Oracle® TimesTen In-Memory Database

Installation Guide

Release 11.2.1

E13063-02

August 2009

Oracle TimesTen In-Memory Database Installation Guide, Release 11.2.1

E13063-02

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

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

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

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

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

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

Contents

2 Data Store Upgrades

Preface

Oracle TimesTen In-Memory Database is a high-performance, in-memory data manager that supports the ODBC (Open DataBase Connectivity) and JDBC (Java DataBase Connectivity) interfaces.

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

Related documents

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

`http://www.oracle.com/technology/documentation/timesten_doc.html`

Conventions

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

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 variable in a code example that you must replace. For example: `Driver=`*`install_dir`*`/lib/libtten.sl` Replace *`install_dir`* with the path of your TimesTen installation directory.
[ ]	Square brackets indicate that an item in a command line is optional.
{ }	Curly braces indicated that you must choose one of the items separated by a vertical bar (|) in a command line.
|	A vertical bar (or pipe) that separates alternative arguments.

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

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

Convention	Meaning
install_dir	The path that represents the directory where the current release of TimesTen is installed.
TTinstance	The instance name for your specific installation of TimesTen. Each installation of TimesTen must be identified at install time with a unique alphanumeric instance name. This name appears in the install path.
bits or *bb*	Two digits, either 32 or 64, that represent either the 32-bit or 64-bit operating system.
release or *rr*	Two digits that represent the first two digits of the current TimesTen release number, with or without a dot. For example, 70 or 7.0 represents TimesTen Release 7.0.
jdk_version	Two digits that represent the version number of the major JDK release. Specifically, 5 represents JDK 5.
DSN	The data source name.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at http://www.oracle.com/accessibility/.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Deaf/Hard of Hearing Access to Oracle Support Services

To reach Oracle Support Services, use a telecommunications relay service (TRS) to call Oracle Support at 1.800.223.1711. An Oracle Support Services engineer will handle technical issues and provide customer support according to the Oracle service request process. Information about TRS is available at `http://www.fcc.gov/cgb/consumerfacts/trs.html`, and a list of phone numbers is available at `http://www.fcc.gov/cgb/dro/trsphonebk.html`.

Technical support

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

`http://www.oracle.com/support/contact.html`

What's New

This section summarizes the new features of Oracle TimesTen In-Memory Database release 11.2.1 that are documented in this guide and provides links to more information.

Installation user

In this release, TimesTen recommends that the instance not be installed by user `root`. The user installing TimesTen must be a member of the TimesTen administrators group and is referred to as the TimesTen instance administrator.

Access Control

TimesTen is now installed with Access Control enabled. It cannot be disabled. All users must be created within TimesTen and granted the appropriate privileges. For information on types of users, how to create users and how to grant the appropriate privileges, see the "Managing Access Control" chapter in the *Oracle TimesTen In-Memory Database Operations Guide*.

Platform support

Platform support has changed in this release. For a list of platforms on which TimesTen is supported in 11.2.1, see Platform support in Chapter 1, "TimesTen Installation."

1 TimesTen Installation

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

- Platforms and configurations
- Installation instances
- Choosing the appropriate TimesTen components
- Installation prerequisites
- Default installation directories
- Pre-Install requirements for operating system group and file permissions

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

- Installing TimesTen on UNIX systems
- Installing TimesTen on Windows systems

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

- ODBC installation
- Environment variables
- TimesTen Quick Start
- Viewing the online documentation
- TimesTen on HP-UX Memory Windows

Finally, this chapter contains information that helps you troubleshoot problems that may arise during the installation process:

- Installation problems

Platforms and configurations

This section includes these topics:

- Platform support
- JDK support
- Client/Server configurations
- Oracle In-Memory Database Cache

- Replication configurations

Platform support

TimesTen Data Manager and TimesTen Client/Server are supported in these environments:

Environment	32-bit	64-bit
Microsoft Windows 2000, Windows XP, Windows Vista and Windows Server 2003 and 2008 for Intel IA-32 and EM64T and AMD64 CPUs.	Yes	Yes
Asianux 2.0 and 3.0 for Intel IA-32 and EM64T and AMD64 CPUs.	Yes	Yes
Solaris 9 and 10 for UltraSparc CPUs.	Yes	Yes
Solaris 10 for AMD64 CPUs.		Yes
SuSE LINUX Enterprise Server 10 for Intel IA-32 and EM64T and AMD64 CPUs.	Yes	Yes
SuSE LINUX Enterprise Server 10 for Itanium2 CPUs		Yes
Red Hat Enterprise Linux 4 and 5 for Intel Itanium2 CPUs.		Yes
Red Hat Enterprise Linux 4 and 5 for Intel IA-32 and EM64T and AMD64 CPUs.	Yes	Yes
Oracle Enterprise Linux 4 and 5 for Intel IA-32 and EM64T and AMD64 CPUs.	Yes	Yes
MontaVista Linux Carrier Grade Edition Release 4.0 and 5.0 for Intel IA-32, EM64T and AMD64 CPUs.	Yes	Yes
HP-UX 11*i* v2 and 11*i*v3 for PA-RISC	Yes	Yes
HP-UX 11*i* v2 and 11*i*v3 for Itanium2.	Yes	Yes
AIX 5L 5.3 and 6.1 for POWER CPUs	Yes	Yes

TimesTen is supported on Oracle VM guest x86 and x86-64 operating systems on Oracle Enterprise Linux 4 and 5 or Red Hat Enterprise Linux 4 and 5 systems.

JDK support

TimesTen supports the following JDKs on the specified platforms:

Environment	JDK 5.0	JDK 6.0
Microsoft Windows 2000, Windows XP, Windows Vista and Windows Server 2003 and 2008 for Intel IA-32 and EM64T and AMD64 CPUs	Sun and Oracle JRockit	Sun and Oracle JRockit
Asianux 2.0 and 3.0 for Intel IA-32 and EM64T and AMD64 CPUs.	Sun and Oracle JRockit	Sun and Oracle JRockit
Solaris 9 and 10 for UltraSparc CPUs	Sun and Oracle JRockit 64-bit	Sun
Solaris 10 for EM64T and AMD64 CPUs	Sun	Sun
SuSE LINUX Enterprise Server 10 for Intel IA-32 and EM64T and AMD64 CPUs	Sun and Oracle JRockit	Sun and Oracle JRockit

Environment	JDK 5.0	JDK 6.0
SuSE LINUX Enterprise Server 10 for Itanium2 CPUs	Oracle JRockit	
Red Hat Enterprise Linux 4 and 5 for Intel Itanium2 processors	Oracle JRockit	
Red Hat Enterprise Linux 4 and 5 for Intel IA-32	Sun and Oracle JRockit	Sun and Oracle JRockit
Red Hat Enterprise Linux 4 and 5 for EM64T and AMD64 CPUs	Sun and Oracle JRockit	Sun and Oracle JRockit
Oracle Enterprise Linux 4 and 5 for Intel IA-32 CPUs.	Sun and Oracle JRockit	Sun and Oracle JRockit
Oracle Enterprise Linux 4 and 5 for EM64T and AMD64 CPUs.	Sun and Oracle JRockit	Sun and Oracle JRockit
MontaVista Linux Carrier Grade Edition Release 4.0 and 5.0 for Intel IA-32 and EM64T CPUs	Sun and Oracle JRockit	Sun and Oracle JRockit
HP-UX 11i v2 and 11iv3 for PA-RISC 32-bit and 64-bit	HP	HP
HP-UX 11i v2 and 11iv3 for Itanium2	HP	HP
AIX 5L 5.3 and 6.1 for POWER CPUs	IBM	IBM

Client/Server configurations

A TimesTen client on any supported platform can connect to a TimesTen server on any platform where TimesTen is supported.

A TimesTen 6.0 client can connect to a 6.0 TimesTen Server of any patch level. If the `-insecure-backwards-compat` option is set in the `ttendaemon.options` file, a TimesTen 6.0 client can connect to a TimesTen 11.2.1 server.

A TimesTen 7.0 or newer client can connect to a TimesTen 6.0 or newer server, under certain configurations.

For configuration details see "Configuring TimesTen Client and Server" in the *Oracle TimesTen In-Memory Database Operations Guide*.

Oracle In-Memory Database Cache

Oracle In-Memory Database Cache (IMDB Cache) allows you to cache Oracle database data in TimesTen. The TimesTen installation includes Oracle Instant Client, and the following Oracle server releases are supported with this option:

- Oracle Database 10*g* Release 2 (Oracle 10.2.0.4.0 or above)
- Oracle Database 11*g* Release 1 on Windows (32-bit), Linux x86, Linux x86-64, Solaris SPARC (64-bit), AIX (PPC64), HP-UX PA-RISC (64-bit) and HP-UX Itanium

Also, see "Setting up the Oracle and TimesTen systems" in *Oracle In-Memory Database Cache User's Guide*.

Platform support

IMDB Cache is supported on the 32-bit and 64-bit platforms specified in this table:

Environment	32-bit	64-bit
Microsoft Windows 2000, Windows XP, Windows Vista and Windows Server 2003 and 2008 for Intel IA-32 and EM64T and AMD64 CPUs	Yes	Yes
Asianux 2.0 and 3.0 for Intel IA-32 and EM64T and AMD64 CPUs.	Yes	Yes
Solaris 9 and 10 for UltraSparc CPUs	Yes	Yes
Solaris 10 for AMD64 CPUs systems		Yes
SuSE LINUX Enterprise Server 10 for Intel IA-32, EM64T and AMD64 CPUs	Yes	Yes
SuSE LINUX Enterprise Server 10 for Itanium2 CPUs		Yes
Oracle Enterprise Linux 4 and 5 for Intel IA-32 CPUs.	Yes	Yes
Oracle Enterprise Linux 4 and 5 for EM64T and AMD64 CPUs.		Yes
Red Hat Enterprise Linux 4 and 5 running on Intel Itanium2 processors		Yes
Red Hat Enterprise Linux 4 and 5 for Intel IA-32 and EM64T and AMD64 CPUs	Yes	Yes
HP-UX 11i v2 and 11iv3 for PA-RISC 32-bit and 64-bit	Yes	Yes
HP-UX 11i v2 and 11iv3 for Itanium2	Yes	Yes
AIX 5L 5.3 and 6.1 for POWER CPUs	Yes	Yes

Replication configurations

TimesTen-to-TimesTen replication is supported only between identical platforms and bit-levels.

Installation instances

On UNIX, you can install more than one instance of TimesTen. On Windows, you can install only one instance of any major and minor TimesTen release, where the major release numbers are the first 3 numbers and the minor release numbers are those after the major numbers. For the 11.2.1.1.0 release, the major release is 11.2.1 and the minor release is 1.0. For Windows, for example, you can install both 11.2.1.0.0 and 11.2.2.0.0 releases on the same Windows machine. However, you cannot install both 11.2.1.0.0 and 11.2.1.1.0, since they have the same major release of 11.2.1.

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

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

- Instance naming
- Instance port numbers

Instance naming

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

On Windows, the TimesTen installation scripts do not prompt you to supply an instance name. The instance name on Windows is `tt1121_32` on 32-bit systems and `tt1121_64` on 64-bit systems.

On UNIX system, by default, the instance name for this release is `tt1121_32` on 32-bit systems and `tt1121_64` on 64-bit systems. The default location is the TimesTen directory in the home directory of the user installing TimesTen. The instance name is case-insensitive, must be at least one alphanumeric character and up to 255 characters. The name can include underscores (_) or periods (.), but no other special characters.

If you would like to install a second instance of the same TimesTen release, you must supply a unique instance name and port number. The TimesTen installation script can detect if an instance of the particular release of TimesTen already exists on the machine and prompts you for a new instance name and port number for the main TimesTen daemon.

Instance port numbers

Any time that you install more than one instance of TimesTen on the same machine, specify a unique TCP/IP port number for each TimesTen daemon during the install.

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

On UNIX systems, the default port on which the TimesTen main daemon listens is 53384 for 32-bit installations and 53388 for 64-bit applications.

On UNIX systems, the default port on which the TimesTen Server daemon listens is 53385 for 32-bit installations and 53389 for 64-bit applications.

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

Choosing the appropriate TimesTen components

TimesTen allows you to select the components of TimesTen that you wish to install.

- Components available on UNIX
- Components available on Windows

Components available on UNIX

On UNIX, you can install the following components. In addition, the installation script prompts you to install the TimesTen Quick Start and documentation.

Type	Description
TimesTen Client	Installs the TimesTen Client only. No other TimesTen components are installed on the machine. Use this installation to allow the TimesTen Client to access the TimesTen Server from another installation.
TimesTen Data Manager	Installs the TimesTen Data Manager only. Use this installation to run the TimesTen Data Manager locally.

Type	Description
TimesTen Client, Server and Data Manager	Installs the TimesTen Data Manager, Client and Server. Use this installation to perform the following: ■ Allow a Client from another installation to access the TimesTen Server. ■ Allow the TimesTen Client to access the either this TimesTen Server installation or another TimesTen Server installation. ■ Allow applications to access the TimesTen Data Manager locally.

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

Components available on Windows

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

Type	Description
TimesTen Data Manager	Installs the TimesTen Data Manager only. Use this installation to run the TimesTen Data Manager locally.
TimesTen Data Manager Debug Libraries	Installs the TimesTen Data Manager debug libraries. Used particularly during the development phase to allow you to debug problems that may occur. By default, the debug libraries are not installed.
TimesTen Server	Installs the TimesTen Data Server. Use this installation to: ■ Allow a Client on another machine to access the TimesTen Server on this machine. ■ Allow the TimesTen Clients on other machines to access the TimesTen Server on this machine.
TimesTen Client	Installs the TimesTen Client only. No other TimesTen components are installed on the machine. Use this installation to allow the TimesTen Client to access the TimesTen Server on a remote machine.
TimesTen Quick Start	Installs the TimesTen Quick Start, which includes demos. By default, the QuickStart and demos are not installed.
TimesTen Documentation	Installs the TimesTen Documentation Library. By default, the documentation is not installed.

Installation prerequisites

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

On both UNIX and Windows platforms where JDBC is supported you must have the appropriate version of the JDK installed on your machine to use JDBC. See "JDK support" on page 1-2 to learn which JDK is required for your platform.

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

- Windows requirements

General UNIX requirements

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

- The number of semaphores
- Allowable shared memory

In addition, you may need to perform the following:

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

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

- Filesystem options
- Semaphores
- Java
- Other Client/Server settings

Filesystem options

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

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

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

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

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

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

Using `vxtunefs` online option requires Advanced VxFS.

Semaphores

TimesTen consumes 1 SEMMNI per active data store, plus 1 additional SEMMNI per TimesTen instance where Client/Server communication is done through shared memory. For each active data store, TimesTen consumes 100 SEMMSL if the Connections attribute is set to the default value, and one additional SEMMSL for each connection above the default.

Java

If you are running JDBC, install the latest JDK and any vendor required patches. Refer to the web site of the JDK provider for the patches you may need.

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

Other Client/Server settings

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

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

```
set rlim_fd_max = 1080
```

in `/etc/system`.

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

AIX prerequisites

On AIX, before installation, modify replication to improve the performance of TimesTen on your system. The replication agent requests TCP send and receive buffers of a minimum size of 512KB. To ensure this is granted please ensure the kernel parameter `sb_max` is set to a minimum of 512KB. The value may be changed using the following command:

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

To query the value, use the following command:

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

HP-UX prerequisites

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

- Semaphores
- Shared memory
- Large data stores
- Replication

Semaphores

On HP-UX systems, to connect to more than 2 data stores simultaneously, you must increase the value of the kernel parameter semmns.

To view existing kernel parameter settings, log in as user `root`.

For HP-UX 11*i*v2, use the command:

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

Shared memory

On HP-UX systems, you also must increase the value of the parameter shmmax. To make these changes, log in as user root and use the kmtune command, kctune commands or run the **HP System Administration Manager** to see existing kernel parameter settings.

To use the **HP System Administration Manager**, perform the following:

1. Execute the **HP System Administration Manager**, as follows:

   ```
   # /usr/sbin/sam
   ```

2. Double-click **Kernel Configuration**, then double-click **Configurable Parameters**.
3. Scroll down the list of parameters to semmns and change its value to a minimum of 4096 or greater.
4. For HP-UX 11*i* systems, also scroll down the list of parameters to shmmax and change its value to a maximum of 0x40000000.

   Note: For 32-bit systems, the value 0x40000000 (a 4 followed by seven zeroes) indicates that the largest shared memory segment that can be created is 1024 MB. The size of the shared memory segment required for a shared data store is larger than the requested data store size. Set this value high enough to support the largest shared memory segment needed.

5. Recompile the kernel. Choose **Create a New Kernel** from the Actions menu.
6. Reboot the system.

Large data stores

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

To enable large files, create the file systems using the newfs command with the -o largefiles option or alter the file systems using the fsadm command with the -o largefiles option. The following fsadm command alters the file system to enable large files:

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

For example:

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

Replication

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

For HP-UX 11.23 (11iv2)

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

```
# /usr/bin/ndd -set /dev/tcp tcp_recv_hiwater_max 524288
```

Linux prerequisites

For Linux, TimesTen has been tested with Asianux 2.0 and 3.0, Red Hat Enterprise Linux 4 and 5, the MontaVista Linux Carrier Grade Edition Release 4.0 and 5.0 and SuSE LINUX Enterprise Server 10 minimal configurations. The C development tools are required if you intend to do native development on the machine.

> **Note:** TimesTen does not support SELinux. When installing Linux for use with TimesTen, make sure that the SELinux option is disabled.

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

- Large pages
- Semaphores
- Shared memory
- IPC Client/Server
- Replication
- IMDB Cache

Large pages

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

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

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

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

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

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

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

```
cat /proc/meminfo
```

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

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

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

If PAM (Pluggable Authentication Modules) is enabled, you may need to change the `/etc/security/limits.conf` file.

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

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

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

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

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

Semaphores

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

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

On Linux systems, the first parameter of kernel.sem must be a minimum of 128. We recommend that you add the line:

```
kernel.sem = 250 32000 100 128
```

to the `/etc/sysctl.conf` file and either reboot or run the command:

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

Shared memory

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

```
kernel.shmmax=2147483648
```

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

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

To increase the shared memory size without rebooting, use:

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

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

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

You can also change this value by the following command:

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

This command has the same effect as the sysctl command.

IPC Client/Server

On Red Hat Linux systems, to enable more than 6 ShmIpc Client/Server connections, add the line:

```
kernel.sem = 250 32000 128 100
```

to the /etc/sysctl.conf file and either reboot or run the command:

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

Replication

For replication, TCP send and receive buffers should be increased to *a minimum* of 512KB. To increase the buffers to 4 MB, add the lines:

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

to the /etc/sysctl.conf file and either reboot or run the command:

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

IMDB Cache

For IMDB Cache, TCP send and receive buffers should be increased to even greater values. To make these changes, add the lines:

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

to

to the /etc/sysctl.conf file and either reboot or run the command:

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

Solaris prerequisites

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

Filesystem options

In addition to the filesystem options listed in the section "General UNIX requirements" on page 1-7, on Solaris UFS file systems, if you plan to have TimesTen applications that use DurableCommits=1, mount the file system with the `-forcedirectio` option.

IPC semaphores

On Solaris 9, TimesTen checks the IPC configuration at install time. If either the IPC Semaphores module or the IPC Shared Memory module is not installed, you can install them by hand. Use the commands:

```
ryps3# modload /kernel/sys/semsys
ryps3# modload /kernel/sys/shmsys
```

Increase number of semaphores

For Solaris 10 systems, the default semaphore settings should be sufficient without entries in `/etc/system`.

On other Solaris systems, you may need to increase the number of semaphores. TimesTen consumes 1 SEMMNI per active data store, plus one additional SEMMNI per TimesTen instance where Client/Server communication is done through shared memory.

For each data store, TimesTen consumes 100 SEMMSL if the Connections attribute is set to the default value (64), and one additional SEMMSL for each estimated connection above the default. We recommend that you increase the number of semaphores:

1. Log in as user `root`.
2. Set or add the following lines to `/etc/system`:

   ```
   set semsys:seminfo_semmni = 20
   set semsys:seminfo_semmsl = 512
   set semsys:seminfo_semmns = 2000
   set semsys:seminfo_semmnu = 2000
   ```

   > **Note:** The values in this step are the minimum number of required semaphores. You can increase these numbers as needed. You can use the following formula as a guide, although in practice, SEMMNS and SEMMNU can be much less than SEMMNI * SEMMSL because not every program in the system needs semaphores.
   >
   > SEMMNS=SEMMNU = (SEMMNI * SEMMSL).

3. Reboot your system.
4. To view the current limits, use:

   ```
   % /usr/sbin/sysdef
   ```

   This command displays the limits for SEMMSL, SEMMNS, SEMOPM, and SEMMNI. SEMOPM is the maximum number of operations per `semop` call. It does not need to be modified.

Shared memory IPC client connections

For Solaris systems prior to Solaris 10, to have more than 6 ShmIpc-enabled Client DSN connections per process, you must make changes to the SHMSEG kernel parameter. For example, to allow a single process to access 12 data stores, add the following line to `/etc/system` and reboot before using TimesTen:

```
set shmsys:shminfo_shmseg=12
```

Other changes

Other changes that you may need to make to your Solaris system include the following:

- To allow a large number of connections to a data store, add the following lines to `/etc/system` and reboot before using TimesTen:

  ```
  set rlim_fd_cur=4096
  set rlim_fd_max=4096
  ```

- To set shared memory on Solaris 10 systems, specify `project.max-shm-memory`.
- To enable large shared memory objects in Solaris, add the following line to `/etc/system` and reboot before using TimesTen:

  ```
  set shmsys:shminfo_shmmax = 0x40000000
  ```

  Note: The value `0x40000000` (a 4 followed by seven zeroes) indicates that the largest shared memory segment that can be created is 1024 MB. The size of the shared memory segment required for a data store is larger than the data store size permanent size. Set this value high enough to support the largest shared memory segment needed.

Large data stores

If you keep data stores on a Solaris UFS file system, and are using transaction-consistent checkpoints, you may need to change the settings of some kernel parameters to get the best performance for your checkpoints. The Solaris UFS Throttle algorithm causes processes that write a single large file to be put to sleep when a byte count threshold exceeds the high-water mark. To disable the algorithm, add the line:

```
set ufs:ufs_WRITES = 0
```

to the `/etc/system` file.

Alternatively, you can increase the high-water mark by adding the line:

```
set ufs:ufs_HW = desired value
```

to the `/etc/system` file.

You must reboot the system for the new value to take effect.

Setting the high-water mark to the size of the checkpoint file should provide satisfactory performance, although a lower value may as well. More information on the UFS Throttle algorithm may be obtained in the white paper, "Understanding Solaris Filesystems and Paging" (SMLI TR-98-55) available from `http://www.sun.com`.

Replication

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

```
# /usr/sbin/ndd -set /dev/tcp tcp xmit_hiwat=524288
# /usr/sbin/ndd -set /dev/tcp tcp_recv_hiwat=524288
```

Windows requirements

To work properly, the TimesTen debug libraries depend on Visual Studio 2003, 2005 or 2008. If you intend to use debug libraries, make sure that you have first installed Visual Studio 2003, 2005 or 2008.

On Windows Vista, you must have Administrator privileges to perform certain operations, such as starting and stopping the TimesTen daemon. If User Account Control is enabled, and you are logged in as the local Administrator, then you can successfully run these operations in the usual way. However, if you are logged in as "a member of the Administrator group," then you must explicitly invoke these tasks with Windows Administrator privileges.

To start a command prompt window with Windows Administrator privileges:

1. On your Windows Vista Desktop, create a shortcut for the command prompt window. An icon for that shortcut appears on the Desktop.
2. Right click the icon for the newly created shortcut, and specify "Run as administrator."

When you open this window, the title bar reads Administrator: Command Prompt. Commands run from within this window are run with Administrator privileges.

Default installation directories

The TimesTen default installation directories for release 11.2.1 are as follows:

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

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

The temporary directory is operating system-dependent. Usually it is located in these directories:

- On Windows, `C:\Documents and Settings\%USERNAME%\Local Settings\Temp`
- On Solaris and Linux, `/tmp`
- On HP-UX and AIX, `/var/tmp`

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

Note: On Windows, the complete temporary directory path must be less than 190 characters for the installation to complete successfully. In addition, TimesTen does not support file path names that contain multibyte characters. Please make sure that the installation path, data store path, transaction log path, and temporary file path do not contain any multibyte characters.

Pre-Install requirements for operating system group and file permissions

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

- TimesTen instance administrators and TimesTen users groups
- Directory and file permissions
- Creating UNIX TimesTen administrator and users groups

TimesTen instance administrators and TimesTen users groups

For security, we restrict access to the TimesTen installation to members of a single operating system group, under which TimesTen is installed. We refer to this group as the *TimesTen users group*. Only users that are members of the TimesTen users group are allowed to perform direct driver connections to TimesTen and perform operations on TimesTen data stores. Any users connecting to a TimesTen data store through a client connection do not need to be members of the TimesTen users group. For a description of ways to connect to TimesTen, see "TimesTen connection options" in the *Oracle In-Memory Database Cache Introduction*.

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

- On Windows, the TimesTen users group and the TimesTen administrators group are the same operating system group. TimesTen is always installed under the Adminstrators operating system group. Therefore, the instance administrator on a Windows installation must be a member of the Administrators group to install TimesTen. In addition, all users who perform a direct driver connection must also be a member of the Administrators group.
- On UNIX, the TimesTen instance administrators group and the TimesTen users group can be the same or different operating system groups:
  - TimesTen instance administrator group. Any user installing TimesTen must be a member of this group. This group must be granted read and write access to `/etc/TimesTen`, which contains information about all TimesTen instances installed on the machine.
  - TimesTen users group. The instance administrator must also be a member of this group to install TimesTen. After installation, only members of this operating system group are allowed to make direct driver connections to TimesTen and perform operations on TimesTen data stores.

  The details on how to create both operating system groups on UNIX are included in "Creating UNIX TimesTen administrator and users groups" on page 1-17.

Directory and file permissions

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

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

Permissions and instance registry pre-requisites for TimesTen

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

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

For more information on operating system groups, see "TimesTen instance administrators and TimesTen users groups" on page 1-16.

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

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

The details on how to set the directory permissions for `/etc/TimesTen` to the instance administrators group are included in "Creating UNIX TimesTen administrator and users groups" on page 1-17.

Creating UNIX TimesTen administrator and users groups

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

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

Create the TimesTen users group

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

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

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

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

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

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

3. Add the user who is installing and any users who are administering TimesTen to the TimesTen users group that you just created.
4. Provide the name of this group, if not the same as the default TimesTen users group, during the installation at the appropriate time.

The directory and file permissions for the TimesTen installation have the group specified as the group you defined during the installation. This sets the permissions to restrict read and write access for most directories, files, checkpoint files, transaction log files, shared memory segments, and semaphores to this defined group. There are exceptions for certain resources as determined by TimesTen. See "Directory and file permissions" on page 1-17 for more information on permissions.

When installing on HP-UX systems, the operating system user running the TimesTen daemon must belong to an operating system group that has been given the MLOCK privilege, if you want to use the MemoryLock feature of TimesTen.

For example, if the user is a member of a group called timesten, then the following command (run as root) gives the timesten group the MLOCK privilege:

```
# setprivgrp timesten MLOCK
```

The getprivgrp command can be used to check the privileges of a group:

```
$ getprivgrp timesten
timesten: MLOCK
```

> **Note:** On Linux systems, root privileges are required to use the MemoryLock attribute. On Solaris systems, you must be installed as root to use MemoryLock=1 or 2. Data stores in a non-root instance of TimesTen can use settings 3 and 4 for this attribute on Solaris systems.

Create the TimesTen instance registry and administrators group

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

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

1. Login as root.
2. If the directory /etc/TimesTen does not already exist, create it.

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

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

3. Create an operating system group for the TimesTen instance administrators group. You can name this group as you wish. For our examples, we use the name `ttadmin`.

4. Assign ownership permissions on the `/etc/TimesTen` directory to the TimesTen instance administrators group so that only the instance administrator may access and execute. The `instance_info` file contained in the `/etc/TimesTen` directory must be readable and writable by the instance administrators group.

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

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

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

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

Installing TimesTen on UNIX systems

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

> **Note:** Before beginning installation, be sure that the prerequisites defined in "Installation prerequisites" on page 1-6 have been met.

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

- Installing TimesTen
- Working with the TimesTen daemon and server on UNIX systems
- Informational messages on UNIX systems
- Changing the daemon port number on UNIX
- Uninstalling TimesTen on UNIX systems

Installing TimesTen

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

1. Download TimesTen to your system. The download consists of a gzipped tar file that is named `timesten`*`release.platform`*`.tar.gz`, for example, `timesten112110.linux86.tar.gz`

2. Log in as the TimesTen instance administrator and copy the gzip file to the location from which you wish to install.

3. Unzip the installation file:

   ```
   % gunzip timestenrelease.platform.tar.gz
   ```

4. Extract the TimesTen files:

   ```
   % tar -xf timestenrelease.platform.tar
   ```

5. Change to the platform directory:

```
% cd platform
```

6. Run the TimesTen setup script:

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

You can run the setup script with the option `-install` or `-uninstall` (default is `-install`). When you use the `-uninstall` option, the script stops the daemon and Server if they are running and removes all files it had installed.

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

The `setup.sh` script takes these options:

Option	Description
`-install`	Installs TimesTen.
`-uninstall`	Uninstalls TimesTen
`-batch` *`filename`*	Installs or uninstalls TimesTen without having to respond to prompts. If *`filename`* is specified, the installation reads all installation prompts from the file. The batch file *`filename`* is optional. However, TimesTen recommends that you create the batch file and specifically indicate the instance name of the installation. If no batch file is provided or if the batch file does not contain an instance name, TimesTen installs a default instance, using `tt1121_`*`bits`* for the instance name. If an instance with the same name already exists on the installation machine, the install procedure fails.
`-help`	Displays the help message.
`-installDoc`	Installs the Quick Start and the TimesTen documentation.
`-record` *`filename`*	Installs or uninstalls TimesTen and records responses to prompts described in filename. The file can then be used as the parameter to the `-batch` option.
`-verbose`	Displays extra installation information.

The installation contains tar files of TimesTen components. If the setup script cannot find the tar files to extract from, it prompts you for their location.

7. Enter your response to the setup script prompts.

> **Note:** To install or uninstall TimesTen without having to respond to prompts, use the `-batch` flag with the `setup.sh` script. Batch files from older releases of TimesTen cannot be used to install this release. All new prompts in the installation script for this release are assigned default answers and may produce unexpected results when batch files from different releases are used.

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

- Prompts you to:
  - Install a new instance
  - Upgrade an existing instance. (This option allows you to upgrade from a release previous to the 11.2.1 release.)
  - Display information about an existing instance
  - Quit the installation.
- Prompts you to choose the default instance name or choose an instance name for your TimesTen instance.

  > **Note:** Each TimesTen installation is identified by a unique instance name. The instance name must at least one alphanumeric string and no longer than 255 characters.

- Prompts you to install one of the following components:
  - Client/ Server and Data Manager
  - Data Manager only
  - Client only
- Prompts you for the location of your TimesTen instance. By default installs the instance in `$HOME/TimesTen`. The TimesTen documentation refers to the installation directory as *`install_dir`*.
- Prompts you for the location of the TimesTen daemon home directory.
- Prompts you for the location of TimesTen daemon log files. The default is *`install_dir`*`/info`.
- Prompts you to specify the daemon port number. The default port number is 53384 for 32-bit installations and 53388 for 64-bit applications.

  > **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 the `ttVersion` utility.

- Prompts you to set the TimesTen users group or choose world accessibility. For more information on these options, see "Pre-Install requirements for operating system group and file permissions" on page 1-16 for details on the TimesTen users group and file permissions. You can:
  1. Restrict access to group *`default group`*
  2. Restrict access to a different group
  3. Make the TimesTen instance world accessible (not recommended). Choose this option only if all users on this machine are trusted and you wish to disable all operating system-level access control for this installation.
- Prompts you to determine if PL/SQL should be enabled for the instance. Default answer is "yes." If not enabled at install time, PL/SQL can be enabled for the instance at a later time using the `ttmodinstall` utility.

> **Note:** Enabling PL/SQL increases the size of some TimesTen libraries.

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

  You can leave this field blank. If you do not specify the value of the `TNS_ADMIN` environment variable at install time, you can set it at a later time with the `ttmodinstall` utility. However, before using the In-Memory Database Cache, you must set this environment variable.
- Prompts you to specify the server port number. The default port number is 53385 for 32-bit installations and 53389 for 64-bit applications. Installs the client and server components.
- Prompts you to install Quick Start and the TimesTen documentation. The TimesTen Quickstart applications can take up to 64 Mbytes of disk space. The default directory is *`install_dir`*`/quickstart` and *`install_dir`*`/doc`
- Prompts for the location of where to install the demo data store. This indicates that when you install the QuickStart, the TimesTen demo database files are installed in the `DemoDataStore` directory that defaults to the *`install_dir`*`/info/DemoDataStore` location.
- Prompts you to indicate if you want to install TimesTen replication with Oracle Clusterware. Prompts you for the path into which to install the Oracle Clusterware installation on this machine and the port number for the TimesTen Clusterware agent.
- The install checks for any nodes where the Oracle Clusterware is currently configured and prompts you to specify a node list for TimesTen replication with Oracle Clusterware.
- Removes any previous installation of this release of TimesTen if you are installing an upgrade.
- Installs the TimesTen components into the appropriate directories.
- Installs the client components.
- Starts the daemon.

8. If you want the TimesTen instance to start each time the machine is rebooted, log in as user root, and run the `setuproot` script as root. The `setuproot` script is located in the *`install_dir`*`/bin` directory:

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

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

Working with the TimesTen daemon and server on UNIX systems

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

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

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

To stop the daemon manually, use the utility command:

```
% ttDaemonAdmin -stop
```

To start the daemon manually, use the utility command:

```
% ttDaemonAdmin -start
```

Informational messages on UNIX systems

As the TimesTen daemon operates, it generates error, warning, informational and debug messages for TimesTen system administration and for debugging applications. At installation time, you determine whether these messages go into a file or to the `syslog` facility.

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

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

```
-facility name
```

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

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

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

`subsystem-facility.severity-level`

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

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

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

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

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

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

Changing the daemon port number on UNIX

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

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

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

```
% ttmodinstall -port 12345
```

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

Uninstalling TimesTen on UNIX systems

To uninstall all TimesTen components, follow these steps:

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

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

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

Installing TimesTen on Windows systems

This section discusses installation and related issues for Windows systems.

For a list of Windows platforms supported by TimesTen, see "Platforms and configurations" on page 1-1.

On Windows 64-bit systems, TimesTen 32-bit and 64-bit instances cannot co-exist on the same machine.

> **Note:** Before beginning installation, ensure that the prerequisites defined in "Installation prerequisites" on page 1-6 have been met.

Installing TimesTen

An InstallShield program installs your TimesTen instance on Windows systems. To install TimesTen manually, run the `setup.exe` command:

> **Note:** Each time you execute `SETUP.EXE`, the install program checks for previous installations. If a previous release of TimesTen exists, the installer returns an error message asking you to use the Add or Remove Programs control panel to uninstall the previous release of TimesTen. To install a new release of TimesTen where the major and minor release numbers (e.g. 11.2.1.0.0 and 11.2.1.1.0) match, you must first uninstall the previous release of TimesTen and then run `SETUP.EXE` again.

The TimesTen installation script performs these actions:

- Prompts you for the location of the installation. By default, TimesTen is installed as `C:\TimesTen\tt1121_`*`bits`*.
- Prompts you to select the components that you would like to install:
  - TimesTen Data Manager
  - TimesTen Data Manager Debug Libraries
  - TimesTen Server
  - TimesTen Client
  - Optional Components
    - Timesten Quick Start
    - TimesTen Documentation

  For more information, see "Components available on Windows" on page 1-6.
- Prompts for the location to install the demo data store. This indicates that when you install the QuickStart, the TimesTen demo database files are installed in the `DemoDataStore` directory that defaults to the `C:\Documents and Settings\`*`username`*`\Application Data\TimesTen\`.
- Prompts you to set the location to be supplied for the `TNS_ADMIN` environment variable that specifies the directory where the `tnsnames.ora` file can be found.

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

- Prompts you to select the Program Folder for the Start Menu. Browse to choose the folder that you want for this installation either from existing folders or a new folder. The default is Timesten 11.2.1 (*bits*).
- Asks if you want permissions on this installation to be readable and writable by anyone who has access to the machine. This is not recommended. If disabled, permissions are restricted to users who are members of the Administrators group. See "Pre-Install requirements for operating system group and file permissions" on page 1-16 for details on permissions and world accessibility. Choose this option only if you wish to disable all operating system-level access control for this installation.
- Prompts you to register environment variables. If selected, the installation program adds TimesTen directories to the system environment variables LIB and INCLUDE and sets other appropriate variables. If you decide not to register the environment variables at installation time, you can set the environment variables at any time after installation on a per session basis by running the script `*install_dir*\bin\ttenv.bat`. The `ttenv` script is described in Setting environment variables for TimesTen.
- Prompts you to select the JDK version, if any, to add to the CLASSPATH variable.
- Displays your installation selections before continuing to install TimesTen.
- Prompts you to display the release notes and launch the Quick Start Guide. For information on the Quick Start, see "TimesTen Quick Start" on page 1-34.

Note: TimesTen cannot be installed in a mapped network drive. Attempting to install TimesTen in a mapped network drive results in an error.

Installing TimesTen in silent mode

TimesTen allows you to save installation options to a batch file that you can later use to install TimesTen without having to answer each option in a dialog box. To set up silent mode:

- From a command-line, run:

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

  With this option, TimesTen walks you through a normal setup operation. TimesTen saves your responses to the file `C:\WINDOWS\setup.iss`.

You can now use this file to run an installation in silent mode:

- From a command-line, run: `setup.exe -s -f1 *response_file*`. For example:

  ```
  C:> setup.exe -s -f1C:\WINDOWS\setup.iss
  ```

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

Note: Batch files from releases older than TimesTen Release 11.2.1 should not be used to install this release. All new prompts in the installation script for this release are assigned default answers and may produce unexpected results when batch files from different releases are used.

Verifying installation

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

1. Check that the TimesTen 11.2.1 Start menu shortcut has been added to the Windows Desktop **Start > All Programs** menu.
2. On the Windows Desktop, choose **Start > Settings > Control Panel > Administrative Tools > Data Sources (ODBC)**. This opens the ODBC Data Source Administrator.
3. Click **Drivers**. Check to see that the correct drivers are installed. You should see the **TimesTen Data Manager** 11.2.1 driver. If you installed TimesTen Client, you should see the **TimesTen Client 11.2.1** driver. Click **OK**.
4. On the Windows Desktop, choose **Start > Settings > Control Panel > Administrative Tools > Services** and check that the TimesTen Data Manager 11.2.1 service has the word "Started" in the Status field. At this time, you can also set Recovery options to attempt to restart the service after a failure.

These steps verify that the system has been installed properly.

Verifying TimesTen Client and Server installation

To verify that the Client and Server have been properly installed:

Note: The instructions in this section are valid if you are installing 32-bit TimesTen on 32-bit Windows or 64-bit TimesTen on 64-bit Windows. However, if you are installing 32-bit TimesTen on 64-bit Windows, verify the TimesTen ODBC entries by executing the following:

```
%windir%\SysWOW64\odbcad32.exe
```

1. On the Windows Desktop, choose **Start > Control Panel > Administrative Tools > Data Sources (ODBC)**. This opens the ODBC Data Source Administrator.
2. Click **System DSN**.
3. Select the `sampledb_1121` sample data source and click **Configure**.

   Note: The `sampledb_1121` DSN is used for client applications that use TCP/IP communications with the TimesTen Server. The `sampledbCS_1121` DSN is used for Client/Server connections.

   This opens the TimesTen Client Data Source Setup dialog.
4. Click **Test TimesTen Server Connection** to attempt a connection to the server.

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

   - ODBC, Windows sockets, and the TimesTen Client are installed on the machine.
   - The TimesTen Server you have selected is defined.
   - The host machine for the TimesTen Server is running.

- The TimesTen Server is running.

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

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

   - The data source you have chosen is defined on the server.
   - The TimesTen Client can connect to the data source.

Working with the Data Manager Service and the Server on Windows

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

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

To change the startup mode:

1. On the Windows desktop, choose **Start > Settings > Control Panel > Administrative Tools** >Services. This displays all currently available services.
2. Select **TimesTen Data Manager 11.2.1**.
3. Choose either **Manual** or **Automatic** from the Startup type list. Click **OK**.

If the TimesTen Data Manager startup mode is Manual, follow these instructions to start and stop the service:

1. On the Windows desktop, choose **Start > Settings > Control Panel > Administrative Tools** >Services. This displays all currently available services.
2. Select **TimesTen Data Manager 11.2.1**.
3. Click **Start** to start the service. If the service is already running, click **Stop** to stop the service.

> **Note:** TimesTen writes events into the Event Log file. The Windows Application Event Log can get full. To avoid filling the Application Event Log, check the log settings in the Event Viewer. You can change the size of the Event Log or control whether it overwrites old events.

Informational messages on Windows systems

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

Uninstalling TimesTen on Windows systems

To uninstall TimesTen for Windows:

- On the Windows Desktop, choose **Start > Control Panel > Add/Remove Programs**.

To verify that removal was successful, check that:

- The TimesTen 11.2.1 Start menu shortcut has been removed from the Start > Programs menu.
- The TimesTen Data Manager 11.2.1 has been removed from the Services list.
- The TimesTen 11.2.1 drivers have been removed from the ODBC Drivers tab in the ODBC Control Panel.

Note: DSNs created by TimesTen installation are removed upon TimesTen uninstall. DSNs created by users are not removed during TimesTen uninstall.

ODBC installation

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

The ODBC SDK redistributable components are installed in `C:\WINDOWS\SYSTEM32` on Windows systems. Microsoft only permits TimesTen to redistribute portions of the ODBC SDK; those portions are installed automatically (if they are not already present). Other components-Microsoft sample programs, online help files, and C language header files-are available separately from Microsoft as part of the Microsoft ODBC SDK, which can be installed separately as required. Additionally, the ODBC C language header files and ODBC online help are bundled as part of Microsoft Visual Studio .NET 2003 or Microsoft Visual Studio 2005 or 2008. Most TimesTen developers do not need to install the SDK separately.

On UNIX systems, no separate SDK installation is required.

Environment variables

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

Environment variable	What to include	For settings and other information, see:
LIB, LIBPATH, LD_LIBRARY_PATH or SHLIB_PATH	On UNIX systems, include the `lib` directory under the TimesTen installation directory	"Shared library path environment variable" on page 1-32

Environment variable	What to include	For settings and other information, see:
NLS_LANG	If NLS_LANG is set to as NA, an OCI connection error or the ORA-12705 message is thrown.	On Windows, if an older version of Oracle has been installed, such as Oracle9*i*, the registry key HKEY_LOCAL_MACHINE\Software\ORACLE\NLS_LANG may be set to an invalid value, such as NA. If this value is NA, the TimesTen installer replaces the value with AMERICAN_AMERICA.US7ASCII. This ensures that TimesTen OCI, Pro*C, and IMDB Cache can connect properly. TimesTen uses the Oracle Instant Client to make these connections. The Oracle Instant Client requires this value to be a valid NLS_LANG setting.
ODBCINI	The location where the odbc.ini file used by TimesTen data stores is to be found.	"ODBCINI environment variable" on page 1-31
PATH	Include the bin directory under the TimesTen installation directory. On Windows, also include the path to the Oracle installation if you are using the IMDB Cache option.	"PATH environment variable" on page 1-31, "Shared library path environment variable" on page 1-32 and "Installing TimesTen on Windows systems" on page 1-24
SYSODBCINI	Set to the location where the sys.odbc.ini file used by TimesTen system data stores is to be found. This environment variable should be set in the start-up script.	"SYSODBCINI environment variable" on page 1-32
SYSTTCONNECTINI	Set to the location where the sys.ttconnect.ini file used by TimesTen Client applications to define logical server names.	"SYSTTCONNECTINI environment variable" on page 1-32
TMP or TMPDIR	Set to the location of the temporary directory. TimesTen uses this directory during recovery and other operations.	"Default installation directories" on page 1-15
TNS_ADMIN	If using IMDB Cache, set to the location of the tnsnames.ora file. Required if you are using the IMDB Cache option.	"TNS_ADMIN environment variable" on page 1-32
Java	For Java applications, there are certain environment variables that must be set.	"Java environment variables" on page 1-32

The following sections describe environment variables in TimesTen:

- Setting environment variables for TimesTen
- PATH environment variable
- ODBCINI environment variable
- SYSODBCINI environment variable

- SYSTTCONNECTINI environment variable
- TNS_ADMIN environment variable
- Shared library path environment variable
- Java environment variables

Setting environment variables for TimesTen

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

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

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

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

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

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

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

or

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

PATH environment variable

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

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

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

ODBCINI environment variable

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

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

SYSODBCINI environment variable

TimesTen applications use the sys.odbc.ini file to define system data sources and their data store attributes. (For a description of data store attributes, see "Data Store Attributes" in the *Oracle TimesTen In-Memory Database Reference*.) A system data source can be used by any user on the machine. On Windows, system DSNs are defined from the **System DSN** tab of the ODBC Data Source Administrator. On UNIX, system DSNs are defined in the file *install_dir*/info/sys.odbc.ini. To override the name and location of this file at run-time, set the $SYSODBCINI environment variable to the pathname of a sys.odbc.ini file before launching the TimesTen application.

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

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

SYSTTCONNECTINI environment variable

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

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

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

TNS_ADMIN environment variable

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

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

Shared library path environment variable

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

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

On HP-UX 32-bit systems, add *install_dir*/lib to the SHLIB_PATH environment variable.

On HP-UX 64-bit systems, add *install_dir*/lib to the LD_LIBRARY_PATH environment variable.

Java environment variables

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

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

Set the CLASSPATH variable

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

```
install_dir/ttjdbcjdk_ver.jar
```

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

> **Note:** If more than one jar file is listed in the CLASSPATH, make sure the TimesTen jar file is listed first.

On UNIX, CLASSPATH elements are separated by colon. For example:

```
set CLASSPATH .:install_dir/lib/ttjdbc5.jar
```

or

```
setenv CLASSPATH .:install_dir/lib/ttjdbc5.jar
```

On Windows, CLASSPATH elements are separated by semicolons.

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

For example, this is correct:

```
set CLASSPATH=.;install_dir/lib/ttjdbc5.jar
```

This is incorrect:

```
set CLASSPATH=.;"install_dir/lib/ttjdbc5.jar"
```

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

```
> java -version
```

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

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

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

```
.:install_dir/lib/ttjdbc5.jar:install_dir/lib/timestenjmsxla.jar
:install_dir/3rdparty/jms1.1/lib/jms.jar:install_dir/lib/orai18n.jar
```

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

Set the shared library path variable

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

System	Name of Variable
Linux	LD_LIBRARY_PATH
Solaris	LD_LIBRARY_PATH
HPUX	SHLIB_PATH or LD_LIBRARY_PATH
AIX	LIBPATH
Windows	PATH

Set the THREADS_FLAG variable (UNIX only)

The TimesTen JDBC driver uses native threads. Green threads are not supported.

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

In csh, the syntax is:

```
setenv THREADS_FLAG native
```

In sh, the syntax is:

```
THREADS_FLAG=native
export THREADS_FLAG
```

Set the PATH variable

Make sure the javac and java executables are both on your executable search path.

TimesTen Quick Start

During installation, you have the option of installing TimesTen Quick Start, which includes a variety of tutorials, demo applications, and other resources. If you choose to install Quick Start, it is installed by default under the directory *install_dir*/quickstart. On UNIX you have the option of specifying an alternative location. Regardless of where you install Quick Start, the home page for further information is *install_dir*/quickstart.html.

Quick Start provides tutorials, demos, and sample code for administration, access control, application development, replication, and caching, including the following areas. The demo data stores are first installed at the time that you install the Quick Start.

Configuration and setup:

- Creating, loading, and unloading a TimesTen database
- Setting up user accounts and privileges

- Using TimesTen utilities and built-in procedures for system operations
- Setting up In-Memory Database Cache to cache an Oracle database
- Setting up replication using active standby pairs and Oracle Clusterware
- Configuring a Java EE or J2EE application server to work with TimesTen
- Using Oracle SQL Developer with TimesTen
- Using Oracle Enterprise Manager with TimesTen

Application development:

- C applications using ODBC and XLA
- C applications using Oracle Call Interface (OCI) or the Pro*C/C++ Precompiler
- C++ applications using TimesTen TTClasses and XLA
- Java applications using JDBC and JTA
- PL/SQL applications

Performance and best practices:

- Response time demo
- Programming tips
- Database schema setup tips
- Hardware configuration tips
- Operating system configuration tips

Through the Quick Start home page, you can find information to set up and run the demos.

- Demo schema and setup: The `build_sampledb` script creates a sample database and demo schema. You must run this before you start using the demos.
- Demo environment and setup: The `ttquickstartenv` script, a superset of the `ttenv` script generally used for TimesTen setup, sets up the demo environment. You must run this each time you enter a session where you want to compile and run any of the demos.
- Demos and setup: Quick Start demos are in subdirectories under the `quickstart/sample_code` directory. For instructions on compiling and running the demos, see the README files in the subdirectories. Also see "Getting Started" and the various API links under "Sample Programs" on the Quick Start home page.

Viewing the online documentation

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

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

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

TimesTen on HP-UX Memory Windows

The following sections discuss installation and related topics for HP-UX Memory Windows:

- Installing TimesTen on HP-UX Memory Windows
- Using TimesTen in a memory window
- Address space considerations

Installing TimesTen on HP-UX Memory Windows

An instance of TimesTen can run in a memory window. A separate instance of TimesTen is required for each memory window. During installation, the TimesTen installer prompts you to indicate whether this instance is to be run in a memory window.

For a memory windows installation, the installer appends the instance name and port number of the daemon to `/etc/services.window` allowing the instance name to be used as a key to the `getmemwindow(1M)` command. Use the `getmemwindow <instance>` command to determine which port is being used by the instance.

Using TimesTen in a memory window

To use a TimesTen instance running in a memory window, you must launch your application using the HP-UX `setmemwindow(1M)` command.

For example, given instance `tt1121_32`, use:

```
% setmemwindow -j -i `getmemwindow tt1121_32` <prog>
```

TimesTen utilities are used without the `setmemwindow` command, for example:

```
% ttBackup ...
```

Address space considerations

The maximum size for any one data store remains 1GB with 32-bit TimesTen.

TimesTen allocates a single shared memory segment per data store. TimesTen may also allocate shared memory segments when configured to use the shared memory IPC mechanism for client/server.

The daemon and utility programs (programs) provided by TimesTen are linked with `EXEC_MAGIC`, using the `-N` option to `ld(1)`. You may change the TimesTen programs to be marked `SHMEM_MAGIC`, enabling 2GB of shared memory within the window. Any single data store is still limited to 1GB.

For example, to use `SHMEM_MAGIC`, log in as root and use:

```
# chatr -M tt_instance/bin/timesten* tt_instance/bin/*Cmdtt_instance/bin/ttcserver
```

To return to `EXEC_MAGIC`, use:

```
# chatr -N tt_instance/bin/timesten* tt_instance/ bin/ *Cmdtt_
instance/bin/ttcserver
```

To determine if a program is SHMEM_MAGIC or EXEC_MAGIC, use:

```
# chatr binary
```

The chatr(1M) command prints "normal executable" for EXEC MAGIC programs. It prints "SHMEM_MAGIC" for programs so marked.

> **Note:** If the TimesTen programs are marked SHMEM_MAGIC, the user application must be marked SHMEM_MAGIC also. Failure to mark the application SHMEM_MAGIC may result with an Invalid Argument error (EINVAL, errno=22) when attempting to connect to TimesTen.

HP-UX Memory Windows Troubleshooting

TimesTen support may ask for all of the following to diagnose a problem using memory windows.

- How many memory windows do you have configured?

  ```
  % /usr/sbin/kmtune -q max_mem_windows
  ```

- What is the maximum shared memory segment size?

  ```
  % /usr/sbin/kmtune -q shmmax
  ```

- How many windows are you using?

  ```
  % cat /etc/services.window
  ```

- Do you have the correct instance in your path?

  ```
  % ttVersion % ttStatus % getmemwindow tt_instance
  ```

- Can you connect with a utility provided by TimesTen?

  ```
  % ttIsql -connStr dsn=my_dsn
  ```

- If you installed the QuickStart, can you successfully run a demo program? The TimesTen demos are located under *install_dir*/quickstart/sample_code.
- What other segments are in use?

  ```
  % ipcs -m -a
  ```

- Does setmemwindow(1M) or a TimesTen utility such as ttStatus return silently when you expected output?
- Check the error status from the setmemwindow command.
- What does the memwin_stats tool show?

  ```
  % memwin_stats -w
  ```

  The memwin_stats tool may be downloaded from HP at ftp://contrib:9unsupp8@hprc.external.hp.com/
- What error are you getting when you try to connect?

The following list is not exhaustive but may help sort out the problem.

- Not enough core (ENOMEM, errno=12) indicates a problem allocating the requested amount of shared memory. Can you attach with small PermSize and TempSize attributes?

- Shared memory can be fragmented. Sometimes, you can attach with increasingly larger segments until you allocate what you want. Are you attempting to allocate more than 1GB within your window (2GB if using `SHMEM_MAGIC`)?
- Permission Denied (`EACCES, errno=13`) indicates that you are attempting to attach to the wrong instance or are pointing to the wrong memory window. Which `-i` argument is passed to `setmemwindow(1M)`?
- Invalid Argument (`EINVAL, errno=22`) indicates that the shared segment may have been allocated in another quadrant. Did you mark the TimesTen programs `SHMEM_MAGIC`? Did you also mark your application `SHMEM_MAGIC`?
- No space left on device (`ENOSPC, errno=28`) may indicate that the system is not configured for enough shared memory segments or identifiers or that the system may have insufficient swap space to allocate the shared segment. Check the values of `shmseg`, `shmmni`, `maxswapchunks` and run the `swapinfo(1M)` command.

Installation problems

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

- You have installed all required operating system patches.
- You are running a supported version of the operating system.
- You have made all required kernel configuration changes.
- You have sufficient disk space.
- On UNIX, you are a member of the TimesTen administrator's group. See "TimesTen instance administrators and TimesTen users groups" on page 1-16.
- On Windows, you are installing as user `Administrator` who is a member of the local `Administrators` group.

2 Data Store Upgrades

This chapter includes the following topics:

- Introduction
- Utilities used when upgrading TimesTen data stores
- Access control
- Data type conversion
- Data store character set conversion
- Upgrade modes
- Performing an in-place data store upgrade
- Performing an offline upgrade
- Performing an online upgrade with replication
- Performing an online upgrade with active standby pair replication
- Performing a Client/Server online upgrade from TimesTen release 6.0 and above
- Record of upgrades

Introduction

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

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

- Data store compatibility
- Data type compatibility
- Data store character set

Data store compatibility

Prior releases of TimesTen used either five numbers (7.0.0.0.0 and later), or three numbers (releases previous to 7.0.0.0.0, such as 6.0.17) to represent the release. Starting with TimesTen release 11.2.1.1.0, TimesTen release numbers consist of three components. The first three numbers in the release are used to indicate a major release of TimesTen, such as 11.2.1. The fourth number indicated the patch release of a major release of TimesTen, and the fifth number indicates the port patch. For example, TimesTen release number 11.2.1.5.1 indicates the first port patch of the fifth patch release of TimesTen release 11.2.1.

TimesTen data stores are not compatible between major releases, but they are always compatible between patch releases. For example, a data store created with TimesTen release 7.0.5.1.0 is not compatible with a TimesTen release 11.2.1.1.0 application, but a data store created with TimesTen 11.2.1.1.0 is compatible with a TimesTen release 11.2.1.5.0 application.

When referring to a TimesTen release, the release number is often abbreviated to the major release number. For example, release 11.2.1.1.0 may be abbreviated to 11.2.1.

Data type compatibility

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

TimesTen backward-compatible data types are replication-compatible with the data types in releases of TimesTen prior to 7.0. However, TimesTen backward-compatible data types are not compatible with TimesTen IMDB Cache to Oracle—only Oracle data types can be used with IMDB Cache to Oracle. If you wish to use IMDB Cache to Oracle, you must convert any original TimesTen data types to the new Oracle data types when performing a data store upgrade with ttMigrate. See "Converting data types to Oracle data types" on page 2-4 for details.

Oracle data types are not replication-compatible with releases of TimesTen prior to 7.0. If you wish to perform an upgrade that requires replication with a release of TimesTen from before 7.0, you must upgrade the original data types as TimesTen data types. See "Upgrading data types as TimesTen data types" on page 2-5 for more information.

Data store character set

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

Before upgrading your data store from a release prior to TimesTen 7.0, you must specify a data store character set by adding the DatabaseCharacterSet attribute to your data store's DSN. This attribute is ignored by releases of TimesTen prior to 7.0. In most cases, you want to choose a data store character set that makes sense for your region and that matches the character data that is already present in your data store. However, there are three important restrictions you must consider:

- If you plan to use the data store with TimesTen IMDB Cache to Oracle, you must specify a value for DatabaseCharacterSet that is the same as the character set specified for the Oracle database that the TimesTen data store connects to.

- Replication is not possible between data stores with different character sets. Because data stores created with releases of TimesTen prior to 7.0 do not have a data store character set specified, a special data store character set, TIMESTEN8, has been created, which allows replication compatibility between data stores created by the current release of TimesTen and those created by earlier releases. If you plan to perform the data store upgrade as an online upgrade with replication (see "Performing an online upgrade with replication" on page 2-15), then you must specify a DatabaseCharacterSet of TIMESTEN8 in your DSN.
- If you use TimesTen Client/Server and intend to connect to the upgraded data store with an application linked to a Client ODBC library from a release prior to TimesTen 7.0, you must specify a DatabaseCharacterSet of TIMESTEN8 in your DSN to ensure compatibility. See "Performing a Client/Server online upgrade from TimesTen release 6.0 and above" on page 2-30.

> **Note:** The TIMESTEN8 data store character set is intended for use only when transitioning from a release of TimesTen prior to 7.0. When you no longer need your data store to replicate to a pre-7.0 release of TimesTen, or to connect to a pre-7.0 client application, you should use `ttMigrate` to convert your data store to a data store character set other than TIMESTEN8. See "Data store character set conversion" on page 2-5 for details.

Utilities used when upgrading TimesTen data stores

TimesTen includes three utilities that may be used when upgrading data stores.

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

- Move a data store from one TimesTen instance to another TimesTen instance running the same major release version of TimesTen.
- Move a data store from one machine to another machine running the same major release version of TimesTen.
- Move a data store from one patch release of TimesTen (such as 11.2.1.1.0) to a different patch release of TimesTen (such as 11.2.1.2.0).

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

- Move a data store from one major release of TimesTen (such as 7.0) to a different major release (such as 11.2.1).
- Move a data store from a 32-bit version of TimesTen to a 64-bit version (or vice versa).
- Reduce the size of a TimesTen data store.

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

On Windows, these utilities use the ODBC driver manager.

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

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

Access control

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

```
GRANT ADMIN TO PUBLIC;
```

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

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

Data type conversion

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

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

Converting data types to Oracle data types

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

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

To convert the data types from a release prior to TimesTen 7.0 to Oracle data types, you must use the `-convertTypesToOra` option for `ttMigrate` when you restore your data store as part of your upgrade procedure. For example, if you restore the data store `salesdata` as part of an upgrade procedure, you may use the following to upgrade the data types to Oracle data types:

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

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

Note: Because the Oracle and TimesTen versions of some data types behave slightly differently, you should thoroughly test any applications written for releases of TimesTen prior to 7.0 with the Oracle data types before deploying them with the new release of TimesTen.

Upgrading data types as TimesTen data types

Note: If you intend to perform an online upgrade using replication, you must upgrade your data types as TimesTen data types. See "Online upgrades with replication" on page 2-8 for more information.

If you choose to upgrade the data types in a data store from a release prior to TimesTen 7.0 as TimesTen data types, you do not need to use any special options when restoring the data store with `ttMigrate`. The data types from a release prior to TimesTen 7.0 are automatically restored as TimesTen data types.

Note: The default TypeMode attribute for data stores is 0, which indicates Oracle type mode. In Oracle type mode, a data type such as CHAR has the semantics of the Oracle CHAR data type rather than the TimesTen CHAR data type. To guarantee compatibility with applications written for TimesTen releases before 7.0, you should configure the DSN for your data store with a TypeMode of 1 before restoring the data store with `ttMigrate` as part of the upgrade procedure.

Data store character set conversion

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

- You have specified the data store character set as TIMESTEN8 to upgrade your data store from a release of TimesTen prior to 7.0 using online upgrade with replication and/or client/server. After the upgrade is complete for all data stores and client applications, you should convert each data store from this special transitional character set to the national character set you prefer to use for your region. See "Converting from the TIMESTEN8 character set" on page 2-6.

- You need to change your data store's character set from the one that you originally specified to a new one that fits your requirements more closely. See "Converting from a character set other than TIMESTEN8" on page 2-6.

Converting from the TIMESTEN8 character set

You may use `ttMigrate` to convert a data store from TIMESTEN8 to any other character set by completing the following steps:

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

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

2. Destroy the data store:

   ```
   ttDestroy salesdata
   ```

3. Change the value of the DSN attribute DatabaseCharacterSet for your data store to the value specifying the new character set. For example, if you want your data store to use the WE8ISO8859P1 character set instead of TIMESTEN8, use the following line in your ODBCINI file:

   ```
   DatabaseCharacterSet=WE8ISO8859P1
   ```

4. Load the data store from the file using `ttMigrate` with the `-noCharsetConversion` command-line option. This option ensures that no character values are changed when the data is loaded into the DSN using the new character set. For example:

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

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

Converting from a character set other than TIMESTEN8

You may use `ttMigrate` to convert a data store from any character set to any other character set by completing the following steps:

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

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

2. Destroy the data store:

   ```
   ttDestroy salesdata
   ```

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

   ```
   DatabaseCharacterSet=WE8ISO8859P1
   ```

4. Load the data store from the file using `ttMigrate`. TimesTen automatically converts the character data from the character set the file was saved with to the character set used by the DSN. For example:

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

> **Note:** It is possible that character data is lost in the conversion process if no mapping exists from one character set to the other for a given character.

Upgrade modes

TimesTen allows you to perform these kinds of upgrades:

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

In-place upgrades

In-place upgrades are available for moving to a new patch release of TimesTen, such as moving from the first patch release of 11.2.1, release 11.2.1.1.0, to the second patch release of 11.2.1, release 11.2.1.2.0. As long as your TimesTen data stores do not reside in the TimesTen installation directory, you can uninstall an old release of TimesTen, install a new patch release of TimesTen and connect to existing data stores with the new release. No separate action is required for existing stores.

In-place upgrades require all applications to disconnect from the data store during the upgrade procedure. This kind of upgrade allows you to preserve the existing data store without using TimesTen's backup and migration utilities.

Offline upgrades

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

Offline upgrades are used to:

- move to a new major or patch release of TimesTen.
- move to a different directory or machine.
- reduce data store size.
- move between 32-bit and 64-bit data stores.

You should perform offline upgrades during a time when applications do not need continuous access to the data store. For example, if there is a maintenance window during weekends, schedule the upgrade during that time.

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

Online upgrades with replication

When upgrading to a new major release of TimesTen, you may have a mission-critical data store that needs to remain continuously available to your applications. You can use TimesTen replication to keep two copies of a data store synchronized, even when the data stores are from different releases of TimesTen, allowing your applications to stay connected to one copy of the data store while the other one is being upgraded. When the upgrade is finished, any updates that have been made on the active data store are transmitted immediately to the upgraded data store, and your applications can then be switched to the upgraded data store with no data loss and no down time. For more information, see "Performing an online upgrade with replication" on page 2-15.

The online upgrade process only supports updates to user tables during the upgrade. Data definition changes such as CREATE TABLE or CREATE INDEX are not replicated. In addition, all tables to be replicated must have a PRIMARY KEY or a unique index on non-nullable columns. Because two copies of the data store to be upgraded are required, you must have available twice the memory and disk space that the data store usually requires, if performing the upgrade on a single system.

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

Online upgrades with Client/Server

If you are upgrading a TimesTen Client/Server installation to a new major release, you can minimize downtime by performing a client/server online upgrade. During this process, TimesTen clients from the previous release are able to continue to communicate with a data store that has been upgraded to the new release. See "Performing a Client/Server online upgrade from TimesTen release 6.0 and above" on page 2-30.

The Client/Server online upgrade process minimizes, but does not eliminate, the interruption of client application access to the data store being upgraded. To maintain nearly continuous availability of a data store to all clients, you may use the approach outlined in "Online upgrades with replication" on page 2-8. This procedure keeps an identical copy of the data store available to the old release of TimesTen Server, while you upgrade the original to the new release. Once the upgraded original data store is available to the new release of TimesTen Server, you may stop the old release and start the new release, listening on the same port. The only interruption in availability using this method is the very brief period after the old server is stopped and before the new server is started.

On Windows, more than one release of TimesTen cannot be installed at the same time. Therefore, it is not possible to perform online upgrades with Client/Server on Windows if you are upgrading from a release of TimesTen prior to 6.0. It is also not possible to combine the Client/Server online upgrade procedure with that for performing an online upgrade with replication unless you use two different machines, one for each release of the data store that you are upgrading.

Performing an in-place data store upgrade

To upgrade an existing data store without exporting the data store to an external format, you can perform an in-place upgrade. This requires that all applications disconnect from the data store and that the data store be unloaded from shared memory. The following sections describe how to perform the in-place data store upgrade:

- Unloading a data store
- Moving to a new patch release of TimesTen

Unloading a data store

TimesTen data stores remain loaded in shared memory as long as any applications or TimesTen agents (such as the cache or replication agents) are connected to them. Data stores may also be kept in shared memory, even when no applications or agents are connected, if the RAM policy of the data store has been modified using the `ttAdmin` utility. (See "ttAdmin" in *Oracle TimesTen In-Memory Database Reference*.) Perform the following to unload a data store:

1. Disconnect all applications from the data store.
2. In this example, the data store `original` is from the older release. The data store `upgrade` is in the new release. If replication has been started, pause replication on the data stores, and then stop replication on the data store that you are unloading from memory, using the commands:

   ```
   ttRepAdmin -receiver -name upgrade -state pause original
   ttRepAdmin -receiver -name original -state pause upgrade
   ttAdmin -repStop upgrade
   ```

3. If the cache agent has been started for the data store, stop the cache agent with the command:

   ```
   ttAdmin -cacheStop upgrade
   ```

4. Verify that the RAM policy allows the data store to be unloaded. If the RAM policy is set to `manual`, unload the data store using the command:

   ```
   ttAdmin -ramUnload upgrade
   ```

   If the RAM policy is set to `always` or `inUse`, change it to `manual`. If the RAM policy is `inUse` and a grace period is set, set the grace period to 0 or wait for the grace period to elapse.

5. Use the `ttStatus` utility to verify that the data store has been unloaded from memory. (See "ttStatus" in *Oracle TimesTen In-Memory Database Reference*.)

Moving to a new patch release of TimesTen

All applications concurrently connected to a data store must be directly linked to a TimesTen ODBC driver of the same major release. TimesTen data stores from different patch releases are structurally equivalent or identical. For example, when upgrading from release 11.2.1.1.0 to 11.2.1.3.0, you do not need to migrate your existing data stores using the `ttMigrate` utility. However, during the installation of a new major or patch release, you should disconnect your application and stop the TimesTen daemon. If you have not explicitly performed these steps, the prior release's TimesTen daemon process is stopped, effectively disconnecting all applications from the data store. While preparing for the upgrade, make sure that all data stores are unloaded from memory before upgrading TimesTen.

For a description of the procedures for unloading a data store from memory, see "Unloading a data store" on page 2-9.

Performing an offline upgrade

You can do an offline upgrade by exporting the data store into an external file and then importing the data store with the desired changes. These update procedures require that all applications be disconnected from the data store and that the data store be unloaded from shared memory. For applications that require continuous availability, see "Performing an online upgrade with replication" on page 2-15.

Note: If the data store to be upgraded is replicated, you must use `ttMigrate` to move the data store between releases. In addition if you rename table owners using the `ttMigrate -r -rename` option, you need to also rename the table owners in every other data store involved in the replication scheme.

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

- move to a new major or patch release of TimesTen.
- reduce data store size.
- move between 32-bit and 64-bit data stores.

The general steps in an offline upgrade include:

1. Disconnect all applications from the data store and unload the data store from memory. For more information, see "Unloading a data store" on page 2-9.
2. Use either `ttMigrate` with the `-c` and `-noRepUpgrade` options or `ttBackup` to back up the data store.
3. Install the new release of TimesTen. For more information, see "TimesTen Installation" on page 17.
4. Use either `ttMigrate` with the `-r` and `-noRepUpgrade` options or `ttRestore` to restore the backed up data store to the new TimesTen release.
5. Reconnect applications to the upgraded data store.

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

Moving to a different directory

The TimesTen daemon identifies a data store by the full path name of the data store's checkpoint files. To move a TimesTen data store to a different directory, back up the data store using the `ttBackup` utility, create a new DSN definition that specifies the new data store pathname, then restore the data store into its new location using the `ttRestore` utility. When you have verified that the data store functions properly in the new location, free up the disk space by using `ttDestroy` to remove the old data store.

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

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

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

> **Note:** You must reconfigure replication if the moved data store has been configured to replicate.

Moving to a different machine

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

> **Note:** Before moving a replicated data store to another machine, you should be experienced with TimesTen replication. We highly advise calling TimesTen customer support for help with this procedure.

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

1. Back up the data store on the original system using `ttBackup`.
2. Move the backup to the new system.
3. Reconfigure the replication scheme on any data stores the replicate to the moved store to specify the new host machine. (See *Oracle TimesTen In-Memory Database TimesTen to TimesTen Replication Guide* for details on how to specify a host in a replication scheme.)
4. Restore the backup using `ttRestore`.

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

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

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

Reducing data store size

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

To reduce the allocated size of the permanent partition of a data store, save a copy using the ttMigrate utility with the -noRepUpgrade option. Then recreate the data store with a smaller permanent partition size and restore the data.

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

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

1. Back up the old data store with ttMigrate -c -noRepUpgrade.
2. Create a new DSN definition for the new copy of the data store with a smaller PermSize value.
3. Restore the backup with ttMigrate -r -noRepUpgrade.

Note: If you wish to modify the original DSN rather than create a new one in Step 2, then you must destroy the original data store files using the ttDestroy utility before restoring from the backup.

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

1. ttMigrate -c -noRepUpgrade DSN=salesdata /tmp/salesbackup

2. `ttDestroy salesdata`
3. Update the DSN `salesdata` to have a size of 100 MB. See "Changing data store size" in the *Oracle TimesTen In-Memory Database Operations Guide*.
4. `ttMigrate -r -noRepUpgrade "DSN=salesdata;AutoCreate=1" /tmp/salesbackup`

Note: The temporary partition size of a data store may be changed by simply modifying the TempSize attribute of the DSN, unloading the data store from memory, and then reconnecting to it. See "Unloading a data store" on page 2-9 for the steps to unload a data store from memory.

Moving between 32-bit and 64-bit data stores

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

1. Export the 32-bit data store using the TimesTen 32-bit `ttMigrate` utility with the `-noRepUpgrade` option.
2. Create a Data Source Name (DSN) for the 64-bit data store. See "Creating a DSN on UNIX" or "Creating a DSN on Windows" in *Oracle TimesTen In-Memory Database Operations Guide*.
3. Import the file created in Step 1 into the 64-bit data store DSN using the 64-bit `ttMigrate` utility with the `-noRepUpgrade` option.

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

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

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

Moving to a different major release of TimesTen

You can have multiple TimesTen major releases installed on a system at the same time. However, TimesTen data stores created by one major release cannot be accessed directly by applications of a different major release. To migrate data between TimesTen major releases, for example from TimesTen 6.0 to 11.2.1, you need to export the data using the `ttMigrate` utility from the old release and import it using the `ttMigrate` utility from the new release. The procedure for this upgrade is similar to the steps outlined in "Moving between 32-bit and 64-bit data stores" on page 2-13.

If you need to upgrade two or more data stores that are replicating to each other, you must perform a few extra steps to ensure that replication continues to operate during and after the upgrade. For example, to migrate two replicating data stores, `master1` on host machine `masterhost` and `subscriber1` on host machine subscriberhost, from TimesTen release 6.0 to TimesTen release 11.2.1, perform the following steps:

> **Note:** TimesTen 11.2.1 introduces a new feature called Cache Grid. By default, this feature is enabled, and requires additional configuration before cache groups can be created on a data store. If you are upgrading a data store containing cache groups from a release previous to 11.2.1, you should set the attribute **CacheGridEnable** to 0 in each DSN definition before beginning. For more information, see the *Oracle In-Memory Database Cache User's Guide*.

1. Configure the replication scheme on both data stores to use static TCP/IP ports for replication. This is necessary because the intermediate result of this procedure is that replication occurs between two different releases of TimesTen. Each release does not necessarily know how to find the main daemon of the other to have the replication ports assigned dynamically. See "Port assignments" in the *Oracle TimesTen In-Memory Database TimesTen to TimesTen Replication Guide* for more information.
2. On the machine `masterhost`, use the 6.0 release's `ttAdmin` utility to stop the replication daemon on the data store:

   ```
   ttAdmin -repStop master1
   ```

3. Next, use the 6.0 release's `ttMigrate` utility with the `-c` option to back up data store `master1` to a binary file:

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

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

   ```
   ttDestroy /data_store_path/master1
   ```

5. Use the 11.2.1 release's `ttMigrate` utility with the `-r` option to restore data store `master1` from the binary file. Restoring the data store automatically upgrades it from release 6.0 to release 11.2.1. If you are restoring a very large data store, you should use the `-C` option to tell `ttMigrate` to perform a checkpoint operation on the data store periodically. This saves time if the restore fails at some point before the operation has been completed. For more information, see "Checkpoints" in the *Oracle TimesTen In-Memory Database Operations Guide*.

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

6. Use the 11.2.1 release's `ttAdmin` utility to start the replication daemon:

   ```
   ttAdmin -repStart master1
   ```

   Replication is now occurring between the data store `master1` on release 11.2.1 and the data store `subscriber1` on release 6.0.

   Now upgrade data store `subscriber1` from release 6.0 to release 11.2.1. Perform the following steps:

7. On the machine `subscriberhost`, use the 6.0 release's `ttAdmin` utility to stop the replication daemon:

   ```
   ttAdmin -repStop subscriber1
   ```

8. Use the 6.0 release's `ttDestroy` utility to destroy data store `subscriber1`, where the data store's files are located in the directory *`data_store_path`*:

   ```
   ttDestroy data_store_path/subscriber1
   ```

9. If you are upgrading from a release previous to 11.2.1, you must create a user with ADMIN privileges to duplicate a data store using replication. For example, to create the user pat with the password patpwd on the standby master data store, use:

   ```
   CREATE USER pat IDENTIFIED BY patpwd;
   GRANT ADMIN TO pat;
   ```

10. Use the 11.2.1 release's `ttRepAdmin` utility with the `-duplicate` option to duplicate data store `subscriber1` from data store `master1` using replication.

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

11. Use the 11.2.1 release's `ttAdmin` utility to start the replication daemon:

    ```
    ttAdmin -repStart subscriber1
    ```

The data stores are now upgraded and replicating to each other.

Performing an online upgrade with replication

In "Performing an offline upgrade" on page 2-10 we showed how to perform various maintenance operations on TimesTen data stores that require that all applications be stopped. This section describes how to use the TimesTen replication feature to perform online upgrades for applications that require continuous data availability. You can do an online upgrade when moving between major TimesTen releases. If moving to a patch release, you may instead perform an in-place or offline upgrade.

Normally, applications that require high availability of their data use TimesTen replication to keep at least one extra copy of their data stores up to date. An online upgrade works by keeping one 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 data stores, as described in the *Oracle TimesTen In-Memory Database TimesTen to TimesTen Replication Guide*.

Note: Replication functions across releases only if the data store of the more recent release of TimesTen was upgraded from a data store of the older release of TimesTen. A data store created in the more recent release of TimesTen is not guaranteed to replicate correctly with the older release. For example, replication between a data store created in the 5.1 release of TimesTen and a data store created in the 7.0 release of TimesTen is not supported. However, if one data store was created in the 5.1 release, and the peer data store was created in the 5.1 release and then upgraded to the 7.0 release, replication between them is supported.

Note: For security reasons, replication is not normally allowed between data store in TimesTen 7.0 and data stores in releases previous to TimesTen 7.0. To perform an online upgrade with replication, you must start the main TimesTen 7.0 or later daemon with the `-insecure-backwards-compat` option. See "Communicating with older releases of TimesTen" of the *Oracle TimesTen In-Memory Database Operations Guide* for more information.

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

- Overview
- Limitations
- Requirements
- Online upgrade example

Overview

Upgrading a data store from an older TimesTen release to a newer release is performed by disconnecting all applications from one of two replicated copies of a data store, making a backup of the data store with the `ttMigrate` utility from the older release, loading the backup into a newer release data store using the `ttMigrate` utility from the newer release, and then reconnecting all applications to the upgraded data store.

Note: The `ttMigrate -r -rename` option, used to rename the owner of data store objects, cannot be used with online upgrades.

The general steps in an online upgrade include:

1. Disconnect all applications from the data store being upgraded.
2. Shut down replication on the system being upgraded.
3. Back up the data store residing on the system being upgraded using the older release's `ttMigrate` with the `-c` option.
4. Install the newer release of TimesTen on the system being upgraded.
5. Restore the replicated data store on the system being upgraded using the newer release's `ttMigrate` with the `-r` option.
6. Reconnect all applications to the upgraded data store.
7. Restart replication on the upgraded system.

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

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

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

Step	**Upgrade system**	**Active system**
1.	Configure replication to replicate to the active system using static ports.	Configure replication to replicate to the upgrade system using static ports.
2.		Connect all applications to the active data store, if they are not already connected.
3.	Disconnect all applications from the upgrade data store.	
4.		Set replication to the upgrade system to the `pause` state.
5.	Wait for updates to propagate to the active system.	
6.	Stop replication.	
7.	Back up the data store with `ttMigrate -c`.	
8.	Stop the TimesTen daemon for the old release of TimesTen.	
9.	Install the new release of TimesTen.	
10.	Create a Data Source Name for the upgraded data store for the new TimesTen release, using the ODBC Data Source Administrator on Windows or the `.odbc.ini` file on UNIX.	
11.	Restore the data store from the backup with `ttMigrate -r`.	
12.	Clear the replication bookmark and logs using `ttRepAdmin -receiver -reset` and by setting replication to the active system to the `stop` and then the `start` state.	
13.	Start replication.	
14.		Set replication to the upgrade system to the `start` state, ensuring that the accumulated updates propagate once replication is restarted.
15.		Start replication.
16.		Wait for all of the updates to propagate to the upgrade system.
17.	Reconnect all applications to the upgrade data store.	

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

Limitations

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

Requirements

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

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

Additional disk space must be allocated to hold a backup copy of the data store made by the `ttMigrate` utility. The size of the backup copy is typically about the same as the in-use size of the data store. This size may be determined by querying the `sys.monitor` table, using `ttIsql`:

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

Online upgrade example

This section describes how to perform an online upgrade of two bidirectionally replicated TimesTen data stores, using a concrete example.

Note: TimesTen 11.2.1 introduces a new feature called Cache Grid. By default, this feature is enabled, and requires additional configuration before cache groups may be created on a data store. If you are upgrading a data store containing cache groups from a release previous to 11.2.1, you should set the attribute **CacheGridEnable** to 0 in each DSN definition before beginning. For more information, see the *Oracle In-Memory Database Cache User's Guide*.

We'll refer to the two TimesTen systems being upgraded as the *upgrade* system, on which the TimesTen instance and the data store are upgraded, and the *active* system, which remains operational and connected to the application for the duration of the upgrade. After this procedure is completed, the same steps can be followed to upgrade the active system. However, you may prefer to delay conversion of the active system to first test the upgraded TimesTen instance.

The upgrade system in this example consists of the data store `upgrade` on the server `upgradehost`. The active system consists of the data store `active` on the server `activehost`.

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

Step	Upgrade system	Active system

1.	Use ttIsql to alter the replication scheme repscheme, setting static replication port numbers so that the data stores 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 data stores 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 data store. Any workload being run on the upgrade system must start running on the active system instead.	Use the ttRepAdmin utility to pause replication from the data store active to the data store upgrade: `ttRepAdmin -connStr DSN=active -receiver -name upgrade -state pause` This command temporarily stops the replication of updates from the data store active to the data store upgrade, but it retains any updates made to active in the data store transaction log files. The updates made to active during the upgrade procedure are applied later, when upgrade is brought back up. For details on setting the replication state, see "Setting the replication state of subscribers" in *Oracle TimesTen In-Memory Database TimesTen to TimesTen Replication Guide*.
3.	Wait for all replication updates to be sent to the data store active. You can verify that all updates have been sent by applying a recognizable update to a table reserved for that purpose on the data store upgrade. When the update appears in the data store active, you know that all previous updates have been sent.	
4.	Stop the replication agent with ttAdmin: `ttAdmin -repStop upgrade` From this point on, no updates are sent to the data store active.	Stop the replication agent with ttAdmin: `ttAdmin -repStop active` From this point on, no updates are sent to the data store upgrade. For details on starting and stopping replication agents, see "Starting and stopping the replication agents" in the *Oracle TimesTen In-Memory Database TimesTen to TimesTen Replication Guide*.

Step	Upgrade system	Active system

5.	Use `ttRepAdmin` to stop replication from the data store `upgrade` to the data store `active`:
	`ttRepAdmin -connStr DSN=upgrade`
	`  -receiver -name active`
	`  -state stop`
	This step prevents `active` from accumulating updates to send to `upgrade` and resets some of the replication bookmarks.	

6.	Use `ttMigrate` to back up the data store `upgrade`. If the data store is very large, this step could take a significant amount of time. If sufficient disk space is free on the `/backup` file system, the following `ttMigrate` command can be used:
	`ttMigrate -c DSN=upgrade`
	`/backup/upgrade.dat`	

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

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

9.	Use `ttMigrate` to load the backup created in Step 6. into the data store `upgrade` for the new release of TimesTen:
	`ttMigrate -r`
	` "DSN=upgrade;AutoCreate=0"`
	` /backup/upgrade.dat`
	If the data store is temporary (Temporary=1), first use `ttAdmin -ramLoad`:
	`ttAdmin -ramLoad upgrade`
	Note: In this step, you must use the `ttMigrate` utility supplied with the new release of TimesTen to which you are upgrading.	

Step	Upgrade system	Active system

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

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

Performing an online upgrade with active standby pair replication

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

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

Upgrading the data stores in an active standby pair with no cache groups

This section includes the following topics:

- Upgrades for TimesTen patch releases on the standby master data store and subscriber stores
- Upgrades for TimesTen patch releases on the active master data store
- Upgrades for major TimesTen releases, application software and hardware

Upgrades for TimesTen patch releases on the standby master data store and subscriber stores

To upgrade to a TimesTen patch release on the standby master data store and subscriber stores, complete the following tasks on each store:

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

   ```
   ttAdmin -repStop master2
   ```

2. Install the TimesTen patch. See "Performing an in-place data store upgrade" on page 2-9.
3. Restart the replication agent using the `ttRepStart` built-in procedure or the `ttAdmin` utility:

   ```
   ttAdmin -repStart master2
   ```

Upgrades for TimesTen patch releases on the active master data store

To upgrade the active master data store, you must first reverse the roles of the active and standby master data stores, and then perform an in-place upgrade:

1. Pause any applications that are generating updates on the active master data store.
2. Execute the `ttRepSubscriberWait` built-in procedure on the active master data store, using the DSN and host of the standby master data store. For example, to ensure that all transactions are replicated to the standby master `master2` on the host `master2host`, use:

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

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

   ```
   ttAdmin -repStop master1
   ```

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

   ```
   call ttRepDeactivate;
   ```

5. On the standby master data store, set the data store to the ACTIVE state using the `ttRepStateSet` built-in procedure. This store 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 data store that is now acting as the active master (in this example, data store `master2`).

7. Upgrade the former active master data store, which is now the standby master data store. See "Performing an in-place data store upgrade" on page 2-9.

8. Restart replication on the upgraded data store using the `ttRepStart` built-in procedure or the `ttAdmin` utility:

   ```
   ttAdmin -repStart master2
   ```

9. If you wish to make the newly upgraded data store the active master data store again, see "Reversing the roles of the active and standby master data stores" in *Oracle TimesTen In-Memory Database TimesTen to TimesTen Replication Guide*.

Upgrades for major TimesTen releases, application software and hardware

When you upgrade an active standby pair across major releases of TimesTen, you must explicitly specify the TCP/IP port for each data store. If your active standby pair replication scheme is not already configured with a PORT attribute for each data store, you must prepare for the upgrade by following these steps:

1. Stop the replication agent on every data store using the call `ttRepStop` built-in procedure or the `ttAdmin` utility. For example, to stop the replication agent on the data store `master1`, use:

   ```
   ttAdmin -repStop master1
   ```

2. On the active master data store, use the ALTER ACTIVE STANDBY PAIR statement to specify a PORT attribute for every data store in the active standby pair. For example, to set a PORT attribute for the data stores `master1` on the host `master1host`, `master2` on the host `master2host`, and `subscriber1` on the host `subscriber1host`, use:

   ```
   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 data store and all of the subscribers using the `ttDestroy` utility. For example, to destroy the data store `subscriber1`, use:

   ```
   ttDestroy subscriber1
   ```

4. Follow the normal procedure to start an active standby pair and duplicate the standby and subscriber data stores from the active master. See "Setting up an active standby pair" in *Oracle TimesTen In-Memory Database TimesTen to TimesTen Replication Guide*.

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

1. Tell the active master data store to stop replicating updates to the standby master by executing the `ttRepStateSave` built-in procedure on the active master data store. For example, to stop replication to the standby master data store `master2` on host `master2host`, use:

   ```
   call ttRepStateSave( 'FAILED', 'master2', 'master2host' );
   ```

2. Stop the replication agent on the standby master data store using the `ttRepStop` built-in procedure or the `ttAdmin` utility. For example, to stop the replication agent for the standby master data store `master2`, use:

```
ttAdmin -repStop master2
```

3. Upgrade the node where the standby master data store resides. See "Performing an offline upgrade" on page 2-10.

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

   ```
   ttAdmin -repStart master2
   ```

   When the upgraded standby master data store has become synchronized with the active master data store, the upgraded standby master data store moves from the RECOVERING state to the STANDBY state. The upgraded standby master data store also starts sending updates to the subscribers. You can determine when the standby master data store is in the STANDBY state by calling the `ttRepStateGet` built-in procedure on the standby master data store:

   ```
   call ttRepStateGet;
   ```

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

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

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

7. Stop the replication agent on the active master data store using the `ttRepStop` built-in procedure or the `ttAdmin` utility. For example, to stop the replication agent for the active master data store `master1`, use:

   ```
   ttAdmin -repStop master1
   ```

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

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

9. Tell the new active master data store (`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 data store. For example, to stop replication to the standby master data store `master1` on host `master1host`, use:

   ```
   call ttRepStateSave( 'FAILED', 'master1', 'master1host' );
   ```

10. Destroy the former active master data store using the `ttDestroy` utility:

    ```
    ttDestroy master1
    ```

11. Perform the upgrade on the node where the master data store was destroyed. You do not need to do anything to upgrade the data store itself, since it is already been destroyed.

12. If you are upgrading from a release previous to 11.2.1, you must create a user with ADMIN privileges on the current active master data store to duplicate that store. For example, to create the user `pat` with the password `patpwd` on the standby master data store, use:

    ```
    CREATE USER pat IDENTIFIED BY patpwd;
    ```

```
GRANT ADMIN TO pat;
```

13. Duplicate the new standby master data store from the current active master data store using the `ttRepAdmin` utility. For example, to duplicate the data store `master2` on the host `master2host` to the data store `master1`, use the following on the host containing the data store `master1`:

    ```
    ttRepAdmin -duplicate -from master2 -host master2host -uid pat -pwd patpwd
     -setMasterRepStart master1
    ```

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

    ```
    ttAdmin -repStart master1
    ```

15. Stop the replication agent on the first subscriber data store using the `ttRepStop` built-in procedure or the `ttAdmin` utility. For example, to stop the replication agent for the subscriber data store `subscriber1`, use:

    ```
    ttAdmin -repStop subscriber1
    ```

16. Destroy the subscriber data store using the `ttDestroy` utility:

    ```
    ttDestroy subscriber1
    ```

17. Perform the upgrade on the node where the subscriber data store was destroyed.
18. Duplicate the subscriber data store from the standby master data store using the `ttRepAdmin` utility:

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

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

    ```
    ttAdmin -repStart subscriber1
    ```

20. Repeat Step 15 through Step 19 for each other subscriber data store.

Upgrading the data stores in an active standby pair with cache groups

This section includes the following topics:

- Upgrades for TimesTen patch releases on the standby master data store and subscriber stores
- Upgrades for TimesTen patch releases on the active master data store
- Upgrades for major TimesTen releases, application software and hardware

Upgrades for TimesTen patch releases on the standby master data store and subscriber stores

To upgrade to a TimesTen patch release on the standby master data store and subscriber stores, complete the following tasks on each store:

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

    ```
    ttAdmin -repStop master2
    ```

2. Stop the cache agent on the data store using the `ttCacheStop` built-in procedure or the `ttAdmin` utility:

```
ttAdmin -cacheStop master2
```

3. Install the TimesTen patch. See "Performing an in-place data store upgrade" on page 2-9.
4. Restart the cache agent using the `ttCacheStart` built-in procedure or the `ttAdmin` utility:

```
ttAdmin -cacheStop master2
```

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

```
ttAdmin -repStart master2
```

Upgrades for TimesTen patch releases on the active master data store

To upgrade the active master data store, you must first reverse the roles of the active and standby master data stores, and then perform an in-place upgrade:

1. Pause any applications that are generating updates on the active master data store.
2. Execute the `ttRepSubscriberWait` built-in procedure on the active master data store, using the DSN and host of the standby master data store. For example, to ensure that all transactions are replicated to the standby master `master2` on the host `master2host`, use:

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

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

```
ttAdmin -repStop master1
```

4. Stop the cache agent on the current active master data store using the `ttCacheStop` built-in procedure or the `ttAdmin` utility:

```
ttAdmin -cacheStop master1
```

5. Execute the `ttRepDeactivate` build-in procedure on the current active master data store. This puts the store in the IDLE state:

```
call ttRepDeactivate;
```

6. On the standby master data store, set the data store to the ACTIVE state using the `ttRepStateSet` built-in procedure. This store 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 data store that is now acting as the active master (in this example, data store `master2`).
8. Upgrade the former active master data store, which is now the standby master data store. See "Performing an in-place data store upgrade" on page 2-9.
9. Restart the cache agent on the upgraded data store using the `ttCacheStart` built-in procedure or the `ttAdmin` utility:

```
ttAdmin -cacheStart master1
```

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

    ```
    ttAdmin -repStart master1
    ```

11. If you wish to make the newly upgraded data store the active master data store again, see "Reversing the roles of the active and standby master data stores" in *Oracle TimesTen In-Memory Database TimesTen to TimesTen Replication Guide*.

Upgrades for major TimesTen releases, application software and hardware

When you upgrade an active standby pair across major releases of TimesTen, you must explicitly specify the TCP/IP port for each data store. If your active standby pair replication scheme is not already configured with a PORT attribute for each data store, you must prepare for the upgrade by following these steps:

1. Tell the active master data store to stop replicating updates to the standby master by executing the `ttRepStateSave` built-in procedure on the active master data store. For example, to stop replication to the standby master data store `master2` on host `master2host`, use:

    ```
    call ttRepStateSave( 'FAILED', 'master2', 'master2host' );
    ```

2. Stop the replication agent on every data store using the call `ttRepStop` built-in procedure or the `ttAdmin` utility. For example, to stop the replication agent on the data store `master1`, use:

    ```
    ttAdmin -repStop master1
    ```

3. On the active master data store, use the ALTER ACTIVE STANDBY PAIR statement to specify a PORT attribute for every data store in the active standby pair. For example, to set a PORT attribute for the data stores `master1` on the host `master1host`, `master2` on the host `master2host`, and `subscriber1` on the host `subscriber1host`, use:

    ```
    ALTER ACTIVE STANDBY PAIR
     ALTER STORE master1 ON "master1host" SET PORT 30000
     ALTER STORE master2 ON "master2host" SET PORT 30001
     ALTER STORE subscriber1 ON "subscriber1host" SET PORT 30002;
    ```

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

    ```
    ttDestroy subscriber1
    ```

5. Follow the normal procedure to start an active standby pair and duplicate the standby and subscriber data stores from the active master. See "Setting up an active standby pair" in *Oracle TimesTen In-Memory Database TimesTen to TimesTen Replication Guide*.

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

Note: TimesTen 11.2.1 introduces a new feature called Cache Grid. By default, this feature is enabled, and requires additional configuration before cache groups may be created on a data store. If you are upgrading from a release previous to 11.2.1, you should set the attribute **CacheGridEnable** to 0 in each DSN definition before beginning. For more information, see the *Oracle In-Memory Database Cache User's Guide*.

1. Tell the active master data store to stop replicating updates to the standby master by executing the `ttRepStateSave` built-in procedure on the active master data store. For example, to stop replication to the standby master data store `master2` on host `master2host`, use:

   ```
   call ttRepStateSave( 'FAILED', 'master2', 'master2host' );
   ```

2. Stop the replication agent on the standby master data store using the `ttRepStop` built-in procedure or the `ttAdmin` utility. For example, to stop the replication agent for the standby master data store `master2`, use:

   ```
   ttAdmin -repStop master2
   ```

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

   ```
   ttAdmin -cacheStop master2
   ```

4. Upgrade the node where the standby master data store resides. See "Performing an offline upgrade" on page 2-10.
5. Start the cache agent on the standby master data store using the `ttCacheStart` built-in procedure or the `ttAdmin` utility:

   ```
   ttAdmin -cacheStart master2
   ```

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

   ```
   ttAdmin -repStart master2
   ```

   When the upgraded standby master data store has become synchronized with the active master data store, the upgraded standby master data store moves from the RECOVERING state to the STANDBY state. The upgraded standby master data store also starts sending updates to the subscribers. You can determine when the standby master data store is in the STANDBY state by calling the `ttRepStateGet` built-in procedure on the standby master data store:

   ```
   call ttRepStateGet;
   ```

7. Pause any applications that are generating updates on the active master data store.
8. Execute the `ttRepSubscriberWait` built-in procedure on the active master data store, using the DSN and host of the standby master data store. For example, to ensure that all transactions are replicated to the standby master `master2` on the host `master2host`, use:

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

9. Stop the replication agent on the active master data store using the `ttRepStop` built-in procedure or the `ttAdmin` utility. For example, to stop the replication agent for the active master data store `master1`, use:

   ```
   ttAdmin -repStop master1
   ```

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

    ```
    ttAdmin -cacheStop master1
    ```

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

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

12. Tell the new active master data store (`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 data store. For example, to stop replication to the standby master data store `master1` on host `master1host`, use:

    ```
    call ttRepStateSave( 'FAILED', 'master1', 'master1host' );
    ```

13. Destroy the former active master data store using the `ttDestroy` utility:

    ```
    ttDestroy master1
    ```

14. Perform the upgrade on the node where the master data store was destroyed. You do not need to do anything to upgrade the data store itself, since it is already been destroyed.

15. If you are upgrading from a release previous to 11.2.1, you must create a user with ADMIN privileges on the current active master data store to duplicate that store. For example, to create the user `pat` with the password `patpwd` on the standby master data store, use:

    ```
    CREATE USER pat IDENTIFIED BY patpwd;
    GRANT ADMIN TO pat;
    ```

16. Duplicate the new standby master data store from the current active master data store using the `ttRepAdmin` utility. For example, to duplicate the data store `master2` on the host `master2host` to the data store `master1`, where the cache user ID is `terry` and the cache password is `terrypwd`, use the following on the host containing the data store `master1`:

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

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

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

    ```
    ttAdmin -cacheStart master1
    ```

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

    ```
    ttAdmin -repStart master1
    ```

19. Stop the replication agent on the first subscriber data store using the `ttRepStop` built-in procedure or the `ttAdmin` utility. For example, to stop the replication agent for the subscriber data store `subscriber1`, use:

    ```
    ttAdmin -repStop subscriber1
    ```

20. Destroy the subscriber data store using the `ttDestroy` utility:

    ```
    ttDestroy subscriber1
    ```

21. Perform the upgrade on the node where the subscriber data store was destroyed.
22. Duplicate the subscriber data store from the standby master data store using the `ttRepAdmin` utility:

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

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

    ```
    ttAdmin -repStart subscriber1
    ```

24. Repeat Step 15 through Step 19 for each other subscriber data store.

Performing a Client/Server online upgrade from TimesTen release 6.0 and above

TimesTen Server is able to talk directly to all user applications linked with TimesTen Client ODBC driver release 6.0 or higher. There are at least two possible ways to upgrade a TimesTen Client/Server installation, depending on your requirements for client access to the data store:

- If you do not need the data store being upgraded to be available continuously to your client applications, you may simply stop the old server, perform the migration of the data store using `ttMigrate`, and then start the new release of the server, which should be configured to listen on the same server port.
- If it is critical that the data store be continuously available to the client applications, you may use the procedure outlined in "Performing an online upgrade with replication" on page 2-15 to keep a second copy of the data store available during the migration of the first copy.

Note: For security reasons, client/server communication is not normally allowed between TimesTen 7.0 and previous releases. To perform an online upgrade with client/server, you must start the main TimesTen 7.0 daemon with the `-insecure-backwards-compat` option. See "Communicating with older releases of TimesTen" of the *Oracle TimesTen In-Memory Database Operations Guide* for more information.

Client/Server online upgrade

To perform an online upgrade of a TimesTen Client/Server system to a new major release with minimal reconfiguration, perform the following steps:

1. Stop the TimesTen Server on the old release of TimesTen. From this point until the TimesTen Server for the new release is started, your client applications do not have access to the data store. Any attempted updates to the data store by the clients fail, and you should stop user applications if necessary.
2. Install the new release of TimesTen. At install time, configure the server to listen on the same port as the old release of TimesTen.
3. Use `ttMigrate` to migrate the data store from the old release to the new release. See "Moving between 32-bit and 64-bit data stores" on page 2-13 for an example of this procedure.
4. Start the new release of TimesTen Server, if you have not already. The client applications now have access to the upgraded data stores.

Note: Since both releases of TimesTen Server have been configured to listen on the same port in this procedure, if you wish to restart the the server for the old release you must first configure it to listen on a different port.

Client/Server online upgrade with continuous access to the data store

The migration of a data store using `ttMigrate` can be a long process if your data stores are very large. If it is critical that your client applications have nearly continuous access to the data store during the Client/Server online upgrade procedure, you may incorporate the procedure for performing an online upgrade with replication by following these steps:

1. Install the new release of TimesTen, making sure to configure TimesTen Server to listen on the same port as the old release. The installation script asks whether you would like to start the new server. You must answer "no."
2. Follow the steps in "Performing an online upgrade with replication" on page 2-15 to upgrade one copy of the data store. The client applications remain connected to the other, non-upgraded copy of the data store.
3. Disconnect all clients from the old release of the data store.
4. Stop the old release of TimesTen Server.
5. Wait for all updates to finish replicating from the old release of the data store to the new release.

6. Start the new release of TimesTen Server. It begins listening on the same port as the old release, and client applications may now connect to the new release of the data store with no change to their configurations.

Record of upgrades

If replication is configured on a data store, a new set of tables with the prefix `ttrep_schema_version-` is created each time that the `ttMigrate -r` command is used. These tables provide a history of the data store's replication scheme from release to release.

These tables do not take up much space and are helpful when debugging upgrade problems. However, you can drop these tables if you do not experience problems with replication after running `ttMigrate`.

For example, after two migrations, the data store might contain tables which look like this:

```
TTREP_SCHEMA_VERSION_004.REPELEMENTS
TTREP_SCHEMA_VERSION_004.REPLICATIONS
TTREP_SCHEMA_VERSION_004.REPPEERS
TTREP_SCHEMA_VERSION_004.REPSTORES
TTREP_SCHEMA_VERSION_004.REPSUBSCRIPTIONS
TTREP_SCHEMA_VERSION_004.REPTABLES
TTREP_SCHEMA_VERSION_004.TTSTORES
TTREP_SCHEMA_VERSION_005.REPELEMENTS
TTREP_SCHEMA_VERSION_005.REPLICATIONS
TTREP_SCHEMA_VERSION_005.REPPEERS
TTREP_SCHEMA_VERSION_005.REPSTORES
TTREP_SCHEMA_VERSION_005.REPSUBSCRIPTIONS
TTREP_SCHEMA_VERSION_005.REPTABLES
TTREP_SCHEMA_VERSION_005.TTSTORES
```

Index