/TimesTen/ttinstallations/ttinstalllatest/tt22.1.1.11.0/grid
lrwxrwxrwx 1 ttuser1 timesten   54 Jun 28 13:07 include ->
/swdir/TimesTen/ttinstallations/ttinstalllatest/tt22.1.1.11.0/include
lrwxrwxrwx 1 ttuser1 timesten   51 Jun 28 13:07 info ->
/swdir/TimesTen/ttinstallations/ttinstalllatest/tt22.1.1.11.0/info
lrwxrwxrwx 1 ttuser1 timesten   57 Jun 28 13:07 kubernetes ->
/swdir/TimesTen/ttinstallations/ttinstalllatest/tt22.1.1.11.0/kubernetes
drwxrwxr-x 2 ttuser1 timesten 4096 Jun 28 13:07 lib
lrwxrwxrwx 1 ttuser1 timesten   54 Jun 28 13:07 network ->
/swdir/TimesTen/ttinstallations/ttinstalllatest/tt22.1.1.11.0/network
lrwxrwxrwx 1 ttuser1 timesten   50 Jun 28 13:07 nls ->
/swdir/TimesTen/ttinstallations/ttinstalllatest/tt22.1.1.11.0/nls
lrwxrwxrwx 1 ttuser1 timesten   60 Jun 28 13:07 oraclescripts ->
/swdir/TimesTen/ttinstallations/ttinstalllatest/tt22.1.1.11.0/oraclescripts
lrwxrwxrwx 1 ttuser1 timesten   52 Jun 28 13:07 plsql ->
/swdir/TimesTen/ttinstallations/ttinstalllatest/tt22.1.1.11.0/plsql
lrwxrwxrwx 1 ttuser1 timesten   54 Jun 28 13:07 startup ->
/swdir/TimesTen/ttinstallations/ttinstalllatest/tt22.1.1.11.0/startup
lrwxrwxrwx 1 ttuser1 timesten   54 Jun 28 13:07 support ->
/swdir/TimesTen/ttinstallations/ttinstalllatest/tt22.1.1.11.0/support
drwxrwxr-x 3 ttuser1 timesten 4096 Jun 28 13:07 ttoracle_home
```

Destroying an instance on Linux/UNIX

These sections discuss how to destroy an instance using the `ttInstanceDestroy` utility:

- The ttInstanceDestroy utility
- Destroy an instance from information provided interactively

The ttInstanceDestroy utility

The `ttInstanceDestroy` utility destroys an existing instance. The instance that will be destroyed is based on the current setting of the `TIMESTEN_HOME` environment variable, which indicates the instance home directory. Run the `ttInstanceDestroy` utility that is located in the

`bin` directory of the associated installation. (For example, *`installation_dir`*`/tt22.1.1.10.0/bin`.)

Only the instance administrator (the user who created the instance) can destroy the instance.

> **Note:**
>
> There can be no existing connections to databases associated with the instance you are destroying.

Run the `ttInstanceDestroy` utility with the `-help` option to list the available options. See "ttInstanceDestroy" in *Oracle TimesTen In-Memory Database Reference* for detailed descriptions.

> **Note:**
>
> The `-force` option completes the operation without prompting for confirmation.

The `ttInstanceDestroy` utility completes these steps:

- Provides a reminder that if the TimesTen startup scripts for the instance were installed (by `root`), then they should be uninstalled (also by `root`) before the instance is destroyed.

  This refers to the `setuproot -install` and `setuproot -uninstall` commands. See "Start an instance automatically at system startup with System V init scripts" for information.
- Asks for confirmation to destroy the instance (unless `-force` is specified).
- Frees up the ports assigned to the instance.

Destroy an instance from information provided interactively

Use the `ttInstanceDestroy` utility to destroy the instance interactively. Run the utility from the `installation_dir/tt22.1.1.10.0/bin` directory (the installation directory for the instance). Recall that the instance home directory (*`timesten_home`*) contains a subdirectory (`/install`) that is a symbolic link to the top-level directory of the instance's associated installation. See "Instance home" for information.

This example illustrates how to use the `ttInstanceDestroy` utility to destroy an instance.

```
% installation_dir/tt22.1.1.10.0/bin/ttInstanceDestroy

** WARNING **

   The uninstallation has been executed by a non-root user.
   If the TimesTen daemon startup scripts were installed,
   you must run $TIMESTEN_HOME/bin/setuproot -uninstall
```

```
   to remove them. If you proceed with this uninstallation, you
   will have to remove the startup scripts manually.

** WARNING **

   All files in the directory:

   /scratch/ttuser/fullinstance1

   will be removed, including any file that you or other users
   may have created.

   Are you sure you want to completely remove this instance? [ yes ] yes
   Installation will remove all the files from /scratch/ttuser/fullinstance1.
   Do you want to continue? [ yes ]

NOTE: /scratch/ttuser/fullinstance1/info contains information
related to the data store that have been created with this release.
If you remove /scratch/ttuser/fullinstance1/info
you will no longer be able to access your data stores,
nor would you be able to restore nor migrate your data.


NOTE: /scratch/ttuser/fullinstance1/conf contains information
related to the instance configuration.
/scratch/ttuser/fullinstance1 Removed
The TimesTen instance fullinstance1 has been destroyed.
```

Copying an installation on Linux/UNIX

You can copy an installation to another host by copying the installation ZIP file from the source host to the target host and unzip it on that host. You can also use any suitable mechanism, such as the ZIP utility, to bundle an installation on the source host and then extract it into another location or onto another host. This might be useful, for example, if the original TimesTen distribution is not conveniently available. The copied installation is valid as long as the file permissions are maintained appropriately.

This example uses the ZIP utility to copy an installation between hosts:

1. On the originating host, from the installation directory (the directory that the installation was extracted into), create a ZIP file containing the installation (the top-level `tt22.1.1.10.0` directory and its contents).

   ```
   % zip my2211100install.zip tt22.1.1.10.0
   ```

2. Transport the ZIP file to the target host.
3. On the target host, from the directory above the desired location of the installation, extract the ZIP file:

   ```
   % unzip my2211100install.zip
   ```

   This will create the `tt22.1.1.10.0` directory and extract the installation there.

On the target host, optionally run `ttInstallationCheck` to verify the new copy of the installation.

From the directory you copied the installation into:

```
% installation_dir/tt22.1.1.10.0/bin/ttInstallationCheck
This installation has been verified.
```

Deleting an installation on Linux/UNIX

This section describes how to delete an installation.

Uninstall TimesTen

To uninstall, the user who installed TimesTen (unzipped the distribution) performs these steps:

1. Ensures that there are no Timesten instances, databases or important files located under the TimesTen installation.
2. Ensures that no TimesTen instances use the installation. (There is no automated way to do this, as TimesTen does not maintain any inventory of TimesTen installations or instances.)

   If there are instances that use the installation, then before the installation is deleted, the instances must be either modified to point to a different installation or deleted. See "Associate an instance with a different installation (upgrade or downgrade)" for details.
3. Grants write permission to the installation, such as through the `chmod -R` command. For example, from the installation directory where a TimesTen tt22.1.1.10.0 installation was extracted:

   ```
   % chmod -R 750 installation_dir/tt22.1.1.10.0
   ```
4. Deletes the `tt22.1.1.10.0` directory and all files and subdirectories that it contains, such as through the `rm -rf` command. From the installation directory:

   ```
   % rm -rf installation_dir/tt22.1.1.10.0
   ```
5. If the installation directory is empty, optionally deletes it.

Complete example for installing and uninstalling TimesTen Classic

This example provides the procedure for creating an installation and for working with instances. This example assumes you have chosen the System V init mechanism. See About creating an instance on Linux/UNIX for details on the System V init mechanism.

1. Create the full installation
2. Create the full instance
3. Source the environment variables
4. Configure Linux kernel parameters
5. Run the ttStatus utility
6. Create and connect to the database
7. Run ttStatus to show connections
8. Modify the full instance
9. Destroy the full instance

10. Delete the installation

Create the full installation

This example creates a full installation from the `timesten2211100.server.linux8664.zip` distribution.

The operating system user that you designated as the instance administrator first creates the installation by downloading the TimesTen distribution on the host that will contain the instance. For example, download `timesten2211100.server.linux8664.zip`.

This example then shows the steps to perform once the instance administrator downloads the distribution. See "Creating an installation on Linux/UNIX" for more information.

The example creates the `installation1` installation directory and unzips the distribution into that directory.

```
% mkdir installation1
% cd installation1
% unzip /timesten2211100.server.linux8664.zip
[...UNZIP OUTPUT...]
```

Create the full instance

Run the `ttInstanceCreate` utility interactively to create a full instance from the installation in `/installation1`. The `/installation1` directory is referred to as `/installation_dir`. See "Creating an instance on Linux/UNIX: Basics" for information.

This example illustrates how to create a full instance interactively. Navigate to the `/bin` area of the installation directory and run the `ttInstanceCreate` utility located in that directory. The `ttInstanceCreate` utility must be run from the installation directory. User input is shown in bold.

```
% installation_dir/tt22.1.1.10.0/bin/ttInstanceCreate

NOTE: Each TimesTen instance is identified by a unique name.
      The instance name must be a non-null alphanumeric string, not longer
      than 255 characters.

Please choose an instance name for this installation? [ tt221 ] fullinstance1
Instance name will be 'fullinstance1'.
Is this correct? [ yes ]
Where would you like to install the fullinstance1 instance of TimesTen? [ /home/
ttuser ] /scratch/ttuser
Creating instance in /scratch/ttuser/fullinstance1 ...
INFO: Mapping files from the installation to /scratch/ttuser/fullinstance1/install

NOTE: If you are configuring TimesTen for use with Oracle Clusterware, the
      daemon port number must be the same across all TimesTen installations
      managed within the same Oracle Clusterware cluster.

NOTE: All installations that replicate to each other must use the same daemon
      port number that is set at installation time. The daemon port number can
      be verified by running 'ttVersion'.

The default port number is 6624.

Do you want to use the default port number for the TimesTen daemon? [ yes ]
The daemon will run on the default port number (6624).
```

```
In order to use the cache features in any TimesTen databases
created within this instance, you must set a value for the TNS_ADMIN
environment variable. It can be left blank, and a value can be supplied later
using <install_dir>/bin/ttInstanceModify.

Please enter a value for TNS_ADMIN (s=skip)? [  ] s
What is the TCP/IP port number that you want the TimesTen Server to listen on?
[ 6625 ]

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

NOTE: The TimesTen daemon startup/shutdown scripts have not been installed.

The startup script is located here :
        '/scratch/ttuser/fullinstance1/startup/tt_fullinstance1'

Run the 'setuproot' script :
        /scratch/ttuser/fullinstance1/bin/setuproot -install
This will move the TimesTen startup script into its appropriate location.

The 2211100 Release Notes are located here :
  '/installation_dir/tt22.1.1.10.0/README.html'

Starting the daemon ...
TimesTen Daemon (PID: 20396, port: 6624) startup OK.
```

Source the environment variables

You must set the environment variables to make use of TimesTen. In particular, ensure `TIMESTEN_HOME` is set. See "Setting environment variables for TimesTen" for information.

This example illustrates how to source the environment variables. Navigate to the `/bin` area of the instance directory (instance home) and source the `ttenv` script. This example uses `ttenv.csh`.

```
% source /scratch/ttuser/fullinstance1/bin/ttenv.csh
[...ttenv.csh output...]
```

Run the Linux/UNIX `printenv` command to verify the `TIMESTEN_HOME` variable is set.

```
% printenv TIMESTEN_HOME
/scratch/ttuser/fullinstance1
```

Configure Linux kernel parameters

After you create the full instance and source the environment variables, you must configure Linux kernel parameters. See "Configure Linux kernel parameters" for details.

Run the ttStatus utility

You can optionally run the `ttStatus` utility to verify that the TimesTen daemon is running. See "ttStatus" in the *Oracle TimesTen In-Memory Database Reference* for information on this utility.

This example illustrates how to use the `ttStatus` utility to verify that the daemon is running.

```
% ttStatus
TimesTen status report...
Daemon pid 20396 port 6624 instance database1
TimesTen server pid 20403 started on port 6625
------------------------------------------------------------------------
------------------------------------------------------------------------
Accessible by group g900
End of report
```

Create and connect to the database

This example illustrates how to create and connect to the database. It uses the Linux `vi` editor to modify the *$TIMESTEN_HOME*`/conf/sys.odbc.ini` file to add the `database1` DSN and specify connection attributes for the DSN. The example then runs the `ttIsql` utility to create and connect to the `database1` DSN. As a final step, the example runs a simple query.

For information on DSNs, see "Specifying Data Source Names to identify TimesTen databases" in the *Oracle TimesTen In-Memory Database Operations Guide*. For information on connection attributes, see "Connection Attributes" in the *Oracle TimesTen In-Memory Database Reference*.

```
% vi $TIMESTEN_HOME/conf/sys.odbc.ini

[database1]
DataStore=/disk1/databases/database1
LogDir=/disk2/logs
DatabaseCharacterSet=AL32UTF8
PermSize=1280
TempSize=640
LogBufMB=1024
```

After saving the `sys.odbc.ini` file and exiting from the `vi` editor, run the `ttIsql` utility to create the database and connect to the `database1` DSN. (A database is created when the instance administrator first connects to the newly created `database1` DSN).

```
% ttIsql database1

Copyright (c) 1996, 2023, Oracle and/or its affiliates. All rights reserved.
Type ? or "help" for help, type "exit" to quit ttIsql.



connect "DSN=database1";
Connection successful: DSN=database1;UID=ttuser;DataStore=/databases/database1;
DatabaseCharacterSet=US7ASCII;ConnectionCharacterSet=US7ASCII;
DRIVER=fullinstancedir/fullinstance1/install/lib/libtten.so;
(Default setting AutoCommit=1)
```

Run a simple query.

```
Command> SELECT * FROM dual;
< X >
1 row found.
```

Run ttStatus to show connections

You can run `ttStatus` to show connections to the `database1` database. See "ttStatus" in the *Oracle TimesTen In-Memory Database Reference* for information on this utility.

This example runs the `ttStatus` utility to show the connections to the database.

```
% ttStatus
TimesTen status report...

Daemon pid 20396 port 6624 instance database1
TimesTen server pid 20403 started on port 6625
------------------------------------------------------------------------
------------------------------------------------------------------------
Data store /databases/database1
Installation will expire after...
Daemon pid 20396 port 6624 instance database1
TimesTen server pid 20403 started on port 6625
There are 14 connections to the data store
Shared Memory KEY 0x02100497 ID 4292609
PL/SQL Memory KEY 0x03100497 ID 4325378 Address 0x5000000000
Type            PID     Context             Connection Name              ConnID
Process         20766   0x00007f06f4ccf010  database1                         1
Subdaemon       20401   0x00007f92b40008c0  Checkpoint                     2041
Subdaemon       20401   0x00007f92bc0008c0  Monitor                        2042
Subdaemon       20401   0x00007f92c40008c0  Deadlock Detector              2044
Subdaemon       20401   0x00007f92dc0008c0  Flusher                        2045
Subdaemon       20401   0x00007f92e485d010  Garbage Collector              2036
Subdaemon       20401   0x00007f92e4c78010  XactId Rollback                2037
Subdaemon       20401   0x00007f92e5194010  IndexGC                        2035
Subdaemon       20401   0x00007f92e56b0010  HistGC                         2040
Subdaemon       20401   0x00007f92e5bcc010  AsyncMV                        2038
Subdaemon       20401   0x00007f92e60e8010  Log Marker                     2039
Subdaemon       20401   0x00007f92e6503010  Aging                          2043
Subdaemon       20401   0x00007f92e77ca010  Rollback                       2046
Subdaemon       20401   0x00007f92e7be5010  Manager                        2047
Replication policy  : Manual
Cache Agent policy  : Manual
PL/SQL enabled.
------------------------------------------------------------------------
Accessible by group g900
End of report
```

Modify the full instance

This example runs the `ttInstanceModify` utility to modify the daemon port number. The instance that is modified is the one that `$TIMESTEN_HOME` references. Running this command:

1. Prompts you for confirmation
2. Stops the TimesTen daemon for the instance
3. Edits the `$TIMESTEN_HOME/conf/timesten.conf` file with the updated information
4. Starts the TimesTen daemon for the instance

See "Modifying an instance on Linux/UNIX" for details.

This example runs the Linux/UNIX `echo` command to display the instance that `$TIMESTEN_HOME` references and reviews the `$TIMESTEN_HOME/conf/timesten.conf` file to check the setting of the current port number. It then runs `ttInstanceModify` to change the port number to `6524`. The example then reviews the `$TIMESTEN_HOME/conf/timesten.conf` file to ensure the port number has been changed. As a final step, the daemon is started.

```
% echo $TIMESTEN_HOME
/scratch/ttuser/fullinstance1

% cat $TIMESTEN_HOME/conf/timesten.conf
# TimesTen Instance Configuration File
# Created by ttInstanceCreate
# Commented values are default values
hostname=host1
timesten_release=22.1
instance_name=fullinstance1
daemon_port=6624
server_port=6625
admin_user=myadminuser
admin_uid=4738
group_name=timesten
instance_guid=9EEF0277-21C0-45F1-AB63-F0C5F48B6FE0
verbose=1
```

Run `ttInstanceModify` to change the port number to `6524`.

```
% $TIMESTEN_HOME/bin/ttInstanceModify -port 6524
The daemon will now run on port 6524 ...
You must restart the daemon for these changes to take effect.
```

Review the `$TIMESTEN_HOME/conf/timesten.conf` file to ensure the port number is changed. See "Instance configuration file (timesten.conf)" for information on this file.

```
% cat $TIMESTEN_HOME/conf/timesten.conf
# TimesTen Instance Configuration File
# Created by ttInstanceCreate
# Commented values are default values
hostname=host1
timesten_release=22.1
instance_name=fullinstance1
daemon_port=6524
server_port=6625
admin_user=myadminuser
admin_uid=4738
group_name=timesten
instance_guid=9D37C711-DF86-4007-A959-2AB52DA46035
verbose=1
```

Restart the daemon

```
% ttDaemonAdmin -start -force
TimesTen Daemon (PID: 11635, port: 6524) startup OK.
```

Destroy the full instance

You use the `ttInstanceDestroy` utility interactively to destroy the instance. This example runs the utility without options. You must run this utility from the `bin` area of the installation directory. See "Destroying an instance on Linux/UNIX" for information.

```
% installation_dir/tt22.1.1.10.0/bin/ttInstanceDestroy

** WARNING **

   The uninstallation has been executed by a non-root user.
   If the TimesTen daemon startup scripts were installed,
   you must run $TIMESTEN_HOME/bin/setuproot -uninstall
   to remove them. If you proceed with this uninstallation, you
   will have to remove the startup scripts manually.

** WARNING **

   All files in the directory:

   /scratch/ttuser/fullinstance1

   will be removed, including any file that you or other users
   may have created.

   Are you sure you want to completely remove this instance? [ yes ] yes
   Installation will remove all the files from /scratch/ttuser/fullinstance1.
   Do you want to continue? [ yes ]

NOTE: /scratch/ttuser/fullinstance1/info contains information
related to the data store that have been created with this release.
If you remove /scratch/ttuser/fullinstance1/info
you will no longer be able to access your data stores,
nor would you be able to restore nor migrate your data.


NOTE: /scratch/ttuser/fullinstance1/conf contains information
related to the instance configuration.
/scratch/ttuser/fullinstance1 Removed
The TimesTen instance fullinstance1 has been destroyed.
```

Delete the installation

You can delete the installation. See "Deleting an installation on Linux/UNIX" for details.

Installing Oracle Clusterware for use with TimesTen

To install Oracle Clusterware, see "Oracle Clusterware" in the Oracle Database documentation. Also see "Using Oracle Clusterware to Manage Active Standby Pairs" in the *Oracle TimesTen In-Memory Database Replication Guide*.

Follow these recommendations:

- Create an NFS share 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.
- Do not install in a shared Oracle home, as this 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 /swdir/oracle
    mount storage-server:/export/compute_node_1/general /swdir/oracle
    ```

  - On host 2, do as follows.

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

  - On host *n*, do as follows.

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

- On each host: Install Oracle Clusterware in `/swdir/oracle/crs`.

3

Using systemd to Manage a TimesTen Service

Traditionally Linux systems used the System V init mechanism to coordinate system startup operations. More recently, Linux has moved to a new method based on the systemd daemon and its associated infrastructure. Although systemd is currently the standard for system startup coordination and for service management on Linux, the System V init mechanism is still supported.

It has always been possible to set a TimesTen instance's main daemon to start automatically on system startup. This is achieved by installing suitable System V init startup scripts. TimesTen provides the `setuproot` utility script for this purpose. On systems that support systemd, as an alternative to using the System V init mechanism, it is possible to configure a TimesTen instance's main daemon to run as a system service.

See "About creating an instance on Linux/UNIX" for the choices you can make if your Linux system supports both the System V init mechanism and systemd.

> **Note:**
>
> You can only use systemd in TimesTen Classic.

The chapter provides an overview of how systemd can be used for automatic management of the TimesTen daemon. It explains the importance of the systemd service unit configuration file as a mechanism for setting operating system limits for the TimesTen environment. The chapter details the process for creating and modifying an instance for automatic systemd management. It concludes with examples illustrating how to create and modify an instance for automatic systemd management.

See the *Oracle TimesTen In-Memory Database Release Notes* for information on the TimesTen Linux platforms that support systemd.

Topics:

About creating a TimesTen instance that uses automatic systemd management

systemd is a system and service manager for Linux operating systems. If you have a Linux system that supports both the System V init mechanism and systemd, you have choices as to how to manage the TimesTen daemon. This section focuses on systemd.

> **Note:**
>
> - If TimesTen is managed by Oracle Clusterware, you cannot use systemd for automatic management of the TimesTen daemon.
> - Using systemd for automatic management of the TimesTen daemon is supported in TimesTen Classic only. It is not supported in TimesTen Scaleout.

systemd provides a service unit type to start and control daemons and their associated processes. systemd uses a systemd service unit configuration file, whose name ends in `.service`, to encode information about the daemon process. systemd then uses this information to control and manage the process.

The TimesTen full distribution provides a systemd service unit configuration file customized for TimesTen. Its purpose is to provide the necessary information so that systemd can automatically control and manage the TimesTen daemon. TimesTen makes this file available in the *timesten_home*`/startup` directory when the instance administrator creates the TimesTen instance. This file is called the *TimesTen service* file. It has the naming convention of `tt_`*myinstance*`.service`, where *myinstance* is the name of the TimesTen instance. The systemd service that is created from this TimesTen service file is called the *TimesTen service*. This TimesTen service, when enabled and started, allows for automatic management of the TimesTen daemon by systemd.

The TimesTen service file is in a format specific to the requirements of the systemd service unit configuration file and the settings have been customized for TimesTen. systemd ignores the `memlock` entries in the `/etc/security/limits.conf` file. These `memlock` entries control the amount of memory a user can lock and are important settings when configuring the operating system requirements for TimesTen. Instead of using the `memlock` entries in the `/etc/security/limits.conf` file, systemd uses the `LimitMEMLOCK` setting in the `#Service Limit Settings` section of the service unit configuration file (TimesTen service file). This setting and two other settings are important for your TimesTen environment. Modify them as appropriate.

These are the settings and their values:

- `LimitNOFILE: 65536`
- `LimitMEMLOCK: infinity`
- `TasksMax: 65536`

For example, here is the section in the TimesTen service file:

```
% cat timesten_home/start/tt_myinstance.service
#
# Oracle TimesTen In-Memory Database
# Copyright (c) 2020, 2021, Oracle and/or its affiliates.
...
# Service Limit Settings
LimitNOFILE = 65536
LimitMEMLOCK = infinity
TasksMax=65536
...
```

> **Note:**
>
> Do not make any other modifications to this file. Doing so alters the systemd configuration for TimesTen.

The process for creating a TimesTen instance that uses automatic systemd management involves:

- Configuring the TimesTen instance for systemd: The instance administrator uses the `ttInstanceCreate` utility with the `-systemd` option to create the TimesTen instance. Alternatively, the instance administrator can run the `ttInstanceCreate` utility interactively. When prompted with the `Would you like to use systemd to manage TimesTen` question, the instance adminstrator answers `Yes` (or takes the default).
- Configuring the Linux kernel parameters: You must configure Linux kernel parameters after creating the TimesTen full instance. These include `shmmax` and `shmmall` shared memory kernel parameters, HugePages, semaphores, and the `SHMMNI` parameter.
- Modifying the TimesTen service file:

  If necessary, the instance administrator modifies the `LimitNOFILE`, the `LimitMEMLOCK`, and the `TasksMax` settings in the `# Service Limit Settings` section of the `timesten_home/startup/tt_myinstance.service` Times Ten service file (where `myinstance` is the name of the TimesTen instance). This ensures the operating system limits requirements are met for the TimesTen environment.
- Installing the TimesTen utility script:

  A TimesTen instance that is configured with systemd is not enabled automatically. The `root` user uses the *`timesten_home`*`/bin/setuproot` utility script with the `-install -systemd` options to enable automatic management of the TimesTen instance by systemd.

Reference information:

- https://www.freedesktop.org/software/systemd/man/systemd.service.html for information on systemd.service - Service unit configuration.
- https://www.freedesktop.org/software/systemd/man/systemd.exec.html for information on systemd.exec - Execution environment configuration. This is important for the `LimitNOFILE` and the `LimitMEMLOCK` settings.

- https://www.freedesktop.org/software/systemd/man/systemd.resource-control.html for information on systemd.resource-control - Resource control unit settings. This is important for the `TasksMax` setting.
- "ttInstanceCreate" in the *Oracle TimesTen In-Memory Database Reference* for information on the TimesTen `ttInstanceCreate` utility.

Creating a TimesTen instance that uses automatic systemd management

Perform the procedures in these sections to create an instance that uses automatic systemd management:

- Configure the TimesTen instance for systemd
- Configure Linux kernel parameters for systemd
- Complete remaining steps for automatic systemd management

Configure the TimesTen instance for systemd

Follow these steps to create the TimesTen instance with systemd:

1. Run the *installation_dir*`/tt22.1.1.10.0/bin/ttInstanceCreate` utility to create the TimesTen instance. This example runs `ttInstanceCreate` interactively. Ensure that you do not enter `yes` for Oracle Clusterware. If TimesTen is managed by Oracle Clusterware, systemd cannot be used.

   Note the following:

   - Enter `yes` to the question: `Would you like to use systemd to manage TimesTen?`
   - Observe the name of the `/scratch/ttuser/myinstance/startup/` TimesTen service file (`tt_myinstance.service`, in this example). Later, you will run the `/scratch/ttuser/myinstance/bin/setuproot -install -systemd` script so that TimesTen can copy this file to the appropriate systemd location (`/lib/systemd/system`). This enables systemd to automatically manage the TimesTen daemon.

```
% installation_dir/tt22.1.1.10.0/bin/ttInstanceCreate

NOTE: Each TimesTen instance is identified by a unique name.
      The instance name must be a non-null alphanumeric string, not
longer
      than 255 characters.

Please choose an instance name for this installation? [ tt221 ]
myinstance
Instance name will be 'myinstance'.
Is this correct? [ yes ]
Where would you like to install the myinstance instance of
TimesTen? [ /home/ttuser ] /scratch/ttuser
Creating instance in /scratch/ttuser ...

NOTE: If you are configuring TimesTen for use with Oracle
```

```
Clusterware, the
      daemon port number must be the same across all TimesTen
installations
      managed within the same Oracle Clusterware cluster.

NOTE: All installations that replicate to each other must use the same
daemon
      port number that is set at installation time. The daemon port
number can
      be verified by running 'ttVersion'.

The default port number is 6624.

Do you want to use the default port number for the TimesTen daemon?
[ yes ]
The daemon will run on the default port number (6624).

In order to use the cache features in any TimesTen databases
created within this instance, you must set a value for the TNS_ADMIN
environment variable. It can be left blank, and a value can be supplied
later
using <install_dir>/bin/ttInstanceModify.

Please enter a value for TNS_ADMIN (s=skip)? [  ] s
What is the TCP/IP port number that you want the TimesTen Server to
listen on? [ 6625 ]

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

Would you like to use systemd to manage TimesTen? [ no ] yes

NOTE: The TimesTen daemon startup/shutdown scripts have not been
installed.

The startup script is located here :
        '/scratch/ttuser/myinstance/startup/tt_myinstance.service'

Run the 'setuproot' script :
        /scratch/ttuser/myinstance/bin/setuproot -install -systemd
This will move the TimesTen startup script into its appropriate location.

The 22.1 Release Notes are located here :
  '/installation_dir/tt22.1.1.1.0/README.html'

Instance created successfully.
```

2. Run the `ttVersion` utility to verify the TimesTen instance is being managed by systemd.

```
$TIMESTEN_HOME/bin/ttVersion
TimesTen Release 22.1.1.10.0 (64 bit Linux/x86_64) (myinstance:6624)
2021-07-18T07:37:31Z
  Instance admin: instanceadmin
  Instance home directory: /scratch/ttuser/myinstance
  Group owner: g900
  Daemon home directory: /scratch/ttuser/myinstance/info
```

```
PL/SQL enabled.
Daemon is managed by systemd.
```

You have successfully configured the TimesTen instance for systemd.

Configure Linux kernel parameters for systemd

After creating your TimesTen instance, configure these Linux kernel parameters:

- Configure shmmax and shmall
- Configure HugePages
- Set the semaphore values
- Set the SHMMNI parameter

Configure shmmax and shmall

You must configure Linux shared memory so that the maximum size of a shared memory segment (the `shmmax` memory kernel parameter) is large enough to contain the size of the total shared memory segment for the database. In TimesTen Classic, the entire database resides in a single shared memory segment. There is also a second memory segment used for PL/SQL.

On Linux, a shared memory segment consists of pages, where the default page size is normally 4 kB (4096 bytes). You can verify the default page size by running the `getconf PAGESIZE` command:

```
% getconf PAGESIZE
4096
```

Configure these shared memory kernel parameters to control the size of the shared memory segment:

- `shmmax`: The maximum size of a single shared memory segment expressed in bytes. The value must be large enough to accommodate the size of the total shared memory segment for the database.
- `shmall`: The total size of shared memory segments system wide expressed in pages. The value is expressed in multiples of the page size (4 kB) and `shmall * pagesize` must be greater or equal to the value of `shmmax`. It is recommended that you set the value of `shmall` to less than or equal to the total amount of physical RAM. To display the total amount of physical memory, run the Linux `cat /proc/meminfo` command.

Use the `ttShmSize` utility to determine the size of the shared memory segment. The `ttShmSize` utility uses the values of the `PermSize`, the `TempSize`, the `LogBufMB` and the `Connections` connection attributes (for a specified database) to determine this size. See "ttShmSize" in *Oracle TimesTen In-Memory Database Reference* for details on the `ttShmSize` utility and see "PermSize", "TempSize", "LogBufMB", and "Connections" in *Oracle TimesTen In-Memory Database Reference* for details on each connection attribute.

For example, use the `ttShmSize` utility with the `-connStr` option to determine the size of the shared memory segment using the `database1` DSN. Supply a `PermSize` value of

32GB (32768 MB), a `TempSize` value of 4 GB (4096 MB), a `LogBufMB` value of 1 GB (1024 MB) and a `Connections` value of 2048.

```
% ttShmSize -connstr
"DSN=database1;PermSize=32768;TempSize=4096;LogBufMB=1024;Connections=2048"
The required shared memory size is 39991547720 bytes.
```

Note:

- The `-connStr` option of the `ttShmSize` utility requires that you have defined a DSN in either the user `.odbc.ini` or the system `sys.odbc.ini` file. You may use any DSN. Note that for any connection attribute not specified in the `-connStr` option, `ttShmSize` uses the setting defined in either the user `.odbc.ini` file or the system `sys.odbc.ini` file for the specified DSN. If the connection attribute is missing from both the `-connStr` option and either the user `.odbc.ini` file or the `sys.odbc.ini` file, `ttShmSize` uses the default value for the connection attribute.
- You can add a DSN to either the user `.odbc.ini` file of the user or the system `sys.odbc.ini` file. For example, to add the `database1` DSN to the user `.odbc.ini` file of the current operating system user:

  ```
  % vi ~/.odbc.ini
  ...
  [database1]
  ```

To size `shmmax` and `shmall`:

1. As the `root` user, edit the `/etc/sysctl.conf` file, modifying `kernel.shmmax` and `kernel.shmall`. Set `shmmax` to 39,991,547,720 bytes and `shmall` to 9,763,561 pages, which is `shmmax/pagesize`.

   ```
   % sudo vi /etc/sysctl.conf
   ...
   kernel.shmmax=39991547720
   kernel.shmall=9763561
   ```

2. To reload the settings from the modified `/etc/sysctl.conf` file:

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

3. Run the Linux `ipcs lm` command to display the current `shmmax` and `shmall` settings. The `max seg size (kbytes)` is the `shmmax` value and the `max total shared memory (kbytes)` is the value of `shmall` times `max number of segments`.

   ```
   % ipcs -lm

   ------ Shared Memory Limits --------
   max number of segments = 4096
   max seg size (kbytes) = 39054246
   max total shared memory (kbytes) = 39054246
   min seg size (bytes) = 1
   ```

> **Note:**
>
> - The settings for `shmmax` and `shmall` in these examples can be increased if there are other applications that require them to be greater.
> - If you are unsure of the size of your database, you can set `shmmax` and `shmall` to correspond to a percentage of the size of physical memory (such as 80%).

Configure HugePages

You can configure `HugePages` for more efficient memory management.

Once configured, the memory allocated for `HugePages` is taken from the total RAM on the Linux host and is not available for any other use. In addition, the `HugePages` memory segment is automatically locked and cannot be swapped to disk.

To configure `HugePages`, you need to know:

- The maximum size of the shared memory segment for the database
- The `HugePages` page size on your Linux host
- The group ID of the instance administrator

Using the examples in the "Configure shmmax and shmall" section, where the value of `shmmax` value is 39,054,246 kB, and the "Create the TimesTen Users Group " section, where the group ID of the `instanceadmin` user is 10000:

- The size of the total shared memory segment is is 39,054,246 kB.
- The `HugePages` page size is 2048 KB. (This value is fixed for each platform and is not configurable.)

  To determine the HugePages page size, run the Linux `cat /proc/meminfo|grep Hugepagesize` command:

  ```
  % cat /proc/meminfo | grep Hugepagesize
  Hugepagesize:       2048 kB
  ```

- The group ID is 10000.

  To determine the group ID of the instance administrator, log in as the `instanceadmin` user, and run the Linux `id` command:

  ```
  % id
  uid=55000(instanceadmin) gid=10000(g10000)groups=10000(g10000)
  ```

To configure `HugePages`:

1. Determine the number of `HugePages` by dividing the size of the total shared memory segment (expressed in MB) by the value of `Hugepagesize` (expressed in MB). In this example, the total shared memory segment is 39,054,246 kB (~38,138 MB) and the `Hugepagesize` value is 2,048 kB (2 MB):

   ```
   38138 MB/ 2 MB = 19069
   ```

2. As the root user, edit the `/etc/sysctl.conf` file, and set `vm.nr_hugepages` to the number of `HugePages` and set `vm.hugetlb_shm_group` to the group ID of the

instance administrator. The latter setting restricts access to `HugePages` to members of the group.

```
% sudo vi /etc/sysctl.conf
...
vm.nr_hugepages=19069
vm.hugetlb_shm_group=10000
```

3. Reload the settings from the modified `/etc/sysctl.conf` file:

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

4. To verify that you have configured `HugePages` correctly, run:

```
% cat /proc/meminfo | grep HugePages
HugePages_Total:   19069
HugePages_Free:    19069
...
```

Note:

- Because `HugePages` must be allocated in contiguous available memory space, the requested allocation may not be granted, or may be only partially granted, until after the host is restarted. Check the `HugePages_Total` and `HugePages_Free` values from `/proc/meminfo`. Restarting grants the full allocation, assuming enough memory is available in the host.
- The TimesTen PL/SQL shared memory segment consumes some of the configured HugePages allocation, determined by the value of the `PLSQL_MEMORY_SIZE` connection attribute. See "PLSQL_MEMORY_SIZE" in the *Oracle TimesTen In-Memory Database Reference* for more information.
- On Linux, the `HugePages` segment is automatically locked such that the memory segment is not a candidate to be swapped to disk. Therefore, if you configure `HugePages`, you do not need to set the `MemoryLock` connection attribute.

Set the semaphore values

TimesTen has an upper bound on the maximum number of connections to the database. The database connections consist of:

- User connections: established by user applications
- System connections: established internally by TimesTen (set at 48 connections)
- Other required connections (set at 107 connections)

Each of these connections is assigned one semaphore, such that the total semaphores for a database are:

```
Total semaphores = user connections (N) + system connections (48) +
                   other required connections (107)

Total semaphores = N + 155
```

The semaphore settings are located in the `kernel.sem` configuration directive in `/etc/sysctl.conf`:

```
kernel.sem = SEMMSL SEMMNS SEMOPM SEMMNI
```

where:

- SEMMSL is the maximum number of semaphores per array. This value is related to the maximum number of connections. Configure this value to be 155 plus the number of simultaneous user connections.
- SEMMNS is the maximum number of semaphores system wide. Use the formula SEMMNS = (SEMMNI * SEMMSL) as a guideline. However, in practice, SEMMNS can be much less than SEMMNI * SEMMSL.
- SEMOPM is the maximum number of operations for each semop call.
- SEMMNI is the maximum number of arrays.

Follow these steps to configure the SEMMSL and the SEMMNI settings. Ensure that the user is root:

1. View the existing kernel parameter settings:

   ```
   # /sbin/sysctl -a | grep kernel.sem
   kernel.sem = 2500 320000 1000 1280
   ```

2. Edit the /etc/sysctl.conf file, changing SEMMSL (the first of the four values in kernel.sem) to 155 plus the number of simultaneous user connections.

   In this example, assume the number of simultaneous user connections is 4000. Set the SEMMSL value to 4155 (=4000+155) or greater.

   In addition, change SEMMNI (the last of the four values in kernel.sem) to the value of SEMMSL plus the number of TimesTen-specific shared memory segments. (These shared memory segments include the TimesTen database and PL/SQL.) In this example, set SEMMNI to 4157 (=4155+2) or greater.

   > **Note:**
   >
   > TimesTen uses the value of the Connections first connection attribute to determine the maximum number of simultaneous user connections that can be connected to the TimesTen database. TimesTen returns an error if the number of simultaneous user connections exceeds this value. The default value is the lesser of 2000 or (SEMMSL – 155). See "Defining Server DSNs for TimesTen Server on a Linux or UNIX system" in the *Oracle TimesTen In-Memory Database Operations Guide* for information on setting the Connections attribute. Also see "Connections" in the *Oracle TimesTen In-Memory Database Reference* for information on the Connections first connection attribute.

   ```
   # vi /etc/sysctl.conf
   ...
   ...
   kernel.sem = 4155 400000 2000 4157
   ```

3. Reload the settings from the modified /etc/sysctl.conf file.

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

> **Note:**
>
> If you are using replication, the Linux platform for each host on which the master databases reside must have similar kernel settings for shared memory and semaphores. Specifically, the `SEMMSL` and `SEMMNI` settings must be large enough on all hosts that participate in an active standby replication scheme before the duplication is performed. In the event of a failover, the standby must be able to accommodate the active.

Set the SHMMNI parameter

The `SHMMNI` value controls the number of shared memory segments that the host can create simultaneously. TimesTen creates a shared memory segment for the TimesTen database and a shared memory segment for PL/SQL. In addition, there is a small shared memory segment that is allocated for the duration of each client/server connection. This shared memory segment is created at connect time and is destroyed when the client/server connection is disconnected from the TimesTen database. There is one shared memory segment per client/server connection.

You must configure the `SHMMNI` parameter setting to account for the number of client/server connections. Set `SHMMNI` to a value that is greater than number of expected client/server connections. (Ensure to also take into account the TimesTen shared memory segment, the PL/SQL shared memory segment, and other programs that use shared memory.) As an example, if you expect there to be `8000` client/server connections, an appropriate value is `9000` or greater. A value of `9000` or greater is appropriate as TimesTen has system connections that are not included in the client/server connections count.

Follow these steps to configure the `SHMMNI` setting. Ensure that the user is `root`:

1. View the existing `SHMMNI` parameter setting.

```
# /sbin/sysctl -a | grep shmmni
kernel.shmmni = 4096
```

2. Edit the `/etc/sysctl.conf` file, changing `kernel.shmmni` to a value that is greater than the number of client/server connections. This example sets `kernel.shmmni` to `9000`.

```
 vi /etc/sysctl.conf
...
...
kernel.shmmni = 9000
```

3. Reload the settings from the modified `/etc/sysctl.conf` file.

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

> **Note:**
>
> If you are using replication, the Linux platform for each host on which the master databases reside must have a similar `SHMMNI` kernel setting. Specifically, the `SHMMNI` setting must be large enough on all hosts that participate in an active standby replication scheme before the duplication is performed. In the event of a failover, the standby must be able to accommodate the active.

Complete remaining steps for automatic systemd management

Complete the following steps to first review and, if necessary, modify the TimesTen service file. Then, install the TimesTen utility script by running the *timesten_home*`/bin/setuproot` script. As a final step, run the systemd `systemctl start` command to start the TimesTen service.

1. Review the TimesTen service file located in the *timesten_home*`/startup` directory (`tt_`*myinstance*`.service`, in this example). If necessary, modify the `LimitNOFILE`, the `LimitMEMLOCK`, and the `TasksMax` settings in the `#Service Limit Settings` section of this file. No other section should be modified. This example makes no modifications to the file for the `myinstance` TimesTen instance. In this example, *timesten_home* is `/scratch/ttuser/myinstance`.

   > **Note:**
   >
   > Do not make any other modifications to this file.

   ```
   % cat /scratch/ttuser/startup/tt_myinstance.service
   #
   # Oracle TimesTen In-Memory Database
   # Copyright (c) 2020, 2021, Oracle and/or its affiliates.
   #
   ...
   # Service Limit Settings
   LimitNOFILE = 65536
   LimitMEMLOCK = infinity
   TasksMax=65536
   ```

   You have successfully reviewed and, if necessary, edited the TimesTen service file.

2. As the `root` user, run the *timesten_home*`/bin/setuproot` script with the `-install -systemd` options. The `setuproot` utility operates according to the setting of the *TIMESTEN_HOME* environment variable, which indicates the instance home directory (*timesten_home*).In this example *timesten_home* is `/scratch/ttuser/myinstance`.

Ensure to set the TimesTen environment variables by sourcing the *timesten_home*/bin/ttenv script. This ensures *TIMESTEN_HOME* is set. Then, as root, run the setuproot utility script.

```
$ source /scratch/ttuser/myinstance/bin/ttenv.csh
[...ttenv.csh output...]


# $TIMESTEN_HOME/bin/setuproot -install -systemd
Would you like to install the TimesTen daemon startup scripts into /lib/
systemd/system? [ yes ]
Copying /ttuser/myinstance/startup/tt_myinstance.service to /lib/systemd/
system/tt_myinstance.service

Successfully installed the following scripts :
/lib/systemd/system/tt_myinstance.service
Created symlink from /etc/systemd/system/multi-user.target.wants/
tt_myinstance.service to /lib/systemd/system/tt_myinstance.service.
Created symlink from /etc/systemd/system/tt_myinstance.service to /lib/
systemd/system/tt_myinstance.service.

Use 'systemctl [start|stop] tt_myinstance.service' to manage the service
```

3. As the root user, run the systemd systemctl start command to start the TimesTen service (tt_myinstance.service, in this example). You must use the systemd systemctl commands after the TimesTen service is started. You cannot use the TimesTen ttDaemonAdmin utility. See "https://www.freedesktop.org/software/systemd/man/systemctl.html" for information on the systemd systemctl commands.

```
 # systemctl start tt_myinstance.service
```

4. Run the systemd systemctl status command to verify that systemd is automatically managing the TimesTen daemon. In this example, *installation_dir* is the location of the installation directory for this instance.

```
% systemctl status tt_myinstance.service
● tt_myinstance.service - TimesTen Service
   Loaded: loaded (/lib/systemd/system/tt_myinstance.service; enabled;
vendor preset: disabled)
   Active: active (running) since Wed 2021-08-18 04:56:28 PDT; 7s ago
  Process: 22636 ExecStart=/scratch/ttuser/myinstance/bin/ttDaemonAdmin -
start -force -systemd (code=exited, status=0/SUCCESS)
 Main PID: 22644 (timestend)
   Memory: 53.2M
   CGroup: /system.slice/tt_myinstance.service
           ├─22644 /scratch/ttuser/myinstance/install/bin/timestend -
initfd...
           ├─22648 installation_dir/tt22.1.1.10.0/bin/timestensubd -ve...
           ├─22649 installation_dir/tt22.1.1.10.0/bin/timestensubd -ve...
           └─22651 installation_dir/tt22.1.1.10.0/bin/ttcserver -verbo...

Aug 18 04:56:27 myhost systemd[1]: Starting TimesTen Service...
Aug 18 04:56:27 myhost ttDaemonAdmin[22636]: TimesTen Daemon (PID:
22644, ....
```

```
Aug 18 04:56:28 myhost systemd[1]: Started TimesTen Service.
Hint: Some lines were ellipsized, use -l to show in full.
```

5. Run the TimesTen `ttStatus` utility to verify the TimesTen daemon for the instance is under the control and management of systemd.

```
$TIMESTEN_HOME/bin/ttStatus
TimesTen status report as of Wed Aug 18 12:13:45 2021

Daemon pid 22644 port 6624 instance myinstance
TimesTen server pid 22651 started on port 6625
------------------------------------------------------------------------
----
------------------------------------------------------------------------
----
Accessible by group g900
TimesTen daemon is managed by systemd
End of report
```

You have successfully completed the steps necessary to create and configure the TimesTen instance. The TimesTen daemon for the instance is under automatic systemd control and management.

About modifying a TimesTen instance to enable or disable management by systemd

The instance administrator uses the `ttInstanceModify` utility to modify an existing TimesTen instance. The instance administrator specifies the `-systemd` option to configure systemd for the existing instance or the `-nosystemd` option to remove the systemd configuration from the existing instance.

If `ttInstanceModify -systemd` is used: The `root` user runs the *`timesten_home`*`/bin/setuproot` script with the `-install -systemd` options. This enables systemd to automatically manage the TimesTen daemon.

If `ttInstanceModify-nosystemd` is used: The `root` user runs the *`timesten_home`*`/bin/setuproot` script with the `-uninstall -systemd` options. This results in systemd no longer automatically managing the TimesTen daemon.

For more information, see:

- "Modifying a TimesTen instance to enable management by systemd" for an example illustrating how to modify an instance using the `ttInstanceModify -systemd` option.
- "Modifying a TimesTen instance to disable management by systemd" for an example illustrating how to modify an instance using the `ttInstanceModify -nosystemd` option.

Modifying a TimesTen instance to enable management by systemd

This example assumes you have created the `myinstance_2` TimesTen instance without systemd configured. It walks you through the process for modifying the `myinstance_2` instance such that it is configured for systemd. It then performs the necessary steps for systemd to automatically manage the TimesTen daemon.

1. Run the *`$TIMESTEN_HOME`*`/bin/ttInstanceModify` utility with the `-systemd` option. TimesTen stops the TimesTen daemon and modifies the instance to configure it for systemd.

```
% $TIMESTEN_HOME/bin/ttInstanceModify -systemd
Configuring TimesTen daemon with systemd.
Stopping TimesTen daemon.....
TimesTen Daemon (PID: 29038, port: 6640) stopped.
The startup script is located here :
        '/scratch/ttuser2/myinstance_2/startup/myinstance_2.service'

Run the 'setuproot' script :
        /scratch/ttuser2/myinstance_2/bin/setuproot -install -systemd
This will move the TimesTen startup script into its appropriate location.

Please manage the TimesTen daemon with systemd
```

2. As the `root` user, run the *`$TIMESTEN_HOME`*`/bin/setuproot -install -systemd` script.

```
# $TIMESTEN_HOME/bin/setuproot -install -systemd
Would you like to install the TimesTen daemon startup scripts into /lib/
systemd/system? [ yes ]
Copying /scratch/ttuser2/myinstance_2/startup/tt_myinstance_2.service
to /lib/systemd/system/tt_myinstance_2.service

Successfully installed the following scripts :
/lib/systemd/system/tt_myinstance_2.service
Created symlink from /etc/systemd/system/multi-user.target.wants/
tt_myinstance_2.service to /lib/systemd/system/tt_myinstance_2.service.
Created symlink from /etc/systemd/system/tt_myinstance_2.service to /lib/
systemd/system/tt_myinstance_2.service.

Use 'systemctl [start|stop] tt_myinstance_2.service' to manage the service
```

3. As the `root` user, run the systemd `systemctl start` command to start the TimesTen service (`tt_myinstance_2.service`, in this example).

```
 # systemctl start tt_myinstance_2.service
```

4. Run the systemd `systemctl status` command to verify the TimesTen daemon for the instance is automatically managed by systemd.

```
% systemctl status tt_myinstance_2.service
● tt_myinstance_2.service - TimesTen Service
   Loaded: loaded (/lib/systemd/system/tt_myinstance_2.service;
enabled; vendor preset: disabled)
   Active: active (running) since Wed 2021-08-18 05:55:00 PDT; 39s
ago
  Process: 30976 ExecStart=/scratch/ttuser2/myinstance_2/bin/
ttDaemonAdmin -start -force -systemd (code=exited, status=0/SUCCESS)
 Main PID: 30984 (timestend)
   Memory: 53.3M
   CGroup: /system.slice/tt_myinstance_2.service
           ├─30984 /scratch/ttuser2/myinstance_2/install/bin/
timestend -init...
           ├─30988 installation_dir/tt22.1.1.10.0/bin/timestensubd -
ve...
           ├─30989 installation_dir/tt22.1.1.10.0/bin/timestensubd -
ve...
           └─30991 installation_dir/tt22.1.1.10.0/bin/ttcserver -
verbo...
```

5. Run the TimesTen `ttStatus` utility to verify the TimesTen daemon for the instance is automatically managed by systemd.

```
$TIMESTEN_HOME/bin/ttStatus
TimesTen status report as of Wed Aug 18 12:59:15 2021

Daemon pid 30984 port 6640 instance myinstance_2
TimesTen server pid 30991 started on port 6641
------------------------------------------------------------------------
----
------------------------------------------------------------------------
----
Accessible by group g900
TimesTen daemon is managed by systemd
End of report
```

You have successfully modified the TimesTen instance to use systemd. The TimesTen daemon is automatically managed by systemd.

Modifying a TimesTen instance to disable management by systemd

This example assumes the `myinstance_2` TimesTen instance is configured with systemd. The `tt_myinstance_2` TimesTen service is running and is under automatic management of systemd. The example walks through the process of modifying the instance to remove the systemd configuration. It then performs the necessary steps to stop the `tt_myinstance_2` TimesTen service and revert to manual control of the TimesTen daemon.

1. Use the systemd `systemctl status` command to verify the `tt_myinstance_2` TimesTen service is running.

```
% systemctl status tt_myinstance_2.service
● tt_myinstance_2.service - TimesTen Service
   Loaded: loaded (/lib/systemd/system/tt_myinstance_2.service; enabled;
vendor preset: disabled)
   Active: active (running) since Wed 2021-08-18 05:55:00 PDT; 39s ago
  Process: 30976 ExecStart=/scratch/ttuser2/myinstance_2/bin/
ttDaemonAdmin -start -force -systemd (code=exited, status=0/SUCCESS)
 Main PID: 30984 (timestend)
   Memory: 53.3M
   CGroup: /system.slice/tt_myinstance_2.service
           ├─30984 /scratch/ttuser2/myinstance_2/install/bin/timestend -
init...
           ├─30988 installation_dir/tt22.1.1.10.0/bin/timestensubd -ve...
           ├─30989 installation_dir/tt22.1.1.10.0/bin/timestensubd -ve...
           └─30991 installation_dir/tt22.1.1.10.0/bin/ttcserver -verbo...
```

2. Run the *$TIMESTEN_HOME*`/bin/ttInstanceModify` utility with the `-nosystemd` option. TimesTen modifies the instance to remove the systemd configuration.

```
% $TIMESTEN_HOME/bin/ttInstanceModify -nosystemd
The daemon for instance 'myinstance_2' is currently configured to be
managed by systemd.

Unconfiguring TimesTen daemon startup from systemd

** NOTICE **

If the TimesTen daemon startup scripts were installed, you must run
        $TIMESTEN_HOME/bin/setuproot -uninstall -systemd
to remove them.
```

3. Run the TimesTen `ttStatus` utility to illustrate the `myinstance_2` instance is no longer configured for systemd.

```
% $TIMESTEN_HOME/bin/ttStatus
TimesTen status report as of Wed Aug 18 13:25:31 2021

Daemon pid 30984 port 6640 instance myinstance_2
TimesTen server pid 30991 started on port 6641
------------------------------------------------------------------------
------------------------------------------------------------------------
Accessible by group g900
End of report
```

4. Run the systemd `systemctl status` command to illustrate that the `tt_myinstance_2` TimesTen service is still running.

```
% systemctl status tt_myinstance_2.service
● tt_myinstance_2.service - TimesTen Service
   Loaded: loaded (/lib/systemd/system/tt_myinstance_2.service; enabled;
vendor preset: disabled)
```

```
   Active: active (running) since Wed 2021-08-18 05:55:00 PDT;
29min ago
  Process: 30976 ExecStart=/scratch/ttuser2/myinstance_2/bin/
ttDaemonAdmin -start -force -systemd (code=exited, status=0/SUCCESS)
 Main PID: 30984 (timestend)
   Memory: 53.8M
   CGroup: /system.slice/tt_myinstance_2.service
           ├─30984 /scratch/ttuser2/myinstance_2/install/bin/
timestend -init...
           ├─30988 installation_dir/tt22.1.1.1.0/bin/timestensubd -
ve...
           ├─30989 installation_dir/tt22.1.1.1.0/bin/timestensubd -
ve...
           └─30991 installation_dir/tt22.1.1.1.0/bin/ttcserver -
verbo...
```

5. As the `root` user, run the *`$TIMESTEN_HOME`*`/bin/setuproot -uninstall -systemd` script to disable and stop the `tt_myinstance_2` TimesTen service.

```
# $TIMESTEN_HOME/bin/setuproot -uninstall -systemd
Would you like to uninstall the TimesTen daemon startup scripts
in /lib/systemd/system? [ yes ]
Removed symlink /etc/systemd/system/tt_myinstance_2.service.
Removed symlink /etc/systemd/system/multi-user.target.wants/
tt_myinstance_2.service.

Successfully deleted the following scripts :
/lib/systemd/system/tt_myinstance_2.service
```

6. Run the systemd `systemctl status` command to verify systemd is no longer managing the TimesTen daemon.

```
% systemctl status tt_myinstance_2.service
● tt_myinstance_2.service - TimesTen Service
   Loaded: loaded (/usr/lib/systemd/system/tt_myinstance_2.service;
bad; vendor preset: disabled)
   Active: failed (Result: exit-code) since Wed 2021-08-18 06:39:57
PDT; 2min 13s ago
 Main PID: 30984 (code=exited, status=0/SUCCESS)
Warning: tt_myinstance_2.service changed on disk. Run 'systemctl
daemon-reload' to reload units.
```

7. Run the TimesTen `ttStatus` utility. The TimesTen daemon is no longer running and is now under manual control of the instance administrator. As the instance administrator, use the TimesTen `ttDaemonAdmin` to manually start the TimesTen daemon. Then use the `ttStatus` utility to ensure the TimesTen daemon is running.

```
% ttStatus
ttStatus: Could not connect to the TimesTen daemon on port 6640.
If the TimesTen daemon is not running, please start it
by running "ttDaemonAdmin -start".
```

```
/scratch/ttuser2/myinstance_2/info/daemon.status says:
2021-08-18 12:55:00 TimesTen daemon starting up


% ttDaemonAdmin -restart
ttDaemonAdmin: daemon is not running
TimesTen Daemon (PID: 7060, port: 6640) startup OK.


% ttStatus
TimesTen status report as of Wed Aug 18 06:53:52 2021

Daemon pid 7060 port 6640 instance myinstance_2
TimesTen server pid 7067 started on port 6641
------------------------------------------------------------------------
------------------------------------------------------------------------
Accessible by group g900
End of report
```

You have successfully modified the instance to remove the `systemd` configuration. In addition, you have performed the steps to stop the automatic management of the TimesTen daemon by systemd. The instance administrator is now manually managing the TimesTen daemon.

4

Installation and Management of TimesTen on Windows

This chapter discusses how to install a TimesTen client on Windows. Review "Overview of the Installation Process in TimesTen Classic" before beginning the installation.

Topics include:

- Overview of the installation process on Windows
- Creating an installation on Windows
- Deleting an installation on Windows

Overview of the installation process on Windows

The TimesTen installer on Windows creates a single TimesTen client-only installation (and instance). No additional installations (or instances) can be created without first uninstalling the existing one.

Once installed, a Windows host can be used as a TimesTen client that connects to either a database in TimesTen Scaleout or a database in TimesTen Classic.

A Windows host cannot have multiple installations from the same TimesTen major release, such as `22.1`. For example, it cannot have both a `22.1.1` and a `22.1.2.x` installation. But it can have TimesTen installations from different major releases, such as an `18.1.4.17.x` installation and a `22.1.x`. installation.

You cannot modify the installation or the instance.

> **Note:**
>
> TimesTen does not support copying a Windows installation or sharing a Windows installation between hosts. TimesTen must be installed separately on each host. In addition, TimesTen does not support Oracle Names Server for Windows clients.

Creating an installation on Windows

You can install TimesTen in any directory where the user performing the installation has sufficient permission. The user who installs TimesTen must be a member of the TimesTen users group, and the TimesTen files and the directories must be accessible to only members of that group. Installing TimesTen requires Administrator privileges.

Information about the TimesTen installation is contained in the Windows operating system registry.

Do not install TimesTen for Windows on a mapped network drive.

The default top-level directory of the installation is `C:\TimesTen\tt221_64`. The instance name is simply `instance`.

These sections describe how to create the installation and then verify it.

- Create an installation on Windows
- Verify the installation on Windows

After you create and verify the installation, you can connect to the server from the Windows client and set up your DSN. See "Working with the TimesTen Client and Server" in the *Oracle TimesTen In-Memory Database Operations Guide* for details.

Create an installation on Windows

The procedure for the installation is:

1. Download the appropriate TimesTen client distribution ZIP file. For example, download `timesten.2211100.win64.zip`.
2. Extract the contents of the file.
3. From the `Win64` directory, run the `setup.exe` executable (the TimesTen installer).
4. In the initial TimesTen Setup dialog, choose **Next**.
5. In the Choose Installation Folder dialog, select or specify the desired folder. The default is `C:\TimesTen\tt221_64`. Then choose **OK**.
6. In the Select Program Folder dialog, select or specify the desired folder. The default is `TimesTen 22.1 (64-bit)`. Then choose **Next**.
7. In the TimesTen Configuration Options dialog, there is a check mark in the check box for Register TimesTen environment variables. If you choose to register the TimesTen environment variables, leave this box checked. Then choose **Next**.

> **Note:**
>
> - Settings made by registering environment variables are reflected in the System control panel and persist across sessions. It is advisable to reboot the system after the installation to ensure that Windows re-reads the registry and propagates the environment settings.
> - If you do not register environment variables during installation, you can run `ttenv.bat` from the TimesTen `bin` directory (in a DOS window) for each session. This sets `PATH`, `CLASSPATH`, and `TIMESTEN_HOME` (but not `LIB` and `INCLUDE`).
> - Registering environment variables is not advisable if you are installing multiple releases of TimesTen, in which case the path could cause unexpected behavior. Multiple installations must be from different major releases, such as `18.1.`*w.x* and `22.1.`*y.z*.

8. In the "Select which version of Java to add to the `CLASSPATH` " dialog, select one of the JDK options or select `None`. Then choose **Next**.
9. In the Installation Information dialog, confirm settings. Choose **Back** to make changes. Otherwise, choose **Next**.

The Setup Status dialog displays with a message indicating your new installation is being configured.

Additional dialogs and windows display, indicating that `timesten.conf` is updated and TimesTen is configured.

10. In the InstallShield Wizard Complete dialog, choose **Finish** (optionally choosing to display the TimesTen release notes).

Verify the installation on Windows

Verifying the installation on Windows includes:

- Verify proper installation on Windows
- Review the installation directories on Windows
- Review the timesten.conf file on Windows

Verify proper installation on Windows

To verify that the TimesTen client has been properly installed:

1. Confirm that the TimesTen 22.1 shortcut (for example, "TimesTen 22.1 (64-bit)") appears under **Apps & features** (or the equivalent).
2. From **Administrative Tools**, select **ODBC Data Sources (64-bit)**. This opens the ODBC Data Source Administrator (64-bit) menu. Choose the **Drivers** tab.
3. Confirm that the **TimesTen Client 22.1** driver is installed. Click **OK**.

Review the installation directories on Windows

This is a sample directory structure after a successful installation. Not all directories are included in this sample.

```
C:\TimesTen
   tt221_64
      bin [contains TimesTen utilities and executables]
      doc [contains Help files only]
      include [contains TimesTen include files]
      instance
         bin               [link to tt221_64\bin]
         conf
         diag
         info
         install           [link to tt221_64]
         plsql
            utl_file_temp  [for PL/SQL UTL_FILE package]
      lib
      plsql
```

Review the timesten.conf file on Windows

You may also review the `timesten.conf` file located in the `instance\conf` directory of the installation folder (for example, `C:\TimesTen\tt221_64\instance\conf`). The `daemon_port` value and the `server_port` value can be ignored. The `hostname` indicates the host name on which you performed the installation.

```
# TimesTen Instance Configuration File
# Commented values are default values
hostname=host1
timesten_release=22.1
instance_name=instance
daemon_port=
server_port=
tns_admin=:TNS_ADMIN:
admin_user=ttuser
userlog=C:\TimesTen\tt221_64\instance\diag\tterrors.log
supportlog=C:\TimesTen\tt221_64\instance\diag\ttmesg.log
client_only=yes
# By default, turn verbose logging on
verbose=1
```

Deleting an installation on Windows

This section discusses the steps to delete the installation and also the steps to verify the uninstallation was successful:

- Steps to delete the installation on Windows
- Verify the uninstallation is successful on Windows

Steps to delete the installation on Windows

Uninstall TimesTen on Windows through the **Apps & features** menu list, much as you would other Windows programs.

> **Note:**
>
> Uninstallation will delete the TimesTen installation as well as the TimesTen instance that was created as part of the installation.

1. From the **Apps & features** menu list, locate **Oracle TimesTen In-Memory Database 22.1 (64-bit)**.
2. Choose **Uninstall**.
3. The "This app and its related info will be uninstalled" dialog displays. Choose **Uninstall**.
4. In the Question dialog, the "Are you sure you want to uninstall TimesTen In-Memory Database Client 22.1?" question displays. Choose **Yes**.
5. In the Confirm Uninstall dialog, the "Do you want to completely remove the selected application and all its features?" question displays. Choose **OK**.
6. In the Question dialog, the "Would you like to delete all DSNs associated with TimesTen 22.1?" question displays. If you choose Yes, you will not be able to access your existing databases. Choose **Yes** or **No**.

> **Note:**
>
> The uninstallation process begins. Even for uninstallation, there is a dialog that says, "Configuring your new software installation."

7. The Maintenance Complete dialog displays indicating the uninstallation is complete. Choose **Finish**.

Verify the uninstallation is successful on Windows

To verify that uninstallation was successful, confirm that:

1. TimesTen 22.1 has been removed from **Apps & features**.
2. The TimesTen Client 22.1 driver is not listed in the **Drivers** tab in the ODBC Data Source Administrator.
3. The installation directory has been deleted (by default, `C:\TimesTen\tt221_64`). If not, delete it and any remaining contents manually.

   If this is the only TimesTen installation, you can delete the `TimesTen` directory. If there are other installations, then delete only the `tt221_64` directory.

> **Note:**
>
> If any files are open to Windows Explorer, you may have to use the Task Manager to end the `explorer` process before you can delete the installation directory and its contents.

5

Client-only Installations and Instances

This chapter discusses how to create and manage client installations and client instances. After you complete the necessary steps, you can use the TimesTen client to connect to either a database in TimesTen Scaleout or a database in TimesTen Classic. This chapter does not cover the Windows platform. See "Installation and Management of TimesTen on Windows" for information on the Windows platform.

Ensure you have reviewed the terminology and completed the prerequisites. See "Overview of the Installation Process in TimesTen Classic" for more information.

For information on TimesTen Scaleout, see "Overview of TimesTen Scaleout" in the *Oracle TimesTen In-Memory Database Scaleout User's Guide*.

The examples in this chapter are performed on a macOS host, but are applicable to the platforms that support client-only installations and client-only instances.

Topics include:

- Creating a TimesTen client installation
- Creating a TimesTen client instance
- Modifying a TimesTen client instance
- Destroying a TimesTen client instance
- Deleting a TimesTen client installation

Creating a TimesTen client installation

The instance administrator installs the client installation by unzipping the distribution. See "Distribution media and the distribution" and see "Instance administrator" for details.

Topics include:

- Create a client-only installation
- Verify a client installation

Create a client-only installation

To create the TimesTen client installation that is accessible by the instance administrator's primary group:

1. Place the appropriate distribution file in some desired, accessible location.
2. Create the desired directory for the installation.
3. Change directories to the desired directory for the installation.
4. Extract the distribution into that directory.

In the following example on a macOS host, `ttuser1`, with primary group `timesten`, creates a client installation into the directory `macclientinstall` (a subdirectory of the current directory)

from a distribution file in `/swdir/TimesTen/ttinstallers`. This installation is accessible to members of `timesten`.

```
% mkdir macclientinstall
% cd macclientinstall
% unzip /swdir/TimesTen/ttinstallers/timesten2211100.client.macos64.zip
[...UNZIP OUTPUT...]
```

The top level directory of the installed files is of the form `tt22.1.1.10.0`.

For example, this directory is created under `macclientinstall`:

```
dr-xr-x--- 19 ttuser1 timesten 4096 Aug  2 22:07 tt22.1.1.10.0
```

The `tt22.1.1.10.0` directory includes such files as (Not all files are listed):

```
dr-xr-x---  2 ttuser1 timesten    4096 Aug  2 22:07 bin
dr-xr-x---  3 ttuser1 timesten    4096 Aug  2 22:07 include
dr-xr-x---  2 ttuser1 timesten    4096 Aug  2 22:07 info
dr-xr-x---  2 ttuser1 timesten    4096 Aug  2 22:07 lib
dr-xr-x---  8 ttuser1 timesten    4096 Aug  2 22:07 plsql
dr-xr-x---  3 ttuser1 timesten    4096 Aug  2 22:07 ttoracle_home
```

Verify a client installation

To verify the installation, run the `ttInstallationCheck` utility. In addition, you can review the installation directory and subdirectories.

In this example, the installation is verified and is successful:

```
% installation_dir/tt22.1.1.10.0/bin/ttInstallationCheck
This installation has been verified.
```

See "ttInstanceCreate" in the *Oracle TimesTen In-Memory Database Reference* for information on the `ttInstanceCreate` utility.

A TimesTen client-only installation includes these subdirectories located under the top-level *`installation_dir`*`/tt22.1.1.10.0` directory (Not all of the subdirectories are included in this list).

- `bin`: TimesTen utilities and executables
- `include`: TimesTen include files, among them `timesten.h` (for TimesTen ODBC features) and `tt_errCode.h` (for information about TimesTen error codes)
- `lib`: TimesTen libraries
- `plsql`: Files and resources for TimesTen PL/SQL
- `ttoracle_home`: Oracle Database Instant Client files and resources, for OCI, Pro*C/C++, and ODP.NET

Creating a TimesTen client instance

After the instance administrator unzips the distribution, thereby creating an installation, the next step is to create a TimesTen client instance.To do this, you run the `ttInstanceCreate` utility, located in the *`installation_dir`*`/tt22.1.1.10.0/bin` directory.

Note:

- Only the user who unpacked the distribution (the creator of the installation) can create an instance from the installation.
- This user must be a member of the TimesTen users group.
- This user is the only user that can be the instance administrator of this instance and all other instances associated with the installation.

Topics include:

- Overview of the ttInstanceCreate utility
- Examples creating TimesTen client instances
- Review the instance home directory and subdirectories
- Review the timesten.conf file

Overview of the ttInstanceCreate utility

You can specify options for the `ttInstanceCreate` utility:

- On the command line
- In a file
- Interactively as the utility runs

You can also run `ttInstanceCreate` with the `-help` option to list the available options. See "ttInstanceCreate" in the *Oracle TimesTen In-Memory Database Reference* for detailed descriptions.

> **Note:**
>
> The `ttInstanceCreate -clientonly` option is optional for a client-only installation on a macOS or a Linux 32-bit host.

Examples creating TimesTen client instances

A TimesTen client instance can be used to connect either to a database in TimesTen Scaleout or a database in TimesTen Classic.

To create a client instance, navigate to the *`installation_dir`*`/tt22.1.1.10.0/bin` area and run the `ttInstanceCreate` utility.

After you create the client instance, you can set up your DSN and then follow the steps to connect to the server from the client. See "Working with the TimesTen Client and Server" in the *Oracle TimesTen In-Memory Database Operations Guide* for details.

These sections illustrate how to use the `ttInstanceCreate` utility to create a client instance:

- Use ttInstanceCreate to create a client instance interactively
- Use ttInstanceCreate to specify options on the command line

Use ttInstanceCreate to create a client instance interactively

This example runs the `ttInstanceCreate` utility interactively to create a client instance. You run the utility from the installation directory. User input is shown in bold.

```
% installation_dir/tt22.1.1.10.0/bin/ttInstanceCreate

* Client installation detected.

NOTE: Each TimesTen instance is identified by a unique name.
      The instance name must be a non-null alphanumeric string, not longer
      than 255 characters.

Please choose an instance name for this installation? [ tt221 ] clientmac
Instance name will be 'clientmac'.
Is this correct? [ yes ]
Where would you like to install the clientmac instance of TimesTen?
[/home/ttuser ] /scratch/ttuser
Creating instance in /scratch/ttuser/clientmac ...
INFO: Mapping files from the installation to
 /scratch/ttuser/clientmac/install
The 22.1 Release Notes are located here :
  '/installation_dir/tt22.1.1.10.0/README.html'
```

Use ttInstanceCreate to specify options on the command line

This example runs the `ttInstanceCreate` utility interactively to create a client instance. You run the utility from the installation directory. User input is shown in bold.

```
% installation_dir/tt22.1.1.10.0/bin/ttInstanceCreate

* Client installation detected.

NOTE: Each TimesTen instance is identified by a unique name.
      The instance name must be a non-null alphanumeric string, not longer
      than 255 characters.

Please choose an instance name for this installation? [ tt221 ] clientmac
Instance name will be 'clientmac'.
Is this correct? [ yes ]
Where would you like to install the clientmac instance of TimesTen?
[/home/ttuser ] /scratch/ttuser
Creating instance in /scratch/ttuser/clientmac ...
INFO: Mapping files from the installation to
 /scratch/ttuser/clientmac/install
The 22.1 Release Notes are located here :
  '/installation_dir/tt22.1.1.10.0/README.html'
```

Review the instance home directory and subdirectories

You can review the instance home directory and subdirectories for informational purposes. When you create an instance, each instance includes these subdirectories under *$TIMESTEN_HOME* (Not all of the subdirectories are included in this list):

- `bin`: TimesTen utilities and executables tailored and specific to the instance

This includes `ttenv.sh` (or `ttenv.csh`), which sets environment variables appropriately for the TimesTen environment for your session, and `setuproot.sh`, which can be run as root to cause data instances to be automatically started whenever the operating system reboots.

Note that `ttenv` also puts the `bin` directory in your path.

- `conf`: Contains the `timesten.conf` file, which is the TimesTen instance configuration file
- `diag`: Diagnostic output, including the daemon log and error log
- `info`: Working directory of the TimesTen daemon, containing persistent state about the TimesTen instance
- `install`: Symbolic link referencing the installation associated with this instance.
- `plsql`: Contains this subdirectory:
  - `utl_file_dir`: The only directory that can be read from or written to by PL/SQL blocks using the `UTL_FILE` package

Review the timesten.conf file

The instance configuration file defines the attributes of the TimesTen instance. See "Instance configuration file (timesten.conf)" for more information.

A sample file follows. Comments are indicated "#".

```
# TimesTen Instance Configuration File
# Created by ttInstanceCreate
# Commented values are default values
hostname=host1
client_only=yes
timesten_release=22.1
instance_name=clientmac
daemon_port=-
admin_user=myadmin
admin_uid=12345
group_name=ttgroup
instance_guid=FE8D9351-E749-41B4-AEC9-6E27BCA882A5
verbose=1
```

Modifying a TimesTen client instance

The `ttInstanceModify` utility modifies the attributes of the instance. The relevant `ttInstanceModify` options for a client instance are the `-install` and the `-tnsadmin` options.

The client instance that is modified is the instance that the `$TIMESTEN_HOME` environment variable references. Ensure you run the `ttInstanceModify` utility from the *`$TIMESTEN_HOME`*`/bin` directory.

See "ttInstanceModify" in the *Oracle TimesTen In-Memory Database Reference* for more information.

These sections illustrate how to use the `ttInstanceModify`:

- Use the ttinstanceModify utility interactively
- Use the ttInstanceModify utility with the -install option

Use the ttinstanceModify utility interactively

This example runs `ttInstanceModify` interactively. The utility asks if you want to change the installation that the instance points to. This is equivalent to running `ttInstanceModify` with the `-install` option.

This example upgrades from the `ttinstall` installation to the `ttinstalllatest` installation. The current directory is *`timesten_home`* for the `clientmac` instance.

```
% $TIMESTEN_HOME/bin/ttInstanceModify

Instance Info
-------------

Name:           clientmac
Version:        22.1.1.10.0
Location:       /scratch/ttuser/clientmac
Installation:   /swdir/TimesTen/ttinstallations/ttinstall/tt22.1.1.10.0

* Client-Only Installation

Would you like to change the installation that this instance points to? [ no ]
yes
Please provide the path to the new installation? [  ]
/swdir/TimesTen/ttinstallations/ttinstalllatest/tt22.1.1.11.0
Are you sure you want to point this instance to the installation in
/swdir/TimesTen/ttinstallations/ttinstalllatest/tt22.1.1.11.0? [ no ] yes


Instance Info (UPDATED)
-----------------------

Name:           clientmac
Version:        22.1.1.11.0
Location:       /scratch/ttuser/clientmac
Installation:   /swdir/TimesTen/ttinstallations/ttinstalllatest/tt22.1.1.11.0

* Client-Only Installation


The instance clientmac now points to the installation in
/swdir/TimesTen/ttinstallations/ttinstalllatest/tt22.1.1.11.0
```

Use the ttInstanceModify utility with the -install option

The example runs the `ttInstanceModify` utility with the `-install` option to upgrade from the `ttinstall` installation to the `ttinstalllatest` installation. The current directory is *`timesten_home`* for the `clientmac` instance.

```
% $TIMESTEN_HOME/bin/ttInstanceModify -install
 /swdir/TimesTen/ttinstallations/ttinstalllatest/tt22.1.1.11.0

Instance Info (UPDATED)
-----------------------

Name:           clientmac
Version:        22.1.1.11.0
Location:       /scratch/ttuser/clientmac
```

```
Installation:   /swdir/TimesTen/ttinstallations/ttinstalllatest/tt22.1.1.11.0

* Client-Only Installation


The instance clientmac now points to the installation in
/swdir/TimesTen/ttinstallations/ttinstalllatest/tt22.1.1.11.0
```

Destroying a TimesTen client instance

The `ttInstanceDestroy` utility destroys an existing instance. The instance that will be destroyed is based on the current setting of the `TIMESTEN_HOME` environment variable, which indicates the instance home directory. Run the `ttInstanceDestroy` utility that is located in the `bin` directory of the associated installation. (For example, *`installation_dir`*`/tt22.1.1.10.0/bin`.)

Only the instance administrator (the user who created the instance) can destroy the instance.

> **Note:**
>
> There can be no existing connections to databases associated with the instance you are destroying.

Run the `ttInstanceDestroy` utility with the `-help` option to list the available options. See "ttInstanceDestroy" in *Oracle TimesTen In-Memory Database Reference* for detailed descriptions.

> **Note:**
>
> The `-force` option completes the operation without prompting for confirmation.

This example runs the `ttInstanceDestroy` utility interactively to destroy the client. instance. The utility runs from the `installation_dir/tt22.1.1.10.0/bin` directory (the installation directory for the instance). Recall that the instance home directory (*`timesten_home`*) contains a subdirectory (`/install`) that is a symbolic link to the top-level directory of the instance's associated installation. See "Instance home" for information.

```
% installation_dir/tt22.1.1.10.0/bin/ttInstanceDestroy

** WARNING **

   The uninstallation has been executed by a non-root user.
   If the TimesTen daemon startup scripts were installed,
   you must run $TIMESTEN_HOME/bin/setuproot -uninstall
   to remove them. If you proceed with this uninstallation, you
   will have to remove the startup scripts manually.

** WARNING **

   All files in the directory:
```

```
   /scratch/ttuser/clientmac

   will be removed, including any file that you or other users may have created.

   Are you sure you want to completely remove this instance? [ yes ] yes
   Installation will remove all the files from
 /scratch/ttuser/clientmac.
   Do you want to continue? [ yes ] yes

NOTE: /scratch/ttuser/clientmac/info contains information related to the
      data stores that have been created with this release. If you remove
      /scratch/ttuser/clientmac/info
      you will no longer be able to access your data stores,
      nor would you be able to restore nor migrate your data.


NOTE: /scratch/ttuser/clientmac/conf contains information related to the instance
configuration.
/scratch/ttuser/clientmac Removed
The TimesTen instance clientmac has been destroyed.
```

Deleting a TimesTen client installation

To delete (uninstall) the client installation, the instance administrator performs these steps:

1. Ensures that there are no Timesten instances, databases or important files located under the TimesTen installation.
2. Ensures that no TimesTen instances use the installation. (There is no automated way to do this, as TimesTen does not maintain any inventory of TimesTen installations or instances.)

   If there are instances that use the installation, then before the installation is deleted, the instances must be either modified to point to a different installation or deleted. See "Modifying a TimesTen client instance" for details.
3. Grants write permission to the installation, such as through the `chmod -R` command. For example, from the installation directory where a TimesTen tt22.1.1.10.0 installation was extracted:

   ```
   % chmod -R 750 installation_dir/tt22.1.1.10.0
   ```
4. Deletes the `tt22.1.1.10.0` directory and all files and subdirectories that it contains, such as through the `rm -rf` command. From the installation directory:

   ```
   % rm -rf installation_dir/tt22.1.1.10.0
   ```
5. If the installation directory is empty, optionally deletes it.

6

Back Up, Restore, and Migrate Data in TimesTen Classic

This chapter describes the processes for backing up and restoring your database and for migrating the data in your database in TimesTen Classic.

This chapter is only relevant for TimesTen Classic. For information on the backup, restore, and migration processes for a database in TimesTen Scaleout, see "Migrating, Backing Up and Restoring Data" in the *Oracle TimesTen In-Memory Database Scaleout User's Guide*. For information on migrating a database from TimesTen Classic to TimesTen Scaleout, see "Migrating a database from TimesTen Classic to TimesTen Scaleout" in the *Oracle TimesTen In-Memory Database Scaleout User's Guide*.

Topics in this chapter include:

- Backing up and restoring a database
- Migrating a database

Backing up and restoring a database

The ability to backup and restore a database in TimesTen Classic is essential to protect your data. It is recommended that you perform regular backups to minimize the risk of potential data loss.

Use the ttBackup and ttRestore utilities to perform backup and restore procedures. See "ttBackup" and "ttRestore" in *Oracle TimesTen In-Memory Database Reference* for details. You can only run these utilities to backup and restore a database when:

- The first two digits of the version of the utility matches the first two digits of the version of the database and
- The platform of the utility matches the platform of the database

See "Overview of release numbers" for information on the format of release number.

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

Relevant information in this section include:

- Backup features
- Types of backups
- Restoring a database

Backup features

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

A backup operation is atomic. If it completes successfully, it produces a backup you can use to restore a database to the state of its backup point. See "ttBackup" in the *Oracle TimesTen In-Memory Database Reference* for more information.

Note:

- 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`, may compromise the integrity of the backup. In this event, restoration of the database from the backup may not be possible.
- Databases containing cache groups can be backed up with the `ttBackup` utility. However, when restoring such a backup, special consideration is required. The restored data within the cache groups may be out of date or out of sync with the data in the back-end Oracle database. To restore a database that contains cache groups, see "Backing up and restoring a TimesTen Classic database with cache groups" in the *Oracle TimesTen In-Memory Database Cache Guide* for more information.

Types of backups

TimesTen Classic supports these types of backups.

- *Full*: A full backup contains the most recent consistent checkpoint file and all the existing transaction log files. This backup saves the entire database. For full backups, ensure you have enough disk space available to hold both the existing backup and the new backup. A full backup can be a stream backup. A stream backup writes the database backup file to `stdout`.
- *Incremental*: An incremental backup moves the backup point of an existing backup forward in time by augmenting the backup with all the transaction log records created from the most recent incremental or full file incremental-enabled backup. 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 faster than a full backup, as it has less data to copy. The performance gain of incremental backups over full backups comes at the cost of increased disk usage and longer restoration times. Use incremental backups in concert with full backups to achieve a balance between backup time, disk usage, and restoration time. Incremental backups hold transaction logs, so you should run incremental backups regularly.

Before you can perform an incremental backup, you must enable your backup to allow for incremental backups by running the `ttBackup` utility 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 Classic supports the creation of up to eight incremental-enabled backup instances for each database. If you attempt to start a ninth incremental backup, TimesTen Classic 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. To enable incremental backups, you must again run the `ttBackup` utility with the `-fileFullEnable` or the `-fileIncrOrFull`.

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

The list of backup types supported by TimesTen Classic are included in this table. See "ttBackup" in the *Oracle TimesTen In-Memory Database Reference* for more information:

Backup type	File or stream	Full or incremental	Incremental-enabled	Comment
`fileFull`	File	Full	No	Default
`fileFullEnable`	File	Full	Yes	Full backup and resulting backup enabled with incremental backup
`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	Stream backup
`incrementalStop`	None	None	No	Does not perform a backup and disables incremental backups. Prevents transaction logs files from accumulating.

Restoring a database

Run the `ttRestore` utility to restore a database backup previously created with the `ttBackup` utility. See "ttRestore" in the *Oracle TimesTen In-Memory Database Reference* for details on this utility.

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

Backup and restore examples

You can backup and restore a database to a different directory on the same host. You can also backup and restore a database on a different host of the same platform type.

Examples:

- Performing a simple backup and restore
- Moving a database to a different directory
- Moving a database to a different host (same platform)

Performing a simple backup and restore

This example backs up the `database1` database to the `/scratch/backup` directory. It then runs `ttDestroy` to delete the database. (`ttRestore` returns an error if you try to overwrite an existing database.) Finally, it runs `ttRestore` to restore the `database1` database.

```
% ttBackup -dir /scratch/backup database1
Backup started ...
Backup complete

% ttDestroy database1

% ttRestore -dir /scratch/backup database1
Restore started ...
Restore complete
```

Moving a database to a different directory

The TimesTen daemon identifies a database by the full path name of the database checkpoint files. To move a 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.

This example moves a database from `/scratch/old/database1` with a database name `database1` to `/scratch/new/database1` with database name `database1_new`, using the `/scratch/tmp` directory for temporary storage.

1. Create a temporary directory.

   ```
   % mkdir /scratch/tmp
   ```

2. Disconnect applications from the database.
3. Back up the database to the temporary directory

```
% ttBackup -dir /scratch/tmp -fname database1 database1
Backup started ...
Backup complete
```

4. Unload the database from memory. See "Unload the database from memory" for information.
5. Create a new DSN, named `database1_new` and set the `DataStore` attribute to the full database path name of the database and the file name prefix:

```
[database1_new]
DataStore=/scratch/new/database1/database1_new
```

6. Restore the backup in the new location.

```
% ttRestore -dir /scratch/tmp -fname database1 database1_new
Restore started ...
Restore complete
```

7. Remove the temporary directory.

```
rm -r /scratch/tmp
```

8. After you confirm the database is operational in the new location, optionally, run the `ttDestroy` utility to destroy the database in the old location, and then delete the old DSN.

Moving a database to a different host (same platform)

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

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

1. Disconnect applications from the database.
2. Back up the database on the original host using `ttBackup`.

```
%ttBackup -dir /scratch/tmp -fname database1 database1
Backup started ...
Backup complete
```

3. Move the backup to the new host.
4. Create a DSN for the database on the new system. See "Creating a DSN on Linux and UNIX for TimesTen Classic" in the *Oracle TimesTen In-Memory Database Operations Guide* for more information.
5. Restore the backup on the new system using `ttRestore`.

```
% ttRestore -dir /scratch/tmp -fname database1 database1_new
Restore started ...
Restore complete
```

6. After the database is operational on the new host, you can use the `ttDestroy` utility to destroy the database on the old host, and you can then delete the old DSN.

Migrating a database

You can migrate your database from one major release of TimesTen Classic (such as 18.1) to another major release of TimesTen Classic (such as 22.1). You run the `ttMigrate` utility to

achieve this. `ttMigrate` saves database objects in a binary file and upgrades or downgrades database objects by restoring the objects from the binary file into the target database.

The binary files are platform-dependent. For example, a binary file produced on Linux must be restored on Linux. However, you can use the `ttMigrateCS` utility to copy a database between platforms (for example, between Linux and UNIX).

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

> **Note:**
>
> This section discusses the migration of databases in TimesTen Classic. For information on migrating databases from TimesTen Classic to TimesTen Scaleout and on migrating databases in TimesTen Scaleout, see "Migrating, Backing Up and Restoring Data" in the *Oracle TimesTen In-Memory Database Scaleout User's Guide*.

Examples using `ttMigrate` include:

- Moving to a different major release of TimesTen Classic
- Moving a database to a different platform

Moving to a different major release of TimesTen Classic

In TimesTen Classic, you can migrate data between major releases (for example, from TimesTen `18.1` to `22.1`) by using the `ttMigrate` utility to export the data from the old release and import it to the new release.

Prerequisites before migrating a database from one major release to another:

- Install the new release by unzipping the installation zip file. See "Installation of TimesTen Classic on Linux or UNIX" for more information.
- For the database in the old release, create a backup of the database. See "ttBackup" and "ttRestore" in *Oracle TimesTen In-Memory Database Reference* and "Backing up and restoring a database" for details.

Follow these steps:

1. On the old release, disconnect all applications from your database.
2. Save a copy of your database with the `ttMigrate` utility.

   ```
   % ttMigrate -c database1 /tmp/database1.data
   ...
   ```

   For more information about the `ttMigrate` utility, see "ttMigrate" in the *Oracle TimesTen In-Memory Database Reference*.
3. In the old release, unload the database from memory. See "Unload the database from memory" for details.

4. In the old release, if necessary, use the `ttDestroy` utility to destroy the old database. This step is required if the new database will be in the same location as the old database.
5. Copy the migrate object files to a file system that is accessible by the instance in the new release.
6. From the instance of the new release, create a database. Ensure that you have modified the connection attributes in the `sys.odbc.ini` (or `odbc.ini`) file and that you have sourced the environment variables and started the daemon.

   To create the database:

   ```
   % ttIsql -connstr "dsn=new_database1;AutoCreate=1" -e "quit"
   ```

   The database will be empty at this point.
7. From the instance of the new release, use `ttMigrate` with the `-r` and `-relaxedUpgrade` options to restore the backed up database to the new release. For example:

   ```
   % ttMigrate -r -relaxedUpgrade new_database1 /tmp/database1.data
   ```

Once the database is operational in the new release, create a backup of this database to have a valid restoration point for your database. Once you have created a backup of your database, you may delete the `ttMigrate` copy of your database (in this example, `/tmp/database1.data`). You can also delete the instance and installation from the old release.

Ensure you recompile and relink existing ODBC applications after you perform the upgrade and before you use the new release of TimesTen. See "Overview of ODBC API incompatibilities" in the *Oracle TimesTen In-Memory Database C Developer's Guide* for more information.

Moving a database to a different platform

The internal format of a database differs between platforms. You can use this procedure to move a database from one platform to another, such as from AIX to Linux, and reformat it for the target platform.

1. Disconnect applications from the database.
2. From the target system, use `ttMigrateCS` to connect in client/server mode to the database on the originating system and use the `-c` option to create a data file for the database on the target system. For example:

   ```
   ttMigrateCS -c database1 database1data.dat
   ...
   ```
3. On the target system, create a DSN for the database. See "Creating a DSN on Linux and UNIX for TimesTen Classic" in the *Oracle TimesTen In-Memory Database Operations Guide* for information.
4. Confirm that all connection attributes, including the `DataStore` connection attribute, are set correctly for the location on the new system.
5. Recreate the database using `AutoCreate=1`, such as in the following example:

   ```
   ttIsql -connstr "dsn=database2;AutoCreate=1" -e "quit"
   ```

   The database will be empty at this point.
6. On the target system, import the file created by `ttMigrateCS` into the new database using the `ttMigrate` utility with the `-r` option and the `-relaxedUpgrade` option. For example:

```
ttMigrate -r -relaxedUpgrade database2 database1data.dat
```

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

7

Upgrades in TimesTen Classic

This chapter describes the process for upgrading to a new release of TimesTen Classic. For information on the upgrade process for TimesTen Scaleout, see "Upgrading a grid" and "Migrating, Backing Up and Restoring Data" in the *Oracle TimesTen In-Memory Database Scaleout User's Guide*.

Ensure you review the installation process in the preceding chapters before completing the upgrade procedures described in this chapter.

Topics include:

Overview of release numbers

There is a release numbering scheme for TimesTen releases. This scheme is relevant when discussing upgrades. For example, for a given release, *a.b.c.d.e*:

- *a* indicates the first part of the major release.
- *b* indicates the second part of the major release.
- *c* indicates the patch set.
- *d* indicates the patch level within the patch set.
- *e* is reserved.

Important considerations:

- Releases within the same major release (*a.b*) are binary compatible. If a release is binary compatible, you do not have to recreate the database for the upgrade (or downgrade).

- Releases with a different major release are not binary compatible. In this case, you must recreate the database. See "Migrating a database" for details.

As an example, for the `22.1.1.10.0` release:

- The first two numbers of the five-place release number (`22.1`) indicate the major release.
- The third number of the five-place release number (`1`) indicates the patch set. For example, `22.1.1.10.0` is binary compatible with `22.1.1.11.0` because the first two digits in the five-place release number (`22` and `1`) are the same.
- The fourth number of the five-place release number (`1`) indicates the patch level within the patch set. `22.1.1.11.0` is the first patch level within patch set two.
- The fifth number of the five-place release number (`0`) is reserved.

Types of upgrades

TimesTen Classic supports two types of upgrades:

- An offline upgrade requires that you close all TimesTen databases to prevent future connections to these databases, you disconnect all applications from TimesTen, and you stop all TimesTen databases. This type of upgrade is useful when some amount of downtime is acceptable. During this downtime, the TimesTen databases are unavailable.

  An offline upgrade enables you to upgrade (or downgrade) to a new patch release or to upgrade to a major release:

  – Upgrade to a new patch release:

    * Run the `ttInstanceModify` utility with the `-install` option to upgrade the instance. See "About moving to a different patch release by modifying the instance" for details.

    * Run the `ttBackup` and the `ttRestore` utilities to upgrade patches and patch sets, although this is not the preferred method. See "Moving to a different patch release using ttBackup and ttRestore" for information.

  – Upgrade to a new major release: Run the `ttMigrate` utility to export a database to a flat file and then use `ttMigrate` again to import the data into the new database. This is the only method to perform an offline upgrade that involves moving between major releases. See "Moving to a different major release using ttMigrate" for details.

- An online upgrade involves using a pair of databases that are replicated and then performing an offline upgrade of each database in turn. This type of upgrade is useful when it is critically important that downtime be at a minimum. See "Online upgrade: Using TimesTen replication" for details.

About moving to a different patch release by modifying the instance

This section contains information about moving to a different patch release of TimesTen by modifying the TimesTen instance. Moving to a different patch release includes upgrades and downgrades. See "Moving to a different patch release using

ttBackup and ttRestore" for information on moving to a different patch release using backup and restore operations.

Concepts that are important when moving to a different patch release:

- *Start a database*: The subdaemon either creates a new shared memory segment or re-attaches to an existing one. These operations are used to start a database:
  - *Load*: The subdaemon creates a new shared memory segment, and loads the contents of the most recent checkpoint file into this new shared memory segment.
  - *Remap*: The subdaemon re-attaches to an existing shared memory segment.
- *Stop a database*: The subdaemon disconnects from the shared memory segment and either destroys the shared memory segment or preserves it. These operations are used to stop a database:
  - *Unload (clean)*: The shared memory segment is written to the checkpoint file on disk (by performing a static checkpoint operation). The subdaemon disconnects from and destroys the shared memory segment. The load operation starts the database.
  - *Detach (clean)*: The shared memory segment is optionally written to the checkpoint file on disk (by performing a static checkpoint operation). The subdaemon disconnects from the shared memory segment, but does not destroy it. The shared memory segment remains in memory. The remap operation starts the database.

> **Note:**
>
> For a complete list of the operations you use to start and stop a database, see "Managing TimesTen Databases" in the *Oracle TimesTen In-Memory Database Operations Guide*.

There are two types of patch upgrades (or downgrades):

- Basic patch upgrade: A type of upgrade where the shared memory segment is destroyed when the database is stopped. A new shared memory segment is created when the database is started. This is the preferred method for performing a patch upgrade. See "About performing a basic patch upgrade" for details.
- Fast patch upgrade: A type of upgrade where the shared memory segment is preserved in memory when the database is stopped. The same memory segment is used when the database is started. This is the preferred method if your databases are large and you have both critical uptime requirements and short maintenance windows. See "About performing a fast patch upgrade" for details.

About performing a basic patch upgrade

A basic patch upgrade is used when you do not have critical uptime requirements and short maintenance windows. When you stop the database, the contents of the shared memory segment is written to the checkpoint file on disk. The shared memory segment is then destroyed. When you start the database after an upgrade, a new shared memory segment is created and the contents of the checkpoint file are read into this newly created shared memory segment. Depending on the size of your database, the checkpoint operation performed when the database stops and the subsequent `ramLoad` operation performed when the database starts could be time consuming.

The process involves downloading the TimesTen full distribution (the upgrade release) and creating a new installation. The instance that requires upgrading is then modified to point to the new installation. The `ttInstanceModify` utility is used to perform this instance modification. As previously noted, you must close all TimesTen databases and disconnect all applications from TimesTen.

- Download and create the new installation
- Unload the database from memory
- Modify the instance to point to the new installation
- Load the database into memory
- Verify the patch upgrade

Download and create the new installation

To upgrade to a new patch release of TimesTen, you must first create the new installation.

1. Create the subdirectory into which you will download and unzip the new full distribution of TimesTen. Navigate to this directory and download the new full distribution into this directory. Then, use the `ZIP` utility to unpack this distribution. This example creates the *`new_installation_dir`* subdirectory and unpacks the `timesten2211110.server.linux8664.zip` file (the `22.1.1.11.0` full distribution for Linux 64-bit). Unzipping the `timesten2211110.server.linux8664.zip` file creates the new installation that will be used for this patch upgrade.

   ```
   % mkdir -p new_installation_dir
   % cd new_installation_dir
   ```

   Download the full distribution into the *`new_installation_dir`* subdirectory. Then use the `ZIP` utility to unpack the distribution.

   ```
   % unzip /timesten/installations/timesten2211110.server.linux8664.zip
   Archive:  /timesten/installations/
   timesten2211110.server.linux8664.zip
      creating: tt22.1.1.11.0/
   ...
   ```

2. Optional: Use the `ttInstallationCheck` utility, located in the `bin` subdirectory of the new installation (*`new_installation_dir`*`/bin` in this example) to verify the installation is successful.

   ```
   % new_installation_dir/tt22.1.1.11.0/bin/ttInstallationCheck
   This installation has been verified.
   ```

3. Optional: Verify the subdirectories are created under the full installation directory. These subdirectories may change from release to release.

   ```
   % ls new_installation_dir/tt22.1.1.11.0
   3rdparty     bin      info        network        plsql
   ttoracle_home
   ```

```
PERL         grid     kubernetes  nls            startup
README.html  include  lib         oraclescripts  support
```

You have successfully created the new installation.

Unload the database from memory

Perform the following steps to unload the database from memory.

1. Close the database. This prevents any future connections to the database.

   ```
   % ttAdmin -close database1
   RAM Residence Policy            : manual
   Manually Loaded In RAM          : True
   Replication Agent Policy        : manual
   Replication Manually Started    : False
   Cache Agent Policy              : manual
   Cache Agent Manually Started    : False
   Database State                  : Closed
   ```

   See "Opening and closing the database for user connections" in the *Oracle TimesTen In-Memory Database Operations Guide*.

2. Disconnect all applications from the database. See "Disconnecting from a database" in the *Oracle TimesTen In-Memory Database Operations Guide* for details.

3. Unload the database from memory. See "Specifying a RAM policy" in the *Oracle TimesTen In-Memory Database Operations Guide* for information on specifying a RAM policy.

   If the RAM policy is set to `always`, change it to `manual`, then unload the database from memory.

   ```
   % ttAdmin -ramPolicy manual -ramUnload database1

   RAM Residence Policy            : manual
   Manually Loaded In RAM          : False
   Replication Agent Policy        : manual
   Replication Manually Started    : False
   Cache Agent Policy              : manual
   Cache Agent Manually Started    : False
   Database state                  : closed
   ```

   If the RAM policy is set to `manual`, unload the database from memory.

   ```
   ttAdmin -ramUnload database1

   RAM Residence Policy            : manual
   Manually Loaded In RAM          : False
   Replication Agent Policy        : manual
   Replication Manually Started    : False
   Cache Agent Policy              : manual
   Cache Agent Manually Started    : False
   Database state                  : closed
   ```

   If the RAM policy is set to `inUse` and a grace period is set, set the grace period to `0` or wait for the grace period to elapse. TimesTen unloads a database with an `inUse` RAM policy from memory once all active connections are disconnected.

   ```
   % ttAdmin -ramGrace 0 database1
   ```

```
RAM Residence Policy           : inUse
Replication Agent Policy       : manual
Replication Manually Started   : False
Cache Agent Policy             : manual
Cache Agent Manually Started   : False
Database state                 : closed
```

Modify the instance to point to the new installation

The patch upgrade process requires you to modify the existing TimesTen instance to point to the new installation.
Perform these steps:

1. Use the `ttDaemonAdmin` utility to stop the TimesTen main daemon.

```
% ttDaemonAdmin -stop
TimesTen Daemon (PID: 21031, port: 6624) stopped.
```

2. Use the `ttInstanceModify` utility to modify the `myinstance` instance to point to the new installation. Recall that the TimesTen full distribution was unpacked and a new installation was created in *new_installation_dir*`/tt22.1.1.11.0`. See "Download and create the new installation" for details.

```
% $TIMESTEN_HOME/bin/ttInstanceModify -install
new_installation_dir/tt22.1.1.11.0

Instance Info (UPDATED)
-----------------------

Name:           myinstance
Version:        22.1.1.11.0
Location:       /scratch/ttuser/myinstance
Installation:   new_installation_dir/tt22.1.1.11.0
Daemon Port:    6624
Server Port:    6625


The instance myinstance now points to the installation in
new_installation_dir/tt22.1.1.11.0
```

3. Use the `ttDaemonAdmin` utility to restart the TimesTen main daemon. Then run the `ttVersion` utility to verify the `myinstance` instance has been upgraded to the new patch release (`22.1.1.11.0`, in this example).

```
% ttDaemonAdmin -start
TimesTen Daemon (PID: 20699, port: 6624) startup OK.


% ttVersion
TimesTen Release 22.1.1.11.0 (64 bit Linux/x86_64)
(myinstance:6624) 2021-09-15T16:53:47Z
  Instance admin: instanceadmin
  Instance home directory: /scratch/ttuser/myinstance
  Group owner: g900
```

```
Daemon home directory: /scratch/ttuser/myinstance/info
PL/SQL enabled.
```

You have successfully modified the instance to point to the new installation.

Load the database into memory

Follow these steps to load a database into memory.

1. Load the database into memory. This example sets the RAM policy to manual and then loads the `database1` database into memory.

   Set the RAM policy to `manual`.

   ```
   % ttAdmin -ramPolicy manual database1

   RAM Residence Policy            : manual
   Manually Loaded In RAM          : False
   Replication Agent Policy        : manual
   Replication Manually Started    : False
   Cache Agent Policy              : manual
   Cache Agent Manually Started    : False
   Database state                  : closed
   ```

   Load the `database1` database into memory.

   ```
   % ttAdmin -ramLoad database1

   RAM Residence Policy            : manual
   Manually Loaded In RAM          : True
   Replication Agent Policy        : manual
   Replication Manually Started    : False
   Cache Agent Policy              : manual
   Cache Agent Manually Started    : False
   Database state                  : closed
   ```

   See "Specifying a RAM policy" in the *Oracle TimesTen In-Memory Database Operations Guide* for information on the RAM policy.

2. Open the database for user connections.

   ```
   % ttAdmin -open database1
   RAM Residence Policy            : manual
   Manually Loaded In RAM          : True
   Replication Agent Policy        : manual
   Replication Manually Started    : False
   Cache Agent Policy              : manual
   Cache Agent Manually Started    : False
   Database State                  : Open
   ```

   See "Opening and closing the database for user connections" in the *Oracle TimesTen In-Memory Database Operations Guide*.

Verify the patch upgrade

Verify the patch upgrade:

- Verify the instance administrator user (`instanceadmin`, in this example) can connect to the `database1` database and perform a query.

```
% ttisql database1;

Copyright (c) 1996, 2021, Oracle and/or its affiliates. All rights
reserved.
Type ? or "help" for help, type "exit" to quit ttIsql.



connect "DSN=database1";
Connection successful: DSN=database1;UID=instanceadmin;DataStore=/
scratch/ttuser/database1;
DatabaseCharacterSet=AL32UTF8;ConnectionCharacterSet=AL32UTF8;LogBuf
MB=1024;
PermSize=500;TempSize=300;
(Default setting AutoCommit=1)
Command> connect adding "uid=user1;pwd=********" as user1;
Connection successful: DSN=database1;UID=user1;DataStore=/scratch/
ttuser/database1;
DatabaseCharacterSet=AL32UTF8;ConnectionCharacterSet=AL32UTF8;LogBuf
MB=1024;
PermSize=500;TempSize=300;
(Default setting AutoCommit=1)
user1: Command> SELECT COUNT (*) FROM employees;
< 107 >
1 row found.
```

You have successfully performed the patch upgrade.

About performing a fast patch upgrade

Consider performing a fast patch upgrade when you have large databases and you have both critical uptime requirements and short maintenance windows. During the fast patch upgrade, the static checkpoint operation performed at database stop is optional, and the shared memory segment is preserved after the subdaemon disconnects. When the database is started, the checkpoint operation is not performed and a new subdaemon connects to this preserved shared memory segment. This reduces the time it takes to upgrade an instance, especially if your databases are large, by skipping both the load of the database into memory operation and the checkpoint operation.

To use a fast patch upgrade, the `ramPolicy` for the database must be set to `enduring`. This keeps the database image in memory after the subdaemon disconnects. See "Specifying a RAM policy" in the *Oracle TimesTen In-Memory Database Operations Guide* for information on setting a RAM policy.

The size of the TimesTen shared memory segment must remain the same before and after the fast patch upgrade. The TimesTen `ttShmSize` utility is provided to calculate the size of the shared memory segment. Run this utility before you upgrade the instance and then run it again after you upgrade the instance to ensure the size of the shared memory segment has not changed after the upgrade. In addition, ensure you do not modify the `PermSize`, the `TempSize`, the `LogBufMB`, and the `Connections`

connection attributes after the upgrade. These attributes affect the size of the shared memory segment.

See "ttShmSize" and "Connection Attributes" in the *Oracle TimesTen In-Memory Database Reference* for information on the `ttShmSize` utility and the TimesTen connection attributes.

To perform a fast patch upgrade:

- Download and create the new installation
- Prepare to detach the subdaemon from the shared memory segment
- Detach the subdaemon from the shared memory segment
- Modify the instance to point to the new installation
- Attach a new subdaemon to the existing shared memory segment

Download and create the new installation

To upgrade to a new patch release of TimesTen, you must first create the new installation.

1. Create the subdirectory into which you will download and unzip the new full distribution of TimesTen. Navigate to this directory and download the new full distribution into this directory. Then, use the `ZIP` utility to unpack this distribution. This example creates the *`new_installation_dir`* subdirectory and unpacks the `timesten2211110.server.linux8664.zip` file (the `22.1.1.11.0` full distribution for Linux 64-bit). Unzipping the `timesten2211110.server.linux8664.zip` file creates the new installation that will be used for this patch upgrade.

   ```
   % mkdir -p new_installation_dir
   % cd new_installation_dir
   ```

   Download the full distribution into the *`new_installation_dir`* subdirectory. Then use the `ZIP` utility to unpack the distribution.

   ```
   % unzip /timesten/installations/timesten2211110.server.linux8664.zip
   Archive:  /timesten/installations/timesten2211110.server.linux8664.zip
      creating: tt22.1.1.11.0/
   ...
   ```

2. Optional: Use the `ttInstallationCheck` utility, located in the `bin` subdirectory of the new installation (*`new_installation_dir`*`/bin` in this example) to verify the installation is successful.

   ```
   % new_installation_dir/tt22.1.1.11.0/bin/ttInstallationCheck
   This installation has been verified.
   ```

3. Optional: Verify the subdirectories are created under the full installation directory. These subdirectories may change from release to release.

   ```
   % ls new_installation_dir/tt22.1.1.11.0
   3rdparty     bin      info        network        plsql    ttoracle_home
   PERL         grid     kubernetes  nls            startup
   README.html  include  lib         oraclescripts  support
   ```

You have successfully created the new installation.

Prepare to detach the subdaemon from the shared memory segment

Perform these operations on the instance created with the current release of TimesTen (`22.1.1.10.0`, in this example).

1. Optional: Run the `ttVersion` utility to verify the current TimesTen release (`22.1.1.10.0`, in this example).

```
% ttVersion
TimesTen Release 22.1.1.1.0 (64 bit Linux/x86_64) (myinstance:6624)
2021-09-16T07:41:05Z
  Instance admin: instanceadmin
  Instance home directory: /scratch/ttuser/myinstance
  Group owner: g900
  Daemon home directory: /scratch/ttuser/myinstance/info
  PL/SQL enabled.
```

2. Run the `ttStatus` utility to check if the database is open to user connections and if there are connections to the database (`database1`, in this example). In this example, the `database1` database is open and there are two connections to the database.

```
% ttStatus
TimesTen status report as of Fri Sep 24 05:46:05 2021

Daemon pid 21031 port 6624 instance myinstance
TimesTen server pid 21039 started on port 6625
------------------------------------------------------------------------
----
------------------------------------------------------------------------
----
Data store /scratch/ttuser/database1
Daemon pid 21031 port 6624 instance myinstance
TimesTen server pid 21039 started on port 6625
There are 14 connections to the data store
Shared Memory Key 0x0b100699 ID 547979276
PL/SQL Memory Key 0x0a100699 ID 547946502 Address 0x5000000000
Type            PID     Context             Connection
Name              ConnID
Process         15076   0x0000000001f09990
database1                           1
Process         15076   0x00000000020272b0
conn2                               2
Subdaemon       21036   0x0000000000f3c260
Manager                           2047
Subdaemon       21036   0x0000000000fbdbc0
Rollback                          2046
Subdaemon       21036   0x000000000103cf40  XactId
Rollback                 2037
Subdaemon       21036   0x00007f9fbc0008c0  Deadlock
Detector               2043
Subdaemon       21036   0x00007f9fc00008c0
```

```
Checkpoint                      2042
Subdaemon       21036   0x00007f9fc007f9e0  Garbage
Collector               2036
Subdaemon       21036   0x00007f9fc40008c0
Monitor                         2044
Subdaemon       21036   0x00007f9fcc0008c0
Flusher                         2045
Subdaemon       21036   0x00007f9fcc0a0e70
Aging                           2041
Subdaemon       21036   0x00007fa04c0008c0
HistGC                          2039
Subdaemon       21036   0x00007fa0501bbb70  Log
Marker                     2040
Subdaemon       21036   0x00007fa054048370
IndexGC                         2038
Open for user connections
Replication policy  : Manual
Cache Agent policy  : Manual
PL/SQL enabled.
------------------------------------------------------------------------
Accessible by group g900
End of report
```

3. Use the `ttAdmin` utility to close the `database1` database. This prevents further user connections.

```
% ttAdmin -close database1
RAM Residence Policy            : manual
Manually Loaded In RAM          : True
Replication Agent Policy        : manual
Replication Manually Started    : False
Cache Agent Policy              : manual
Cache Agent Manually Started    : False
Database State                  : Closed
```

4. Disconnect all applications from the database. Run the `ttStatus` utility to verify there are no connections to the database (`database1`, in this example).

```
% ttStatus
TimesTen status report as of Fri Sep 24 05:49:55 2021

Daemon pid 21031 port 6624 instance myinstance
TimesTen server pid 21039 started on port 6625
------------------------------------------------------------------------
------------------------------------------------------------------------
Data store /scratch/ttuser/database1
Daemon pid 21031 port 6624 instance myinstance
TimesTen server pid 21039 started on port 6625
There are 12 connections to the data store
Shared Memory Key 0x0b100699 ID 547979276
PL/SQL Memory Key 0x0a100699 ID 547946502 Address 0x5000000000
Type            PID     Context             Connection Name
ConnID
Subdaemon       21036   0x0000000000f3c260
```

```
Manager                         2047
Subdaemon       21036   0x0000000000fbdbc0
Rollback                        2046
Subdaemon       21036   0x000000000103cf40  XactId
Rollback                2037
Subdaemon       21036   0x00007f9fbc0008c0  Deadlock
Detector              2043
Subdaemon       21036   0x00007f9fc00008c0
Checkpoint                      2042
Subdaemon       21036   0x00007f9fc007f9e0  Garbage
Collector              2036
Subdaemon       21036   0x00007f9fc40008c0
Monitor                         2044
Subdaemon       21036   0x00007f9fcc0008c0
Flusher                         2045
Subdaemon       21036   0x00007f9fcc0a0e70
Aging                           2041
Subdaemon       21036   0x00007fa04c0008c0
HistGC                          2039
Subdaemon       21036   0x00007fa0501bbb70  Log
Marker                     2040
Subdaemon       21036   0x00007fa054048370
IndexGC                         2038
Closed to user connections
RAM residence policy: Manual
Data store is manually loaded into RAM
Replication policy  : Manual
Cache Agent policy  : Manual
PL/SQL enabled.
------------------------------------------------------------------------
----
Accessible by group g900
End of report
```

5. Run the `ttShmSize` utility to return the size of the shared memory segment. This size must match the size of the shared memory segment after the fast patch upgrade is completed.

   ```
   % ttShmSize -connStr DSN=database1
   The required shared memory size is 2148239512 bytes.
   ```

You have completed the preparatory steps to disconnect the subdaemon from the shared memory segment.

Detach the subdaemon from the shared memory segment

Perform these steps to disconnect the subdaemon from the shared memory segment.

1. Run the `ttAdmin` utility to check the `ramPolicy` for the `database1` database. In this example, the `ramPolicy` is set to `manual` and the `database1` database is manually loaded in RAM.

   ```
   % ttAdmin -query database1
   RAM Residence Policy            : manual
   ```

```
Manually Loaded In RAM          : True
Replication Agent Policy        : manual
Replication Manually Started    : False
Cache Agent Policy              : manual
Cache Agent Manually Started    : False
Database State                  : Closed
```

2. Use the `ttAdmin` utility to change the `ramPolicy` to `enduring`. The `enduring` setting preserves the shared memory segment in memory when the subdaemon disconnects from the shared memory segment.

```
% ttAdmin -ramPolicy enduring database1
RAM Residence Policy            : enduring
Manually Loaded In RAM          : True
Replication Agent Policy        : manual
Replication Manually Started    : False
Cache Agent Policy              : manual
Cache Agent Manually Started    : False
Database State                  : Closed
```

3. Use the `ttAdmin` utility with the `-shmDetach` option to disconnect the subdaemon from the shared memory segment.

```
% ttAdmin -shmDetach database1
RAM Residence Policy            : enduring
Manually Loaded In RAM          : False
Replication Agent Policy        : manual
Replication Manually Started    : False
Cache Agent Policy              : manual
Cache Agent Manually Started    : False
Database State                  : Closed
```

4. Use the `ttStatus` utility to verify the subdaemon is disconnected from the shared memory segment.

```
% ttStatus
TimesTen status report as of Fri Sep 24 06:12:04 2021

Daemon pid 21031 port 6624 instance myinstance
TimesTen server pid 21039 started on port 6625
------------------------------------------------------------------------
------------------------------------------------------------------------
Data store /scratch/ttuser/database1
Daemon pid 21031 port 6624 instance myinstance
TimesTen server pid 21039 started on port 6625
There are no connections to the data store
Closed to user connections
RAM residence policy: Enduring
Subdaemon is manually detached from data store (Shared Memory Key
0x0b100699 ID 547979276)
Replication policy  : Manual
Cache Agent policy  : Manual
PL/SQL enabled.
------------------------------------------------------------------------
```

```
Accessible by group g900
End of report
```

The subdaemon is disconnected from the shared memory segment. You are now ready to perform the patch upgrade.

Modify the instance to point to the new installation

The patch upgrade process requires you to modify the existing TimesTen instance to point to the new installation.
Perform these steps:

1. Use the `ttDaemonAdmin` utility to stop the TimesTen main daemon.

```
% ttDaemonAdmin -stop
TimesTen Daemon (PID: 21031, port: 6624) stopped.
```

2. Use the `ttInstanceModify` utility to modify the `myinstance` instance to point to the new installation. Recall that the TimesTen full distribution was unpacked and a new installation was created in *new_installation_dir*`/tt22.1.1.11.0`. See "Download and create the new installation" for details.

```
% $TIMESTEN_HOME/bin/ttInstanceModify -install
new_installation_dir/tt22.1.1.11.0

Instance Info (UPDATED)
-----------------------

Name:           myinstance
Version:        22.1.1.11.0
Location:       /scratch/ttuser/myinstance
Installation:   new_installation_dir/tt22.1.1.11.0
Daemon Port:    6624
Server Port:    6625


The instance myinstance now points to the installation in
new_installation_dir/tt22.1.1.11.0
```

3. Use the `ttDaemonAdmin` utility to restart the TimesTen main daemon. Then run the `ttVersion` utility to verify the `myinstance` instance has been upgraded to the new patch release (`22.1.1.11.0`, in this example).

```
% ttDaemonAdmin -start
TimesTen Daemon (PID: 20699, port: 6624) startup OK.


% ttVersion
TimesTen Release 22.1.1.11.0 (64 bit Linux/x86_64)
(myinstance:6624) 2021-09-15T16:53:47Z
  Instance admin: instanceadmin
  Instance home directory: /scratch/ttuser/myinstance
  Group owner: g900
```

```
Daemon home directory: /scratch/ttuser/myinstance/info
PL/SQL enabled.
```

You have successfully modified the instance to point to the new installation.

Attach a new subdaemon to the existing shared memory segment

Perform these steps to connect a new subdaemon to the existing shared memory segment:

1. Run the `ttShmSize` utility to return the size of the shared memory segment. This size must match the size of the shared memory segment before the patch upgrade. Recall the size was `2148239512 bytes`. See "Prepare to detach the subdaemon from the shared memory segment" for details.

   ```
   % ttShmSize -connStr DSN=database1
   The required shared memory size is 2148239512 bytes.
   ```

2. Use the `ttAdmin` utility to attach a new subdaemon to the existing shared memory segment.

   ```
   % ttAdmin -shmAttach database1
   RAM Residence Policy            : enduring
   Manually Loaded In RAM          : True
   Replication Agent Policy        : manual
   Replication Manually Started    : False
   Cache Agent Policy              : manual
   Cache Agent Manually Started    : False
   Database State                  : Closed
   ```

3. Use the `ttStatus` utility to verify the subdaemon is connected to the shared memory segment.

   ```
   % ttStatus
   TimesTen status report as of Fri Sep 24 06:35:10 2021

   Daemon pid 20699 port 6624 instance myinstance
   TimesTen server pid 20706 started on port 6625
   ------------------------------------------------------------------------
   ------------------------------------------------------------------------
   Data store /scratch/ttuser/database1
   Daemon pid 20699 port 6624 instance myinstance
   TimesTen server pid 20706 started on port 6625
   There are 12 connections to the data store
   Shared Memory Key 0x0b100699 ID 547979276
   PL/SQL Memory Key 0x0d100699 ID 548044806 Address 0x5000000000
   Type            PID     Context             Connection Name
   ConnID
   Subdaemon       20704   0x000000000207f260
   Manager                         2047
   Subdaemon       20704   0x0000000002100bc0
   Rollback                        2046
   Subdaemon       20704   0x000000000217ff40
   Aging                           2041
   Subdaemon       20704   0x00007f7ac40008c0
   ```

```
Checkpoint                      2042
Subdaemon       20704   0x00007f7ac407f9e0  Garbage
Collector               2040
Subdaemon       20704   0x00007f7acc0008c0
Monitor                         2045
Subdaemon       20704   0x00007f7acc0a0e70
IndexGC                         2038
Subdaemon       20704   0x00007f7ad00008c0  Deadlock
Detector              2043
Subdaemon       20704   0x00007f7ad007f9e0  XactId
Rollback                2039
Subdaemon       20704   0x00007f7ad40008c0
Flusher                         2044
Subdaemon       20704   0x00007f7ad407f9e0
HistGC                          2037
Subdaemon       20704   0x00007f7b580bed90  Log
Marker                     2036
Closed to user connections
RAM residence policy: Enduring
Data store is manually loaded into RAM
Replication policy  : Manual
Cache Agent policy  : Manual
PL/SQL enabled.
------------------------------------------------------------------------
----
Accessible by group g900
End of report
```

4. Use the `ttAdmin` utility to change the `ramPolicy` back to `manual`.

```
% ttAdmin -ramPolicy manual database1
RAM Residence Policy            : manual
Manually Loaded In RAM          : True
Replication Agent Policy        : manual
Replication Manually Started    : False
Cache Agent Policy              : manual
Cache Agent Manually Started    : False
Database State                  : Closed
```

5. Use the `ttAdmin` utility to open the `database1` database for user connections.

```
% ttAdmin -open database1;
RAM Residence Policy            : manual
Manually Loaded In RAM          : True
Replication Agent Policy        : manual
Replication Manually Started    : False
Cache Agent Policy              : manual
Cache Agent Manually Started    : False
Database State                  : Open
```

6. Verify the instance administrator user (`instanceadmin`, in this example) can connect to the `database1` database and perform a query.

```
% ttIsql database1;

Copyright (c) 1996, 2021, Oracle and/or its affiliates. All rights
reserved.
Type ? or "help" for help, type "exit" to quit ttIsql.



connect "DSN=database1";
Connection successful: DSN=database1;UID=instanceadmin;DataStore=/scratch/
ttuser/database1;
DatabaseCharacterSet=AL32UTF8;ConnectionCharacterSet=AL32UTF8;LogBufMB=102
4;PermSize=500;
TempSize=300;
(Default setting AutoCommit=1)
Command> connect adding "uid=user1;pwd=********" as user1;
Connection successful: DSN=database1;UID=user1;DataStore=/scratch/ttuser/
database1;
DatabaseCharacterSet=AL32UTF8;ConnectionCharacterSet=AL32UTF8;LogBufMB=102
4;PermSize=500;
TempSize=300;
(Default setting AutoCommit=1)
user1: Command> SELECT COUNT (*) FROM employees;
< 107 >
1 row found.
```

A new subdaemon connected to the preserved shared memory segment. The fast patch upgrade is successful.

Moving to a different patch release using ttBackup and ttRestore

You can run the `ttBackup` and `ttRestore` utilities to move to a new patch release, although this is not the preferred method. See "About moving to a different patch release by modifying the instance" for the preferred method.

Perform these steps for each database.

On the old release:

1. Use the `ttAdmin` utility to close the `database1` database. This prevents further user connections.

```
% ttAdmin -close database1
RAM Residence Policy            : manual
Manually Loaded In RAM          : True
Replication Agent Policy        : manual
Replication Manually Started    : False
Cache Agent Policy              : manual
Cache Agent Manually Started    : False
Database State                  : Closed
```

2. Disconnect all applications from the database. Run the `ttStatus` utility to verify there are no connections to the database (`database1`, in this example).

```
% ttStatus
TimesTen status report as of Sat Oct  2 04:37:10 2021

Daemon pid 4649 port 6624 instance myinstance
TimesTen server pid 4656 started on port 6625
------------------------------------------------------------------------
----
------------------------------------------------------------------------
----
Data store /scratch/ttuser/database1
Daemon pid 4649 port 6624 instance myinstance
TimesTen server pid 4656 started on port 6625
There are no connections to the data store
Closed to user connections
RAM residence policy: manual
Data store is manually loaded into RAM
Replication policy  : Manual
Cache Agent policy  : Manual
PL/SQL enabled.
------------------------------------------------------------------------
----
Accessible by group g900
End of report
```

3. Run the `ttVersion` utility to verify the current release (`22.1.1.10.0`, in this example).

```
% ttVersion
TimesTen Release 22.1.1.1.0 (64 bit Linux/x86_64) (myinstance:6624)
2021-09-16T07:41:05Z
  Instance admin: instanceadmin
  Instance home directory: /scratch/ttuser/myinstance20/myinstance
  Group owner: g900
  Daemon home directory: /scratch/ttuser/myinstance/myinstance/info
  PL/SQL enabled.
```

4. Backup the database. In this example, backup the `database1` database for release `22.1.1.10.0`.

```
% ttBackup -dir /tmp/dump/backup -fname database1_2211 database1
Backup started ...
Backup complete
```

5. Unload the database from memory. This example assumes a RAM policy of manual. See "Specifying a RAM policy" in the *Oracle TimesTen In-Memory Database Operations Guide* for information on the RAM policy.

```
% ttAdmin -ramUnload database1
RAM Residence Policy            : manual
Manually Loaded In RAM          : False
Replication Agent Policy        : manual
```

```
Replication Manually Started    : False
Cache Agent Policy              : manual
Cache Agent Manually Started    : False
Database State                  : Closed
```

6. Stop the TimesTen main daemon.

```
% ttDaemonAdmin -stop
TimesTen Daemon (PID: 4649, port: 6624) stopped.
```

For the new release:

1. Create the subdirectory into which you will download and unzip the new full distribution of TimesTen. Navigate to this directory and download the new full distribution into this directory. Then, use the `ZIP` utility to unpack this distribution. This example creates the *`new_installation_dir`* subdirectory and unpacks the `timesten2211110.server.linux8664.zip` file (the `22.1.1.11.0` full distribution for Linux 64-bit). Unzipping the `timesten2211110.server.linux8664.zip` file creates the new installation that will be used for this patch upgrade.

```
% mkdir -p new_installation_dir
% cd new_installation_dir
```

Download the full distribution into the *`new_installation_dir`* subdirectory. Then use the `ZIP` utility to unpack the distribution.

```
% unzip /timesten/installations/timesten2211110.server.linux8664.zip
Archive:  /timesten/installation/timesten2211110.server.linux8664.zip
   creating: tt22.1.1.11.0/
...
```

2. Run the `ttInstanceCreate` utility to create the instance. This example runs the `ttInstanceCreate` utility interactively. See "ttInstanceCreate" in the *Oracle TimesTen In-Memory Database Reference* and "Creating an instance on Linux/UNIX: Basics" in this book for details.

Navigate to the *`new_installation_dir`*`/tt22.1.1.11.0/bin` area of the installation directory and then run the `ttInstanceCreate` utility located in that directory. The `ttInstanceCreate` utility must be run from the installation directory. User input is shown in bold.

```
% new_installation_dir/bin/ttInstanceCreate

NOTE: Each TimesTen instance is identified by a unique name.
      The instance name must be a non-null alphanumeric string, not longer
      than 255 characters.

Please choose an instance name for this installation? [ tt221 ] myinstance
Instance name will be 'myinstance30'.
Is this correct? [ yes ]
Where would you like to install the myinstance instance of TimesTen? [ /
home/ttuser ] /scratch/ttuser
The directory /scratch/ttuser/ does not exist.
Do you want to create it? [ yes ]
```

```
Creating instance in /scratch/ttuser/myinstance ...

NOTE: If you are configuring TimesTen for use with Oracle
Clusterware, the
      daemon port number must be the same across all TimesTen
installations
      managed within the same Oracle Clusterware cluster.

NOTE: All installations that replicate to each other must use the
same daemon
      port number that is set at installation time. The daemon port
number can
      be verified by running 'ttVersion'.

The default port number is 6624.

Do you want to use the default port number for the TimesTen daemon?
[ yes ]
The daemon will run on the default port number (6624).

In order to use the cache features in any TimesTen databases
created within this instance, you must set a value for the TNS_ADMIN
environment variable. It can be left blank, and a value can be
supplied later
using <install_dir>/bin/ttInstanceModify.

Please enter a value for TNS_ADMIN (s=skip)? [  ] s
What is the TCP/IP port number that you want the TimesTen Server to
listen on? [ 6625 ]

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

Would you like to use systemd to manage TimesTen? [ no ]

NOTE: The TimesTen daemon startup/shutdown scripts have not been
installed.

The startup script is located here :
        '/scratch/ttuser/myinstance/startup/tt_myinstance'

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

The 22.1 Release Notes are located here :
  'new_installation_dir/tt22.1.1.11.0/README.html'

Starting the daemon ...
TimesTen Daemon (PID: 11121, port: 6624) startup OK.
Instance created successfully.
```

3. Restore the database. Source the environment variables, make all necessary changes to your connection attributes in the `sys.odbc.ini` (or the `odbc.ini`) file, and start the daemon (if not already started) prior to restoring the database.

```
% ttRestore -dir /tmp/dump/backup -fname database1_2211 database1
Restore started ...
Restore complete
```

Once your databases are correctly configured and fully operational, you can optionally remove the backup file (in this example, `/tmp/dump/backup/database1_2211`).

Moving to a different major release using ttMigrate

Moving to a different major release is done through migration. Migration includes upgrading from one major TimesTen release to a new major TimesTen release, or changing the operating system platform that TimesTen runs on.

Migration involves copying out the schema and data from one database, creating a new database with the new release, and then creating the schema and inserting the data into the new database. The `ttMigrate` utility is used to automate the migration of databases. See "ttMigrate" in the *Oracle TimesTen In-Memory Database Reference* for information on the `ttMigrate` utility.

Before migrating a database from one major release to another, ensure you backup the database in the old release. See "ttBackup" and "ttRestore" in *Oracle TimesTen In-Memory Database Reference* and "Backing up and restoring a database" in this book for details.

Follow these steps to perform the upgrade:

For the old release:

1. Use the `ttAdmin` utility to close the `database1` database. This prevents further user connections.

```
% ttAdmin -close database1
RAM Residence Policy            : manual
Manually Loaded In RAM          : True
Replication Agent Policy        : manual
Replication Manually Started    : False
Cache Agent Policy              : manual
Cache Agent Manually Started    : False
Database State                  : Closed
```

2. Disconnect all applications from the database. Run the `ttStatus` utility to verify there are no connections to the database (`database1`, in this example).

```
% ttStatus
TimesTen status report as of Sat Oct  2 18:31:59 2021

Daemon pid 28436 port 6624 instance myinstance
TimesTen server pid 28443 started on port 6625
------------------------------------------------------------------------
------------------------------------------------------------------------
Data store /scratch/ttuser/database1
Daemon pid 28436 port 6624 instance myinstance
```

```
TimesTen server pid 28443 started on port 6625
There are 13 connections to the data store
Shared Memory KEY 0x061014ae ID 491521
PL/SQL Memory Key 0x071014ae ID 524290 Address 0x5000000000
Type            PID     Context             Connection
Name              ConnID
Subdaemon       28440   0x0000000001893250
Manager                         2047
Subdaemon       28440   0x0000000001914210
Rollback                        2046
Subdaemon       28440   0x00007f55d80008c0  Deadlock
Detector              2043
Subdaemon       28440   0x00007f55d807f330  Log
Marker                   2040
Subdaemon       28440   0x00007f55dc0008c0
Monitor                         2044
Subdaemon       28440   0x00007f55dc07f330
AsyncMV                         2039
Subdaemon       28440   0x00007f55e00008c0
Checkpoint                      2042
Subdaemon       28440   0x00007f55e007f330
Aging                           2041
Subdaemon       28440   0x00007f55e40008c0
Flusher                         2045
Subdaemon       28440   0x00007f55e40a6970
HistGC                          2038
Subdaemon       28440   0x00007f56600008c0  XactId
Rollback                2036
Subdaemon       28440   0x00007f56641b9cb0
IndexGC                         2037
Subdaemon       28440   0x00007f5668048360  Garbage
Collector               2035
Closed to user connections
RAM residence policy: Manual
Data store is manually loaded into RAM
Replication policy  : Manual
Cache Agent policy  : Manual
PL/SQL enabled.
------------------------------------------------------------------------
----
Accessible by group g900
End of report
```

3. Run the `ttVersion` utility to verify the current release.

```
% ttVersion
TimesTen Release tt22.1.1.10.0 (64 bit Linux/x86_64)
(myinstance:6624) 2021-09-12T07:34:06Z
  Instance admin: instanceadmin
  Instance home directory: /scratch/ttuser/myinstance
  Group owner: g900
  Daemon home directory: /scratch/ttuser/myinstance/info
  PL/SQL enabled.
```

4. Use the `ttMigrate` utility to copy out the schema and data from the database (`database1`, in this example).

```
% ttMigrate -c database1 /tmp/database1.data

Saving profile DEFAULT
Profile successfully saved.

Saving profile SYSTEM
Profile successfully saved.

Saving user PUBLIC
User successfully saved.

Saving table TTUSER.COUNTRIES
  Saving foreign key constraint COUNTR_REG_FK
  Saving rows...
  25/25 rows saved.
Table successfully saved.

Saving table TTUSER.DEPARTMENTS
  Saving foreign key constraint DEPT_LOC_FK
  Saving rows...
  27/27 rows saved.
Table successfully saved.

Saving table TTUSER.EMPLOYEES
  Saving index TTUSER.TTUNIQUE_0
  Saving foreign key constraint EMP_DEPT_FK
  Saving foreign key constraint EMP_JOB_FK
  Saving rows...
  107/107 rows saved.
Table successfully saved.

Saving table TTUSER.JOBS
  Saving rows...
  19/19 rows saved.
Table successfully saved.

Saving table TTUSER.JOB_HISTORY
  Saving foreign key constraint JHIST_DEPT_FK
  Saving foreign key constraint JHIST_EMP_FK
  Saving foreign key constraint JHIST_JOB_FK
  Saving rows...
  10/10 rows saved.
Table successfully saved.

Saving table TTUSER.LOCATIONS
  Saving foreign key constraint LOC_C_ID_FK
  Saving rows...
  23/23 rows saved.
Table successfully saved.

Saving table TTUSER.REGIONS
  Saving rows...
```

```
  4/4 rows saved.
Table successfully saved.

Saving view TTUSER.EMP_DETAILS_VIEW
View successfully saved.

Saving sequence TTUSER.DEPARTMENTS_SEQ
Sequence successfully saved.

Saving sequence TTUSER.EMPLOYEES_SEQ
Sequence successfully saved.

Saving sequence TTUSER.LOCATIONS_SEQ
Sequence successfully saved.
```

5. Unload the database from memory. This example assumes a RAM policy of manual. See "Specifying a RAM policy" in the *Oracle TimesTen In-Memory Database Operations Guide* for information on the RAM policy.

```
% ttAdmin -ramUnload database1
RAM Residence Policy            : manual
Manually Loaded In RAM          : False
Replication Agent Policy        : manual
Replication Manually Started    : False
Cache Agent Policy              : manual
Cache Agent Manually Started    : False
Database State                  : Closed
```

6. Stop the TimesTen main daemon.

```
% ttDaemonAdmin -stop
TimesTen Daemon (PID: 28436, port: 6624) stopped.
```

7. Copy the migrated object files (`/tmp/database1.data`, in this example) to a file system that is accessible by the instance in the new release.

For the new release:

1. Create the subdirectory into which you will download and unzip the new full distribution of TimesTen. Navigate to this directory and download the new full distribution into this directory. Then, use the `ZIP` utility to unpack this distribution. This example creates the *`new_installation_dir`* subdirectory and unpacks the `timesten2211100.server.linux8664.zip` file (the `22.1.1.10.0` full distribution for Linux 64-bit). Unzipping the `timesten2211100.server.linux8664.zip` file creates the new installation that will be used for this patch upgrade.

```
% mkdir -p new_installation_dir
% cd new_installation_dir
```

Download the full distribution into the *`new_installation_dir`* subdirectory. Then use the `ZIP` utility to unpack the distribution.

```
% unzip /timesten/installations/timesten2211100.server.linux8664.zip
Archive:  timesten/installations/
```

```
timesten2211100.server.linux8664.zip
   creating: tt2211100/
...
```

2. Run the `ttInstanceCreate` utility to create the instance. This example runs the `ttInstanceCreate` utility interactively. See "ttInstanceCreate" in the *Oracle TimesTen In-Memory Database Reference* and "Creating an instance on Linux/UNIX: Basics" in this book for details.

   User input is shown in bold.

```
% new_installation_dir/bin/ttInstanceCreate

NOTE: Each TimesTen instance is identified by a unique name.
      The instance name must be a non-null alphanumeric string, not longer
      than 255 characters.

Please choose an instance name for this installation? [ tt221 ] myinstance
Instance name will be 'myinstance'.
Is this correct? [ yes ]
Where would you like to install the myinstance instance of TimesTen? [ /
home/ttuser ] /scratch/ttuser
The directory /scratch/ttuser/myinstance does not exist.
Do you want to create it? [ yes ]
Creating instance in /scratch/ttuser/myinstance ...

NOTE: If you are configuring TimesTen for use with Oracle Clusterware, the
      daemon port number must be the same across all TimesTen
installations
      managed within the same Oracle Clusterware cluster.

NOTE: All installations that replicate to each other must use the same
daemon
      port number that is set at installation time. The daemon port
number can
      be verified by running 'ttVersion'.

The default port number is 6624.

Do you want to use the default port number for the TimesTen daemon?
[ yes ]
The daemon will run on the default port number (6624).

In order to use the cache features in any TimesTen databases
created within this instance, you must set a value for the TNS_ADMIN
environment variable. It can be left blank, and a value can be supplied
later
using <install_dir>/bin/ttInstanceModify.

Please enter a value for TNS_ADMIN (s=skip)? [  ] s
What is the TCP/IP port number that you want the TimesTen Server to
listen on? [ 6625 ]

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

```
Would you like to use systemd to manage TimesTen? [ no ]

NOTE: The TimesTen daemon startup/shutdown scripts have not been
installed.

The startup script is located here :
        '/scratch/ttuser/myinstance/startup/tt_myinstance'

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

The 22.1 Release Notes are located here :
  'new_installation_dir/tt22.1.1.1.0/README.html'

Starting the daemon ...
TimesTen Daemon (PID: 2214, port: 6624) startup OK.
Instance created successfully.
```

3. From the instance of the new release, create a database. Ensure you have sourced the environment variables, made all necessary changes to your connection attributes in the `sys.odbc.ini` (or the `odbc.ini`) file, and started the daemon (if not already started).

   To create the database:

```
% ttIsql -connStr "DSN=mynewdatabase;AutoCreate=1" -e "quit"

Copyright (c) 1996, 2021, Oracle and/or its affiliates. All rights
reserved.
Type ? or "help" for help, type "exit" to quit ttIsql.



connect "DSN=mynewdatabase;AutoCreate=1";
Connection successful: DSN=mynewdatabase;UID=ttuser;DataStore=/
scratch/ttuser/mynewdatabase;
DatabaseCharacterSet=AL32UTF8;ConnectionCharacterSet=AL32UTF8;LogBuf
MB=1024;PermSize=500;
TempSize=300;
(Default setting AutoCommit=1)

quit;
Disconnecting...
Done.
```

   The database will be empty at this point.

4. From the instance of the new release, run the `ttMigrate` utility with the `-r` and `-relaxedUpgrade` options to restore the backed up database to the new release. For example:

```
% $TIMESTEN_HOME/bin/ttMigrate -r -relaxedUpgrade
mynewdatabase /tmp/database1.data
```

```
Restoring profile DEFAULT
Profile successfully restored.

Restoring profile SYSTEM
Profile successfully restored.

Restoring table TTUSER.JOBS
  Restoring rows...
  19/19 rows restored.
Table successfully restored.

Restoring table TTUSER.REGIONS
  Restoring rows...
  4/4 rows restored.
Table successfully restored.

Restoring table TTUSER.COUNTRIES
  Restoring rows...
  25/25 rows restored.
  Restoring foreign key dependency COUNTR_REG_FK on TTUSER.REGIONS
Table successfully restored.

Restoring table TTUSER.LOCATIONS
  Restoring rows...
  23/23 rows restored.
  Restoring foreign key dependency LOC_C_ID_FK on TTUSER.COUNTRIES
Table successfully restored.

Restoring table TTUSER.DEPARTMENTS
  Restoring rows...
  27/27 rows restored.
  Restoring foreign key dependency DEPT_LOC_FK on TTUSER.LOCATIONS
Table successfully restored.

Restoring table TTUSER.EMPLOYEES
  Restoring rows...
  107/107 rows restored.
  Restoring foreign key dependency EMP_DEPT_FK on TTUSER.DEPARTMENTS
  Restoring foreign key dependency EMP_JOB_FK on TTUSER.JOBS
Table successfully restored.

Restoring table TTUSER.JOB_HISTORY
  Restoring rows...
  10/10 rows restored.
  Restoring foreign key dependency JHIST_DEPT_FK on TTUSER.DEPARTMENTS
  Restoring foreign key dependency JHIST_EMP_FK on TTUSER.EMPLOYEES
  Restoring foreign key dependency JHIST_JOB_FK on TTUSER.JOBS
Table successfully restored.

Restoring view TTUSER.EMP_DETAILS_VIEW
View successfully restored.

Restoring sequence TTUSER.DEPARTMENTS_SEQ
Sequence successfully restored.
```

```
Restoring sequence TTUSER.EMPLOYEES_SEQ
Sequence successfully restored.

Restoring sequence TTUSER.LOCATIONS_SEQ
Sequence successfully restored.
```

Once the database is operational in the new release, create a backup of this database to have a valid restoration point for your database. Once you have created a backup of your database, you may delete the `ttMigrate` copy of your database (in this example, `/tmp/database1.data`). Optionally, for the old release, you can remove the instance and delete the installation.

Ensure you recompile and relink existing ODBC and OCI applications after you perform the upgrade and before you use the new release of TimesTen. See "Overview of ODBC API incompatibilities" in the *Oracle TimesTen In-Memory Database C Developer's Guide* for more information.

Online upgrade: Using TimesTen replication

When upgrading to a new release of TimesTen Classic, 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 instance for the other database is being upgraded. When the upgrade is finished, any updates that have been made on the active database are transmitted immediately to the database in the upgraded instance, and your applications can then be switched with no data loss and no downtime. See "Performing an online upgrade with classic replication" for information.

The online upgrade process supports only updates to user tables during the upgrade. The tables to be replicated must have a `PRIMARY KEY` or a unique index on non-nullable columns. Data definition changes such as `CREATE TABLE` or `CREATE INDEX` are not replicated except in the case for an active standby pair with `DDLReplicationLevel` set to `2`. In the latter case, `CREATE TABLE` and `CREATE INDEX` are replicated.

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

Note:

- Online major upgrades for active standby pairs with cache groups are only supported for read-only cache groups.
- Online major upgrades for active standby pairs that are managed by Oracle Clusterware are not supported.

Performing an online upgrade with classic replication

This section describes how to use the TimesTen replication feature to perform online upgrades for applications that require continuous data availability.

This procedure is for classic replication in a unidirectional, bidirectional, or multidirectional scenario.

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

Note the following:

- For active standby pairs, see "Online upgrades for an active standby pair with no cache groups" and "Online upgrades for an active standby pair with cache groups" for details. Online major upgrades for active standby pairs with cache groups are only supported for read-only cache groups. Instead see "Offline upgrades for an active standby pair with cache groups" for this information.
- For the use of Oracle Clusterware, see "Performing an online TimesTen upgrade when using Oracle Clusterware" for information. Online major upgrades are not supported for active standby pairs managed by Oracle Clusterware.

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

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* for information.

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 `v$monitor` view, using `ttIsql`:

```
Command> SELECT perm_in_use_size FROM v$monitor;
```

Upgrade steps

The following steps illustrate how to perform an online upgrade while replication is running. The *upgrade* host is the host on which the database upgrade is being performed, and the *active* host is the host containing the database to which the application remains connected.

Step	Upgrade host	Active host
1.	Configure replication to replicate to the active host using static ports.	Configure replication to replicate to the upgrade host using static ports.

Step	Upgrade host	Active host
2.	n/a	Connect all applications to the active database, if they are not connected.
3.	Disconnect all applications from the database that will be upgraded.	n/a
4.	n/a	Set replication to the upgrade host to the `PAUSE` state.
5.	Wait for updates to propagate to the active host.	n/a
6.	Stop replication.	n/a
7.	Back up the database with `ttMigrate -c` and run `ttDestroy` to destroy the database.	n/a
8.	Stop the TimesTen daemon for the old release.	n/a
9.	Create a new installation and a new instance for the new release. See "Creating an installation on Linux/UNIX" and "Creating an instance on Linux/UNIX: Basics" for information.	n/a
10.	Create a DSN for the post-upgrade database for the new release. Adjust parallelism options for the DSN.	n/a
11.	Restore the database from the backup with `ttMigrate -r`.	n/a
12.	Clear the replication bookmark and logs using `ttRepAdmin -receiver -reset` and by setting replication to the active host to the `stop` and then the `start` state.	n/a
13.	Start replication.	n/a
14.	n/a	Set replication to the upgrade host to the `start` state, ensuring that the accumulated updates propagate once replication is restarted.
15.	n/a	Start replication.
16.	n/a	Wait for all of the updates to propagate to the upgrade host.
17.	Reconnect all applications to the post-upgrade database.	n/a

After the above procedures are completed on the upgrade host, the active host can be upgraded using the same steps.

Online upgrade example

This section describes how to perform an online upgrade in a scenario with two bidirectionally replicated databases.

In the following discussion, the two hosts are referred to as the *upgrade* host, on which the instance (with its databases) is being upgraded, and the *active* host, which remains operational and connected to the application for the duration of the upgrade.

After the procedure is completed, the same steps can be followed to upgrade the active host. However, you may prefer to delay conversion of the active host to first test the upgraded instance.

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

Follow these steps in the order they are presented:

Step	Upgrade host	Active host

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 host must start running on the active host instead.	Use the `ttRepAdmin` utility to pause replication from the database `active` to the database `upgrade`: `ttRepAdmin -receiver -name upgrade -state pause active` This command temporarily stops the replication of updates from the database `active` to the database `upgrade`, but it retains any updates made to `active` in the database transaction log files. The updates made to `active` during the upgrade procedure are applied later, when `upgrade` is brought back up. See "Set the replication state of subscribers" in *Oracle TimesTen In-Memory Database Replication Guide* for details.

Step	Upgrade host	Active host
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. For example, call the `ttRepSubscriberWait` built-in procedure. You should expect a value of `<00>` to be returned, indicating there was a clean response, not a time out. (If there is a time out, `ttRepSubscriberWait` returns a value of `01`.) `Command> call ttRepSubscriberWait (,,,,60); < 00 > 1 row found.` See "ttRepSubscriberWait" in the *Oracle TimesTen In-Memory Database Reference* for information.	n/a
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`. See "Starting and stopping the replication agents" in *Oracle TimesTen In-Memory Database Replication Guide* for details.
5.	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 host, use the following `ttMigrate` command: `ttMigrate -c upgrade /backup/ upgrade.dat`	n/a
6.	If the `ttMigrate` command is successful, destroy the database `upgrade`. `ttDestroy upgrade`	Restart the replication agent on the database `active`: `ttAdmin -repStart active`
7.	Create a new installation and a new instance for the new release. See "Creating an installation on Linux/UNIX" and "Creating an instance on Linux/UNIX: Basics" for information.	Resume replication from `active` to `upgrade` by setting the replication state to `start`: `ttRepAdmin -receiver -name upgrade -start start active`

Step	Upgrade host	Active host

8.	Use `ttMigrate` to load the backup created in step 5. into the database `upgrade` for the new release:
	`ttMigrate -r upgrade /backup/upgrade.dat`
	Change the `ramPolicy` to `manual` (The `ramPolicy` is set to `inUse` by default).
	`ttAdmin -ramPolicy manual upgrade`
	`ttAdmin -ramLoad upgrade`
	Note: In this step, you must use the `ttMigrate` utility contained inthe new release of to which you are upgrading.
		n/a

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

10.	Use `ttAdmin` to start the replication agent on the new database `upgrade` and to begin sending updates to the database `active`: `ttAdmin -repStart upgrade`	n/a
11.	Verify that the database `upgrade` is receiving updates from the database `active`. You can verify that updates are sent by applying a recognizable update to a table reserved for that purpose in the database `active`. When the update appears in `upgrade`, you know that replication is operational.	If the applications are still running on the database `active`, let them continue until the database `upgrade` has been successfully migrated and you have verified that the updates are being replicated correctly from `active` to `upgrade`.

Step	Upgrade host	Active host
12.	n/a	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 instance with `active` is ready to be upgraded. **Note**: You may choose to delay upgrading the instance with `active` to the new release until sufficient testing has been performed with the database `upgrade` in the new release. When you are ready to upgrade the instance with the active to the new release, follow the steps in "Online patch upgrade for active master" for details.

Performing an upgrade with active standby pair replication

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

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

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

Online upgrades for an active standby pair with no cache groups

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

- Online patch upgrade for standby master and subscriber
- Online patch upgrade for active master
- Online major upgrade for active standby pair

Also see "Performing an online upgrade with classic replication" for an overview, limitations, and requirements.

Online patch upgrade for standby master and subscriber

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

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

   ```
   ttAdmin -repStop master2
   ```

2. Create a new installation and a new instance for the new release. See "Creating an installation on Linux/UNIX" and "Creating an instance on Linux/UNIX: Basics" for information.

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

   ```
   ttAdmin -repStart master2
   ```

Online patch upgrade for active master

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

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

2. Run the `ttRepSubscriberWait` built-in procedure on the active master database, using the DSN and host of the standby master database. (The result of the call should be `00`. If the value is `01`, you should call ttRepSubscriberWait again until the value `00` is returned.) For example, to ensure that all transactions are replicated to the `master2` standby master on the `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 `master1` active master database:

   ```
   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 (for example, `master2`).

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

7. Upgrade the instance of the former active master database, which is now the standby master database. See "About performing a basic patch upgrade" for details.
8. Restart replication on the database in the upgraded instance, using the `ttRepStart` built-in procedure or the `ttAdmin` utility:

   ```
   ttAdmin -repStart master2
   ```

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

Online major upgrade for active standby pair

When you perform an online upgrade for an active standby pair to a new major release of TimesTen, you must explicitly specify the TCP/IP port for each database. If your active standby pair replication scheme is not configured with a `PORT` attribute for each database, you must use the following steps to prepare for the upgrade. For this procedure, assume there are no cache groups. (Online major upgrades for active standby pairs with cache groups are only supported for read-only cache groups.)

1. Stop the replication agent on every database using the call `ttRepStop` built-in procedure or the `ttAdmin` utility. For example, to stop the replication agent on the `master1` database:

   ```
   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 `master1` database on the `master1host` host and the `master2` database on the `master2host` host and the `subscriber1` database on the `subscriber1host` host:

   ```
   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 `subscriber1` database:

   ```
   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 the *Oracle TimesTen In-Memory Database Replication Guide* for details.

To upgrade the instances of the active standby pair, first upgrade the instance of the standby master. While this node is being upgraded, there is no standby master

database, so updates on the active master database are propagated directly to the subscriber databases. Following the upgrade of the standby node, the active and standby roles are switched and the new standby node is created from the new active node. Finally, the subscriber nodes are upgraded.

1. Instruct the active master database to stop replicating updates to the standby master by executing the `ttRepStateSave` built-in procedure on the active master database. For example, to stop replication to the `master2` standby master database on the `master2host` host:

   ```
   call ttRepStateSave( 'FAILED', 'master2', 'master2host' );
   ```

2. Stop the replication agent on the standby master database using the `ttRepStop` built-in procedure or the `ttAdmin` utility. The following example stops the replication agent for the `master2` standby master database.

   ```
   ttAdmin -repStop master2
   ```

3. Use the `ttMigrate` utility to back up the standby master database to a binary file.

   ```
   ttMigrate -c master2 master2.bak
   ```

   See "ttMigrate" in the *Oracle TimesTen In-Memory Database Reference* for details.

4. Destroy the standby master database, using the `ttDestroy` utility.

   ```
   ttDestroy master2
   ```

5. Create a new installation and a new instance on the `master2host` standby master host. See "Creating an installation on Linux/UNIX" and "Creating an instance on Linux/UNIX: Basics" for information.

6. In the new instance on `master2host`, use `ttMigrate` to restore the standby master database from the binary file created earlier. (This example performs a checkpoint operation after every 20 megabytes of data has been restored.)

   ```
   ttMigrate -r -C 20 master2 master2.bak
   ```

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

   ```
   ttAdmin -repStart master2
   ```

   When the standby master database in the upgraded instance has become synchronized with the active master database, this standby master database moves from the `RECOVERING` state to the `STANDBY` state. The standby master database also starts sending updates to the subscribers. You can determine when the standby master database is in the `STANDBY` state by calling the `ttRepStateGet` built-in procedure.

   ```
   call ttRepStateGet;
   ```

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

9. Execute the `ttRepSubscriberWait` built-in procedure on the active master database, using the DSN and host of the standby master database. (The result of the call should be `00`. If the value is `01`, you should call ttRepSubscriberWait again until the value `00` is returned.) For example, to ensure that all transactions are replicated to the `master2` standby master on the `master2host` host:

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

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

    ```
    ttAdmin -repStop master1
    ```

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

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

12. Instruct the new active master database (`master2`, in our example) to stop replicating updates to what is now the standby master (`master1`) by executing the `ttRepStateSave` built-in procedure on the active master database. For example, to stop replication to the `master1` standby master database on `master1host` host:

    ```
    call ttRepStateSave( 'FAILED', 'master1', 'master1host' );
    ```

13. Destroy the former active master database, using the `ttDestroy` utility.

    ```
    ttDestroy master1
    ```

14. Create the new installation and the instance for the new release on `master1host`. See "Creating an installation on Linux/UNIX" and "Creating an instance on Linux/UNIX: Basics" for information.

15. Create a new standby master database by duplicating the new active master database, using the `ttRepAdmin` utility. For example, to duplicate the `master2` database `master2` on the `master2host` host to the `master1` database, use the following on the host containing the `master1` database:

    ```
    ttRepAdmin -duplicate -from master2 -host master2host -uid pat -pwd patpwd
     -setMasterRepStart master1
    ```

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

    ```
    ttAdmin -repStart master1
    ```

17. 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 `subscriber1` subscriber database:

    ```
    ttAdmin -repStop subscriber1
    ```

18. Destroy the subscriber database using the `ttDestroy` utility.

    ```
    ttDestroy subscriber1
    ```

19. Create a new installation and a new instance for the new release on the subscriber host. See "Creating an installation on Linux/UNIX" and "Creating an instance on Linux/UNIX: Basics" for information.

20. Create the subscriber database by duplicating the new standby master database, using the `ttRepAdmin` utility, as follows.

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

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

    ```
    ttAdmin -repStart subscriber1
    ```

22. Repeat step 17 through step 21 for each other subscriber database.

Online upgrades for an active standby pair with cache groups

This section includes the following topics for online patch upgrades in a scenario with active standby pairs and cache groups:

- Online patch upgrade for standby master and subscriber (cache groups)
- Online patch upgrade for active master (cache groups)
- Online major upgrade for active standby pair (read-only cache groups)

Also see "Performing an online upgrade with classic replication" for an overview, limitations, and requirements.

Online patch upgrade for standby master and subscriber (cache groups)

To perform an online upgrade to a new patch release for the standby master database and subscriber databases, in a configuration with cache groups, complete the following tasks on each database (with exceptions noted).

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

   ```
   ttAdmin -repStop master2
   ```

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

   ```
   ttAdmin -cacheStop master2
   ```

3. Create a new installation and a new instance for the new release. See "Creating an installation on Linux/UNIX" and "Creating an instance on Linux/UNIX: Basics" for information.

4. Restart the cache agent on the standby database using the `ttCacheStart` built-in procedure or the `ttAdmin` utility:

   ```
   ttAdmin -cacheStart master2
   ```

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

   ```
   ttAdmin -repStart master2
   ```

> **Note:**
>
> Steps 2 and 4, stopping and restarting the cache agent, are not applicable for subscriber databases.

Online patch upgrade for active master (cache groups)

To perform an online upgrade to a new patch release for the active master database, in a configuration with cache groups, perform the following steps. You must first reverse the roles of the active and standby master databases, then perform an the upgrade.

1. Pause any applications that are generating updates on the active master database.
2. Stop the cache agent on the current active master database using the `ttCacheStop` built-in procedure or the `ttAdmin` utility:

   ```
   ttAdmin -cacheStop master1
   ```

3. 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 `master2` standby master on the `master2host` host:

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

4. 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 `master1` active master database:

   ```
   ttAdmin -repStop 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, the `master2` database).
8. Upgrade the instance for the former active master database, which is now the standby master database. See "About performing a basic patch upgrade" for details.
9. Restart the cache agent on the post-upgrade database using the `ttCacheStart` built-in procedure or the `ttAdmin` utility:

   ```
   ttAdmin -cacheStart master1
   ```

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

    ```
    ttAdmin -repStart master1
    ```

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

Online major upgrade for active standby pair (read-only cache groups)

Complete the following steps to perform a major upgrade in a scenario with an active standby pair with read-only cache groups. This example upgrades from the `18.1` release to the 22.1

These steps assume that `master1` is the active master database on the `master1host` host and `master2` is the standby master database on the `master2host` host.

> **Note:**
>
> For more information on the built-in procedures and utilities discussed here, see "Built-In Procedures" and "Utilities" in the *Oracle TimesTen In-Memory Database Reference*.

1. On the active master host, run the `ttAdmin` utility to stop the replication agent for the active master database.

   ```
   ttAdmin -repStop master1
   ```

2. On the active master database, use the `DROP ACTIVE STANDBY PAIR` statement to drop the active standby pair. For example, from the `ttIsql` utility:

   ```
   Command> DROP ACTIVE STANDBY PAIR;
   ```

3. On the active master database, use the `CREATE ACTIVE STANDBY PAIR` statement to create a new active standby pair with the cache groups excluded. Ensure that you explicitly specify the TCP/IP port for each database.

   ```
   Command> CREATE ACTIVE STANDBY PAIR master1 ON "master1host",
              master2 ON "master2host"
            STORE master1 ON "master1host" PORT 20000
            STORE master2 ON "master2host" PORT 20010
            EXCLUDE CACHE GROUP cacheuser.readcache;
   ```

   > **Note:**
   >
   > You can use the `cachegroups` command within the `ttIsql` utility to identify all the cache groups defined in the database. In this example, `readcache` is a read-only cache group owned by the `cacheuser` user.

4. On the active master database, call the `ttRepStateSet` built-in procedure to set the replication state for the active master database to `ACTIVE`.

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

   To verify that the replication state for the active master database is set to `ACTIVE`, call the `ttRepStateGet` built-in procedure.

   ```
   Command> call ttRepStateGet();
   < ACTIVE >
   1 row found.
   ```

5. On the active master database, call the `ttRepStart` built-in procedure to start the replication agent.

   ```
   Command> call ttRepStart();
   ```

6. On the standby master host, run the `ttAdmin` utility to stop the replication agent for the standby master database.

   ```
   ttAdmin -repStop master2
   ```

7. On the standby master host, run the `ttAdmin` utility to stop the cache agent for the standby master database.

```
ttAdmin -cacheStop master2
```

8. On the standby master host, run the `ttDestroy` utility to destroy the standby master database. You must either add the `-force` option or first drop all cache groups. After you run the `ttDestroy` utility, run the `cacheCleanUp.sql` script as described below.

   ```
   ttDestroy -force master2
   ```

   Run the *timesten_home*`/install/oraclescripts/cacheCleanUp.sql` SQL*Plus script as the cache administration user to drop the Oracle Database objects. This script takes the host name and the database name (with full path) as parameters. See "Dropping Oracle Database objects used by cache groups with autorefresh" in the *Oracle TimesTen In-Memory Database Cache Guide* for details.

9. Create a new standby master database by duplicating the active master database with the `ttRepAdmin` utility. For example, to duplicate the `master1` database on the `master1host` host of the `master2` database, run the following on the host containing the `master2` database:

   ```
   ttRepAdmin -duplicate -from master1 -host master1host -UID pat -PWD patpwd
     -keepCG -cacheUid cacheuser -cachePwd cachepwd master2
   ```

   > **Note:**
   >
   > You need a user with `ADMIN` privileges defined in the active master database for it to be duplicated. In this example, the `pat` user identified by the `patpwd` password has `ADMIN` privileges.
   >
   > To keep the cache group tables, you need a cache administration user while adding the `-keepCG` option. In this example, the `cacheuser` user identified by the `cachepwd` password is a cache administration user.

10. On the new standby master database, use the `DROP CACHE GROUP` statement to drop all the cache groups.

    ```
    Command> DROP CACHE GROUP cacheuser.readcache;
    ```

11. On the standby master host, run the `ttMigrate` utility to back up the standby master database to a binary file.

    ```
    ttMigrate -c master2 master2.bak
    ```

12. On the standby master host, run the `ttDestroy` utility to destroy the standby master database. After you run the `ttDestroy` utility, run the `cacheCleanUp.sql` script as described below.

    ```
    ttDestroy master2
    ```

    Run the *timesten_home*`/install/oraclescripts/cacheCleanUp.sql` SQL*Plus script as the cache administration user to drop the Oracle Database objects. This script takes the host name and the database name (with full path) as parameters. See "Dropping Oracle Database objects used by cache groups with autorefresh" in the *Oracle TimesTen In-Memory Database Cache Guide* for details.

13. Create a new installation and a new instance for the new release on the standby master host. See "Creating an installation on Linux/UNIX" and "Creating an instance on Linux/UNIX: Basics" for information.

14. In the new instance on the standby master host, run the `ttMigrate` utility to restore the standby master database from the binary file created earlier.

```
ttMigrate -r -C 20 master2 master2.bak
```

> **Note:**
>
> This example performs a checkpoint operation after every 20 MB of data has been restored.

15. On the standby master database, use the `CREATE USER` statement to create a new cache administration user.

```
Command> CREATE USER cacheuser2 IDENTIFIED BY cachepwd;
Command> GRANT CREATE SESSION, CACHE_MANAGER, CREATE ANY TABLE,
         DROP ANY TABLE TO cacheuser2;
```

> **Note:**
>
> You must create the new cache administration user in the Oracle database and grant the user the minimum set of privileges required to perform cache group operations. See "Create users in the Oracle database" in the *Oracle TimesTen In-Memory Database Cache Guide* for information.

16. Connect to the standby master database as the cache administration user, and call the `ttCacheUidPwdSet` built-in procedure to set the new cache administration user name and password. Ensure you specify the cache administration user password for the Oracle database in the `OraclePWD` connection attribute within the connection string.

```
ttIsql "DSN=master2;UID=cacheuser2;PWD=cachepwd;OraclePWD=oracle"
Command> call ttCacheUidPwdSet('cacheuser2','oracle');
```

17. On the standby master database, call the `ttCacheStart` built-in procedure to start the cache agent.

```
Command> call ttCacheStart();
```

18. On the standby master database, call the `ttRepStart` built-in procedure to start the replication agent.

```
Command> call ttRepStart();
```

    The replication state will automatically be set to `STANDBY`. You can call the `ttRepStateGet` built-in procedure to confirm this. (This occurs asynchronously and may take a little time.)

```
Command> call ttRepStateGet();
< STANDBY >
1 row found.
```

19. On the standby master database, use the `CREATE READONLY CACHE GROUP` statement to create all the read-only cache groups.

```
Command> CREATE READONLY CACHE GROUP cacheuser2.readcache
         AUTOREFRESH INTERVAL 10 SECONDS
         FROM oratt.readtbl
           (keyval NUMBER NOT NULL PRIMARY KEY, str VARCHAR(32));
```

> **Note:**
>
> Ensure that the cache administration user has `SELECT` privileges on the cache group tables in the Oracle database. In this example, the `cacheuser2` user has `SELECT` privileges on the `readtbl` table owned by the `oratt` user in the Oracle database. For more information, see "Create the Oracle Database tables to be cached" in the *Oracle TimesTen In-Memory Database Cache Guide*.

20. On the standby master database, use the `LOAD CACHE GROUP` statement to load the data from the Oracle database tables into the TimesTen cache groups.

```
Command> LOAD CACHE GROUP cacheuser2.readcache
         COMMIT EVERY 200 ROWS;
```

21. Pause any applications that are generating updates on the active master database.
22. On the active master database, call the `ttRepSubscriberWait` built-in procedure using the DSN and host of the standby master database. For example, to ensure that all transactions are replicated to the `master2` database on the `master2host` host:

```
Command> call ttRepSubscriberWait(NULL,NULL,'master2','master2host',120);
```

23. On the active master database, call the `ttRepStop` built-in procedure to stop the replication agent.

```
Command> call ttRepStop();
```

24. On the active master database, call the `ttRepDeactivate` built-in procedure to set the replication state for the active master database to `IDLE`.

```
Command> call ttRepDeactivate();
```

25. On the standby master database, call the `ttRepStateSet` built-in procedure to set the replication state for the standby master database to `ACTIVE`. This database and its host become the active master in the active standby pair replication scheme.

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

> **Note:**
>
> In this example, the `master2` database on the `master2host` host just became the active master in the active standby pair replication scheme. Likewise, the `master1` database on the `master1host` host is henceforth considered the standby master in the active standby pair replication scheme.

26. On the new active master database, call the `ttRepStop` built-in procedure to stop the replication agent.

```
Command> call ttRepStop();
```

27. On the active master database, use the `ALTER CACHE GROUP` statement to set the `AUTOREFRESH` mode of all cache groups to `PAUSED`.

```
Command> ALTER CACHE GROUP cacheuser2.readcache
         SET AUTOREFRESH STATE PAUSED;
```

28. On the active master database, use the `DROP ACTIVE STANDBY PAIR` statement to drop the active standby pair.

    ```
    Command> DROP ACTIVE STANDBY PAIR;
    ```

29. On the active master database, use the `CREATE ACTIVE STANDBY PAIR` statement to create a new active standby pair with the cache groups included. Ensure you explicitly specify the TCP/IP port for each database.

    ```
    Command> CREATE ACTIVE STANDBY PAIR master1 ON "master1host",
               master2 ON "master2host"
             STORE master1 ON "master1host" PORT 20000
             STORE master2 ON "master2host" PORT 20010;
    ```

30. On the active master database, call the `ttRepStateSet` built-in procedure to set the replication state for the active master database to `ACTIVE`.

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

31. On the active master database, call the `ttRepStart` built-in procedure to start the replication agent.

    ```
    Command> call ttRepStart();
    ```

32. Resume any applications that were paused in step 21, connecting them to the new active master database.

33. On the new standby master host, run the `ttDestroy` utility to destroy the new standby master database. After you run the `ttDestroy` utility, run the `cacheCleanUp.sql` script as described below.

    ```
    ttDestroy master1
    ```

    Run the *timesten_home*`/install/oraclescripts/cacheCleanUp.sql` SQL*Plus script as the cache administration user to drop the Oracle Database objects. This script takes the host name and the database name (with full path) as parameters. See "Dropping Oracle Database objects used by cache groups with autorefresh" in the *Oracle TimesTen In-Memory Database Cache Guide* for details.

34. Create a new installation and a new instance for the new release on the standby master host. See "Creating an installation on Linux/UNIX" and "Creating an instance on Linux/UNIX: Basics" for information.

35. Create a new standby master database by duplicating the active master database with the `ttRepAdmin` utility. For example, to duplicate the `master2` database on the `master2host` host to the `master1` database, run the following on the host containing the `master1` database:

    ```
    ttRepAdmin -duplicate -from master2 -host master2host -UID pat -PWD patpwd
      -keepCG -cacheUid cacheuser2 -cachePwd cachepwd master1
    ```

36. On the standby master host, run the `ttAdmin` utility to start the cache agent for the standby master database.

    ```
    ttAdmin -cacheStart master1
    ```

37. On the standby master host, run the `ttAdmin` utility to start the cache agent for the standby master database.

    ```
    ttAdmin -repStart master1
    ```

Offline upgrades for an active standby pair with cache groups

Performing a major upgrade in a scenario with an active standby pair with asynchronous writethrough cache groups requires an offline upgrade. This is discussed in the subsection that follows.

Offline major upgrade for active standby pair (cache groups)

Complete the following steps to perform a major upgrade in a scenario with an active standby pair with cache groups. You must perform this upgrade offline. (This example assumes you want to upgrade from release `18.1` to release 22.1

These steps assume `master1` is an active master database on the `master1host` host and master2 is a standby master database on the `master2host` host. (For information about the built-in procedures and utilities discussed, refer to "Built-In Procedures" and "Utilities" in *Oracle TimesTen In-Memory Database Reference*.)

1. Stop any updates to the active database before you upgrade.
2. From `master1`, call the `ttRepSubscriberWait` built-in procedure to ensure that all data updates have been applied to the standby database, where *`numsec`* is the desired wait time.

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

3. From `master2`, call `ttRepSubscriberWait` to ensure that all data updates have been applied to the Oracle database.

   ```
   call ttRepSubscriberWait(null, null, '_ORACLE', null, numsec);
   ```

4. On `master1host`, use the `ttAdmin` utility to stop the replication agent for the active database.

   ```
   ttAdmin -repStop master1
   ```

5. On `master2host`, use `ttAdmin` to stop the replication agent for the standby database.

   ```
   ttAdmin -repStop master2
   ```

6. On `master1host`, call the `ttCacheStop` built-in procedure or use `ttAdmin` to stop the cache agent for the active database.

   ```
   ttAdmin -cacheStop master1
   ```

7. On `master2host`, call `ttCacheStop` or use `ttAdmin` to stop the cache agent for the standby database.

   ```
   ttAdmin -cacheStop master2
   ```

8. On `master1host`, use the `ttMigrate` utility to back up the active database to a binary file.

   ```
   ttMigrate -c master1 master1.bak
   ```

9. On `master1host`, use the `ttDestroy` utility to destroy the active database. You must either use the `-force` option or first drop all cache groups. If you use `-force`, run the script `cacheCleanup.sql` afterward.

   ```
   ttDestroy -force /data_store_path/master1
   ```

The `cacheCleanup.sql` script is a SQL*Plus script, located in the *`installation_dir`*`/oraclescripts` directory (and accessible through *`timesten_home`*`/install/oraclescripts`), that you run after connecting to the Oracle database as the cache user. It takes as parameters the host name and the database name (with full path). For information, refer to "Dropping Oracle Database objects used by autorefresh cache groups" in the *Oracle TimesTen In-Memory Database Cache Guide*.

10. Create a new installation and a new instance for the new major release on `master1host`. See "Creating an installation on Linux/UNIX" and "Creating an instance on Linux/UNIX: Basics" for information.
11. Create a new database in 22.1.*w.x* using `ttIsql` with DSN connection attribute setting `AutoCreate=1`. In this new database, create a cache user. The following example is a sequence of commands to execute in `ttIsql` to create this cache user and give it appropriate access privileges.

    The cache user requires `ADMIN` privilege to execute the next step, `ttMigrate -r`. Once migration is complete, you can revoke the `ADMIN` privilege from this user if desired.

    ```
    Command> CREATE USER cacheuser IDENTIFIED BY cachepassword;
    Command> GRANT CREATE SESSION, CACHE_MANAGER, CREATE ANY TABLE,
             DROP ANY TABLE TO cacheuser;
    Command> GRANT ADMIN TO cacheuser;
    ```

12. In the new instance on `master1host`, use the `ttMigrate` utility as the cache user to restore `master1` from the binary file created earlier. (This example performs a checkpoint operation after every 20 megabytes of data has been restored, and assumes the password is the same in the Oracle database as in TimesTen.)

    ```
    ttMigrate -r -cacheuid cacheuser -cachepwd cachepassword -C 20 -connstr
     "DSN=master1;uid=cacheuser;pwd=cachepassword;oraclepwd=cachepassword"
     master1.bak
    ```

13. On `master1host`, use `ttAdmin` to start the replication agent.

    ```
    ttAdmin -repStart master1
    ```

    > **Note:**
    >
    > This step also sets the database to the active state. You can then call the `ttRepStateGet` built-in procedure (which takes no parameters) to confirm the state.

14. On `master1host`, call the `ttCacheStart` built-in procedure or use `ttAdmin` to start the cache agent.

    ```
    ttAdmin -cacheStart master1
    ```

    Then you can use the `ttStatus` utility to confirm the replication and cache agents have started.

15. Put each automatic refresh cache group into the `AUTOREFRESH PAUSED` state. This example uses `ttIsql`:

    ```
    Command> ALTER CACHE GROUP mycachegroup SET AUTOREFRESH STATE paused;
    ```

16. From `master1`, reload each cache group, specifying the name of the cache group and how often to commit during the operation. This example uses `ttIsql`:

```
Command> LOAD CACHE GROUP cachegroupname COMMIT EVERY n ROWS;
```

You can optionally specify parallel loading as well. See the "LOAD CACHE GROUP" SQL statement in the *Oracle TimesTen In-Memory Database SQL Reference* for details.

17. On `master2host`, use `ttDestroy` to destroy the standby database. You must either use the `-force` option or first drop all cache groups. If you use `-force`, run the script `cacheCleanup.sql` afterward (as discussed earlier).

    ```
    ttDestroy -force /data_store_path/master2
    ```

18. Create the new installation and the new instance for the new major release on `master2host`. See "Creating an installation on Linux/UNIX" and "Creating an instance on Linux/UNIX: Basics" for information.

19. In the new instance on `master2host`, use the `ttRepAdmin` utility with the `-duplicate` option to create a duplicate of active database `master1` to use as standby database `master2`. Specify the appropriate administrative user on `master1`, the cache manager user and password, and to keep cache groups.

    ```
    ttRepAdmin -duplicate -from master1 -host master1host -uid pat -pwd patpwd
    -cacheUid orcluser -cachePwd orclpwd -keepCG master2
    ```

20. On `master2host`, use `ttAdmin` to start the replication agent. (You could optionally have used the `ttRepAdmin` option `-setMasterRepStart` in the previous step instead.)

    ```
    ttAdmin -repStart master2
    ```

21. On `master2`, the replication state will automatically be set to `STANDBY`. You can call the `ttRepStateGet` built-in procedure to confirm this. (This occurs asynchronously and may take a little time.)

    ```
    call ttRepStateGet();
    ```

22. On `master2host`, call the `ttCacheStart` built-in procedure or use `ttAdmin` to start the cache agent.

    ```
    ttAdmin -cacheStart master2
    ```

    After this, you can use the `ttStatus` utility to confirm the replication and cache agents have started.

If you want to create read-only subscriber databases, on each subscriber host you can create the subscriber by using the `ttRepAdmin` utility `-duplicate` option to duplicate the standby database. The following example creates `subscriber1`, using the same `ADMIN` user as above and the `-nokeepCG` option to convert the cache tables to normal TimesTen tables, as appropriate for a read-only subscriber.

```
ttRepAdmin -duplicate -from master2 -host master2host -nokeepCG
-uid pat -pwd patpwd subscriber1
```

For related information, refer to "Rolling out a disaster recovery subscriber" in the *Oracle TimesTen In-Memory Database Replication Guide*.

Performing an offline TimesTen upgrade when using Oracle Clusterware

This section discusses the steps for an offline upgrade of TimesTen when using TimesTen with Oracle Clusterware. You have the option of also upgrading Oracle Clusterware, independently, while upgrading TimesTen. (See "Performing an online TimesTen upgrade when using Oracle Clusterware" for details on online upgrade.)

> **Note:**
>
> - These instructions apply for either a TimesTen patch upgrade (for example, from `22.1.w.x` to `22.1.y.z`) or a TimesTen major upgrade (for example, from `18.1` to `22.1`).
> - Refer to the *Oracle TimesTen In-Memory Database Release Notes* for information about versions of Oracle Clusterware that are supported by TimesTen.

For this procedure, except where noted, you can execute the `ttCWAdmin` commands from any host in the cluster. Each command affects all hosts.

1. Stop the replication agents on the databases in the active standby pair:

   ```
   ttCWAdmin -stop -dsn advancedDSN
   ```

2. Drop the active standby pair:

   ```
   ttCWAdmin -drop -dsn advancedDSN
   ```

3. Stop the TimesTen cluster agent. This removes the hosts from the cluster and stops the TimesTen daemon:

   ```
   ttCWAdmin -shutdown
   ```

4. Upgrade TimesTen on the desired hosts.
   - To perform a TimesTen patch upgrade, each node in the cluster must have TimesTen from the same major release.
   - To perform a TimesTen major upgrade, you must use `ttMigrate`. See "Moving to a different major release using ttMigrate" for details.
5. Upgrade Oracle Clusterware if desired. See "Oracle Clusterware" in the Oracle Database documentation for information.
6. If you have upgraded Oracle Clusterware, use the `ttInstanceModify` utility to configure TimesTen with Oracle Clusterware. On each host, run:

   ```
   ttInstanceModify -crs
   ```

   For Linux or UNIX hosts, see "Change the Oracle Clusterware configuration for an instance" for details.
7. Start the TimesTen cluster agent. This includes the hosts defined in the cluster as specified in `ttcrsagent.options`. This also starts the TimesTen daemon.

```
ttCWAdmin -init
```

8. Create the active standby pair replication scheme:

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

Important: The host from which you run this command must have access to the `cluster.oracle.ini` file. (See "Configuring Oracle Clusterware management with the cluster.oracle.ini file" in the *Oracle TimesTen In-Memory Database Replication Guide* for information about this file.)

9. Start the active standby pair replication scheme:

```
ttCWAdmin -start -dsn advancedDSN
```

Performing an online TimesTen upgrade when using Oracle Clusterware

This section discusses how to perform an online rolling upgrade (patch) for TimesTen, from TimesTen `22.1.`*w.x* to `22.1.`*y.z*, in a configuration where Oracle Clusterware manages active standby pairs. (See "Performing an offline TimesTen upgrade when using Oracle Clusterware" for an offline upgrade.)

The following topics are covered:

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

> **Note:**
>
> - Refer to *Oracle TimesTen In-Memory Database Release Notes* for supported versions of Oracle Clusterware.

Supported configurations

The following basic configurations are supported for online rolling upgrades for TimesTen. In all cases, Oracle Clusterware manages the hosts.

- One active standby pair on two hosts.
- Multiple active standby pairs with one database on each host.
- Multiple active standby pairs with one or more database on each host.

(Other scenarios, such as with additional spare hosts, 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 upgrade does not change the TimesTen environment for the active and standby databases.
- These instructions are for TimesTen patch upgrades only. Online major upgrades are not supported in configurations where Oracle Clusterware manages active standby pairs.
- There are at least two hosts managed by Oracle Clusterware.

  Multiple active or standby databases managed by Oracle Clusterware can exist on a host only if there are at least two hosts in the cluster.

> **Note:**
>
> Upgrade Oracle Clusterware if desired, but *not* concurrently with an online TimesTen upgrade. When performing an online TimesTen patch upgrade in configurations where Oracle Clusterware manages active standby pairs, you must perform the Clusterware upgrade independently and separately, either before or after the TimesTen upgrade.

> **Note:**
>
> For information about Oracle Clusterware, see "Oracle Clusterware" in the Oracle Database documentation.

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 upgrade for the standby database
- Start the standby database
- Switch the roles of the active and standby databases
- Shut down the new standby database
- Perform an 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 instance name is `upgrade2`, and the instance administrator is `terry`.

Verify that the active standby pair is operating properly

Complete these steps to confirm that the active standby pair is operating properly.

1. Verify the following.
   - The active and the standby databases run a TimesTen 22.1.*w.x* release.
   - The active and standby databases are on separate hosts managed by Oracle Clusterware.
   - Replication is working.
   - If the active standby pair replication scheme includes cache groups, the following are true:
     - AWT and SWT writes are working from the standby database in TimesTen to the Oracle database.
     - Refreshes are working from the Oracle database to the active database in TimesTen.
2. Run 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. Run 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. Run 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. Run an Oracle Clusterware command similar to the following to obtain the names of the Oracle Clusterware Master, Daemon, and Agent processes on the host of

the standby database. It is suggested to filter the output by using the `grep TT` command:

```
crsctl status resource -n standbyHostName | grep TT
```

2. Run 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 instance, and the Agent process for the instance.

```
crsctl stop resource TT_Master_upgrade2_terry_myDSN_1
crsctl stop resource TT_Daemon_upgrade2_terry_host2
crsctl stop resource TT_Agent_upgrade2_terry_host2
```

3. Stop the TimesTen main daemon.

```
ttDaemonAdmin -stop
```

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

Perform an upgrade for the standby database

Complete these steps for an offline upgrade of the instance for the standby database.

1. Create a new installation. See "Creating an installation on Linux/UNIX" for information.
2. Point the instance to the new installation. See "Associate an instance with a different installation (upgrade or downgrade)" for details.
3. Configure the new installation for Oracle Clusterware.

Start the standby database

Complete these steps to start the standby database.

1. Run the following `ttCWAdmin` command to start the TimesTen main daemon, the TimesTen Oracle Clusterware agent process and the TimesTen Oracle Clusterware Daemon process:

```
ttCWAdmin -init -hosts localhost
```

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

```
crsctl start resource TT_Master_upgrade2_terry_MYDSN_1
```

Switch the roles of the active and standby databases

Use the `ttCWAdmin -switch` command to switch the roles of the active and standby databases to enable the offline upgrade on the other master database.

```
ttCWAdmin -switch -dsn myDSN
```

Use the `ttCWAdmin -status` command to verify that the switch operation has completed before starting the next task.

Shut down the new standby database

Use the Oracle Clusterware `crsctl status resource` command to obtain the names of the Master, Daemon, and Agent processes on the host of the new standby database. This example assumes the host `host1` and filters the output through `grep TT`:

```
crsctl status resource -n host1 | grep TT
```

Run commands such as those in "Shut down the standby database" and use the appropriate instance name, instance administrator, DSN, and host name. For example:

```
crsctl stop resource TT_Master_upgrade2_terry_MYDSN_0
crsctl stop resource TT_Daemon_upgrade2_terry_host1
crsctl stop resource TT_Agent_upgrade2_terry_host1
ttDaemonAdmin -stop
```

Perform an upgrade of the new standby database

See "Perform an upgrade for the standby database" for the steps.

Start the new standby database

See "Start the standby database" and use the Master process name obtained by the `crsctl status resource` command from "Shut down the new standby database" as outlined above.

```
ttCWAdmin -init -hosts localhost
crsctl start resource TT_Master_upgrade2_terry_MYDSN_0
```

Upgrades for multiple active standby pairs on many pairs of hosts

The process to upgrade the instances for multiple active standby pairs on multiple pairs of hosts is essentially the same as the process to upgrade the instances for a single active standby pair on two hosts. See "Upgrade tasks for one active standby pair" for details. The best practice is to perform the upgrades for the active standby pairs one at a time.

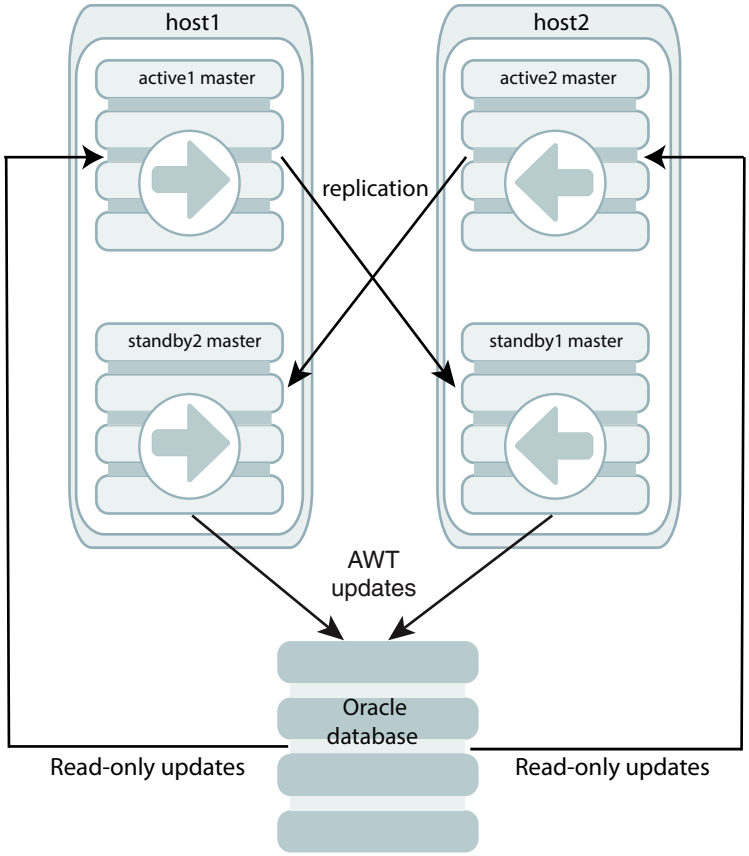
Use the `ttCWAdmin -status` command to determine the state of the databases managed by Oracle Clusterware.

Upgrades for multiple active standby pairs on a pair of hosts

Multiple active standby pairs can be on multiple pairs of hosts. See "Upgrades for multiple active standby pairs on many pairs of hosts" for details. Alternatively, multiple active standby pairs can be on a single pair of hosts. One scenario is for all the active databases to be on one host and all the standby databases to be on the other. A more typical scenario, to better balance the workload, is for each host to have some active databases and some standby databases.

Figure 7-1 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 7-1 Multiple active standby pairs on two hosts



This configuration can result in greater write throughput for cache groups and more balanced resource usage. See the next section, "Sample configuration files: multiple active standby pairs on one pair of hosts", for sample `sys.odbc.ini` entries and a sample `cluster.oracle.ini` file for this kind of configuration. (See "Configuring Oracle Clusterware management with the cluster.oracle.ini file" in the *Oracle TimesTen In-Memory Database Replication Guide* for information about that file.)

The rolling upgrade process for multiple active standby pairs on a single pair of hosts is similar in nature to the process of upgrading multiple active standby pairs on multiple pairs of hosts. See "Upgrades for multiple active standby pairs on many pairs of hosts" for details.

First, however, if the active and standby databases are mixed between the two hosts, switch all standby databases to one host and all active databases to the other host. Use the `ttCWAdmin -switch -dsn` *`DSN`* command to switch active and standby databases between hosts. Once all the active databases are on one host and all the standby databases are on the other host, follow the steps below to perform the upgrade for the entire "standby" host.

Be aware that upgrades affect the entire instance and associated databases on one host.

1. Verify that the standby databases run on the desired host. Use the `ttCWAdmin -status -dsn` *`DSN`* command and the `ttCWAdmin -status` command.
2. Modify the Oracle Clusterware `stop` commands to stop all Master processes on the host where all the standby databases reside.

3. Modify the Oracle Clusterware `start` commands to start all Master processes on the host where all the standby databases reside.

The following subsections contain related samples.

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

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

The following are sample `sys.odbc.ini` entries:

```
[databasea]
Driver=timesten_home/install/lib/libtten.so
DataStore=/scratch/terry/ds/databasea
PermSize=400
TempSize=320
DatabaseCharacterSet=WE8MSWIN1252
OracleNetServiceName=ORCL

[databaseb]
Driver=timesten_home/install/lib/libtten.so
DataStore=/scratch/terry/ds/databaseb
PermSize=400
TempSize=320
DatabaseCharacterSet=WE8MSWIN1252
OracleNetServiceName=ORCL

[databasec]
Driver=timesten_home/install/lib/libtten.so
DataStore=/scratch/terry/ds/databasec
PermSize=400
TempSize=320
DatabaseCharacterSet=WE8MSWIN1252
OracleNetServiceName=ORCL

[databased]
Driver=timesten_home/install/lib/libtten.so
DataStore=/scratch/terry/ds/databased
PermSize=400
TempSize=320
DatabaseCharacterSet=WE8MSWIN1252
OracleNetServiceName=ORCL
```

The following is a sample `cluster.oracle.ini` file:

```
[databasea]
MasterHosts=host1,host2
CacheConnect=Y

[databaseb]
MasterHosts=host2,host1
CacheConnect=Y

[databasec]
MasterHosts=host2,host1
CacheConnect=Y

[databased]
```

```
MasterHosts=host1,host2
CacheConnect=Y
```

The `cluster.oracle.ini` file places one active database and one standby database on each host. This is accomplished by reversing the order of the host names specified for the `MasterHost` attribute.

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

Run an Oracle Clusterware command similar to the following to obtain the names of the Oracle Clusterware Master, Daemon and Agent processes on the host of the standby database. It is suggested to filter the output by using the `grep TT`:

```
crsctl status resource -n standbyHostName | grep TT
```

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

```
crsctl stop resource TT_Master_upgrade2_terry_DATABASEA_0
crsctl stop resource TT_Master_upgrade2_terry_DATABASEB_1
crsctl stop resource TT_Daemon_upgrade2_terry_HOST2
crsctl stop resource TT_Agent_upgrade2_terry_HOST2
ttDaemonAdmin -stop
```

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

```
ttCWAdmin -init -hosts localhost
crs start resource TT_Master_upgrade2_terry_DATABASEA_0
crs start resource TT_Master_upgrade2_terry_DATABASEB_1
```

Upgrades when using parallel replication

You can perform an online or offline upgrade from a database that has not enabled parallel replication to a database that has enabled automatic parallel replication (with or without disabled commit dependencies). See "ReplicationApplyOrdering" attribute, in the *Oracle TimesTen In-Memory Database Reference* for information on setting automatic parallel replication values.

The remainder of this section discusses additional considerations along with scenarios where an offline upgrade is required.

Considerations regarding parallel replication

Be aware of the following considerations when upgrading hosts that use parallel replication:

- Consider an active standby pair without parallel replication enabled. To upgrade the instances to a `22.1` release and use automatic parallel replication (default value of `0` for the `ReplicationApplyOrdering` attribute), use the appropriate procedure for an active standby pair upgrade. See "Performing an upgrade with active standby pair replication" for details.
- Consider an active standby pair with no cache groups and automatic parallel replication enabled (value of `0` for the `ReplicationApplyOrdering` attribute). To upgrade the instances to a `22.1` release to use automatic parallel replication with disabled commit

dependencies (value of `2` for the `ReplicationApplyOrdering` attribute), use the procedure for an active standby pair online major upgrade. See "Online major upgrade for active standby pair" for details. The value for the `ReplicationApplyOrdering` attribute must be changed from `0` to `2` before restoring any of the databases. For example:

```
ttMigrate -r
"DSN=master2;ReplicationApplyOrdering=2;ReplicationParallelism=2;
  LogBufParallelism=4" master2.bak
```

> **Note:**
>
> You may upgrade a database with a replication scheme with `ReplicationApplyOrdering=2` to a database with `ReplicationApplyOrdering=0` by using the same active standby pair online major upgrade procedure.
>
> Automatic parallel replication with disabled commit dependencies supports only asynchronous active standby pairs with no cache groups. For more information, see "Configuring parallel replication" in the *Oracle TimesTen In-Memory Database Replication Guide*.

- You cannot replicate between databases that have the `ReplicationParallelism` attribute set to greater than `1` but have different values for the `ReplicationApplyOrdering` attribute.

Scenarios that require an offline upgrade

You must use an offline upgrade for these scenarios:

- Moving from an automatic parallel replication environment to another automatic parallel replication environment with a different number of tracks, as indicated by the value of the `ReplicationParallelism` attribute.
- Moving between major releases (for example, from `18.1` to `22.1`) and using asynchronous writethrough cache groups.
- Moving from regular replication with asynchronous writethrough in `18.1` to automatic parallel replication with asynchronous writethrough in `22.1`.

Use the procedure described in "Moving to a different major release using ttMigrate" for offline upgrades. Alternatively, you can upgrade one side and use the `ttRepAdmin -duplicate -recreate` command to create the new database.

Performing an upgrade of your client instance

You can upgrade your client instance which is being used to access a database in a full instance. For information on instances, see "Overview of installations and instances" and "TimesTen instances" for details. For information on Client/Server, see "Overview of the TimesTen Client/Server" in the *Oracle TimesTen In-Memory Database Operations Guide*.

To perform the upgrade, follow these steps:

1. Optional: This step is included for informational purposes to assist you in identifying and verifying the TimesTen client release information.

   In the client instance, run the `ttVersion` utility to verify the client release and the client instance. In this example, running `ttVersion` in the client instance shows the client release is `22.1.1.10.0` and the client instance is `instance_221_client`.

   ```
   % ttVersion
   TimesTen Release 22.1.1.10.0  (64 bit Linux/x86_64) (instance_221_client)
   2023-06-29T23:22:07Z
     Instance home directory: /scratch/instance_221_client
     Group owner: g900
   ```

2. Optional: This step is included for informational purposes to establish and then show a client connection to the `database1` database. In the client instance, run `ttIsqlCS` to connect to the `database1` database in the full instance (on the server). Note that the `TCP_PORT` is not specified. The default value is assumed.

   ```
   % ttIsqlCS -connstr "TTC_SERVER=server.mycompany.com;TTC_SERVER_DSN=database1";

   Copyright (c) 1996, 2023, Oracle and/or its affiliates. All rights reserved.
   Type ? or "help" for help, type "exit" to quit ttIsql.



   connect "TTC_SERVER=server.mycompany.com;TTC_SERVER_DSN=database1";
   Connection successful: DSN=;TTC_SERVER=server.mycompany.com;
   TTC_SERVER_DSN=database1;
   ...
   (Default setting AutoCommit=1)
   ```

3. Stop all applications using the client instance. In this example, in the client instance, first run `ttIsqlCS` to connect to the `database1` database, then exit from `ttIsqlCS`.

   ```
   % ttIsqlCS -connstr "TTC_SERVER=server.mycompany.com;
   TTC_SERVER_DSN=database1";

   Copyright (c) 1996, 2021, Oracle and/or its affiliates. All rights reserved.
   Type ? or "help" for help, type "exit" to quit ttIsql.



   connect "TTC_SERVER=server.mycompany.com;TTC_SERVER_DSN=database1";
   Connection successful: DSN=;TTC_SERVER=server.mycompany.com;
   TTC_SERVER_DSN=database1;
   ...
   (Default setting AutoCommit=1)
   Command> exit
   Disconnecting...
   Done.
   ```

4. Create a new client installation in a new location. For example, create the `clientinstall_new` installation directory. Then unzip the new release zip file into that directory. For example, to create the `22.1.1.11.0` installation on Linux 64-bit, unzip `timesten2211110.server.linux8664.zip` into the `clientinstall_new` directory. (Note, there is only one distribution on Linux 64-bit. This distribution contains the server and the client installation.)

   ```
   % mkdir clientinstall_new
   % cd clientinstall_new
   % unzip /swdir/TimesTen/ttinstallers/timesten2211110.server.linux8664.zip
   [...UNZIP OUTPUT...]
   ```

See "TimesTen installations" for detailed information.

5. Modify the client instance to point to the new installation. Do this by running the `ttInstanceModify` utility with the `-install` option from the `$TIMESTEN_HOME/bin` directory of the client instance.

   In this example, point the client instance to the installation in `/clientinstall_new/tt22.1.1.11.0`.

```
% $TIMESTEN_HOME/bin/ttInstanceModify -install
 /clientinstall_new/tt22.1.1.11.0

Instance Info (UPDATED)
-----------------------

Name:           instance_221_client
Version:        22.1.1.11.0
Location:       /scratch/instance__client
Installation:   /clientinstall_new/tt22.1.1.11.0

* Client-Only Installation


The instance instance_221_client now points to the installation in
clientinstall_new/tt22.1.1.10.0
```

6. Optional: In the client instance, run the `ttVersion` utility to verify the client release is `22.1.1.11.0`.

```
% ttVersion
TimesTen Release 22.1.1.11.0 (64 bit Linux/x86_64) (instance_221_client)
2021-06-28T22:37:51Z
  Instance home directory: /scratch/instance_221_client
  Group owner: g900
```

7. Restart the applications that use the client instance.

   In this example, in the client instance, run `ttIsqlCS` to connect to the `database1` database in the full instance.

```
% ttIsqlCS -connstr "TTC_SERVER=server.mycompany.com;
TTC_SERVER_DSN=database1";

Copyright (c) 1996, 2021, Oracle and/or its affiliates. All rights reserved.
Type ? or "help" for help, type "exit" to quit ttIsql.



connect "TTC_SERVER=server.mycompany.com;TTC_SERVER_DSN=database1";
Connection successful: DSN=;TTC_SERVER=server.mycompany.com;
TTC_SERVER_DSN=database1;
...
(Default setting AutoCommit=1)
```

8. Optional: Delete the previous release installation (used for the client).

```
% chmod -R 750 installation_dir/tt22.1.1.11.0
% rm -rf installation_dir/tt22.1.1.11.0
```

Index

H

I

J

L

M

N

O

P

R

S

T

U

V

W