Oracle® Solaris Cluster Data Service for Oracle TimesTen Guide

Part No: E40146
July 2013, Revision A

ORACLE

130604@25097

Contents

Preface 7

1 Installing and Configuring HA for TimesTen 11
HA for TimesTen Overview 11
Overview of Installing and Configuring HA for TimesTen 12
Planning the HA for TimesTen Installation and Configuration 12
Planning the Resource Group Topology for HA for TimesTen 12
TimesTen and Solaris Containers 13
HA for TimesTen and In-Memory Database Cache 13
Configuration Restrictions 13
Configuration Requirements 14
Installing and Configuring TimesTen 15
▼ How to Enable a TimesTen Database to Run in a Global Zone Configuration 15
▼ How to Install and Configure TimesTen 17
Verifying the Installation and Configuration of TimesTen 18
▼ How to Verify Installation and Configuration of TimesTen 19
Installing the HA for TimesTen Package 19
▼ How to Install the HA for TimesTen Package 19
Registering and Configuring HA for TimesTen 20
Specifying Extension Properties for the TimesTen Resource 20
Creating and Enabling Resources for TimesTen 21
Verifying the HA for TimesTen Installation and Configuration 25
▼ How to Verify the HA for TimesTen Installation for Failover Configurations 26
▼ How to Verify the HA for TimesTen Installation for Scalable or Multiple-Master Configurations 26
Define the TimesTen Replication 27
Tuning the HA for TimesTen Fault Monitor 27
Operation of the Fault Monitor for `ORCL.TimesTen_server` Resource Type 27

Operation of the Fault Monitor for the Other HA for TimesTen Resource Types 28
Debugging HA for TimesTen 28
▼ How to Activate Debugging for HA for TimesTen 28

A HA for TimesTen Extension Properties 31
`ORCL.TimesTen_active` Extension Properties 31
`ORCL.TimesTen_clag` Extension Properties 32
Manually Changeable Extension Properties 32
Extension Properties Changeable by TimesTen Utilities 33
`ORCL.TimesTen_cldaemon` Extension Properties 34
`ORCL.TimesTen_dbmon` Extension Properties 34
`ORCL.TimesTen_server` Extension Properties 34
`ORCL.TimesTen_standby` Extension Properties 36

B Deployment Example: Installing HA for TimesTen in a Failover Configuration 37
Target Cluster Configuration 37
Software Configuration 37
Assumptions 38
Installing and Configuring HA for TimesTen on Shared Storage 38
▼ Example: Configuring Cluster Resources for TimesTen 38
▼ Example: Preparing the Cluster for TimesTen 39
▼ Example: Installing TimesTen Software on Shared Storage 39
▼ Example: Enabling TimesTen Software to Run in the Cluster 41

C Deployment Example: Installing HA for TimesTen in an Active-Active Configuration 43
Target Cluster Configuration 43
Software Configuration 43
Assumptions 44
Installing and Configuring HA for TimesTen on Local Storage 44
▼ Example: Preparing the Cluster for HA for TimesTen 44
▼ Example: Installing TimesTen Software on Local Storage 45
▼ Example: Enabling TimesTen Software to Run in the Cluster 46
▼ Example: Defining TimesTen Replication in an Active-Active Configuration 47

D Deployment Example: Installing HA for TimesTen in a Scalable Subscriber Configuration ... 49
Target Cluster Configuration ... 49
Software Configuration ... 49
Assumptions ... 50
Installing and Configuring HA for TimesTen on Local Storage in a Scalable Subscriber Configuration ... 50
▼ Example: Preparing the Cluster for HA for TimesTen ... 50
▼ Example: Installing TimesTen Software on Local Storage ... 51
▼ Example: Enabling TimesTen Software to Run in the Cluster ... 52
▼ Example: Defining TimesTen Replication in a Scalable Subscriber Configuration ... 53

Index ... 55

Preface

Oracle Solaris Cluster Data Service for Oracle TimesTen In-Memory Database Guide explains how to install and configure Oracle Solaris Cluster data services.

Note – This Oracle Solaris Cluster release supports systems that use the SPARC and x86 families of processor architectures. In this document, "x86" refers to the larger family of x86 compatible products. Information in this document pertains to all platforms unless otherwise specified.

This document is intended for system administrators with extensive knowledge of Oracle software and hardware. Do not use this document as a planning or presales guide. Before reading this document, you should have already determined your system requirements and purchased the appropriate equipment and software.

The instructions in this book assume knowledge of the Oracle Solaris Operating System and expertise with the volume-manager software that is used with Oracle Solaris Cluster software.

Bash is the default shell for Oracle Solaris 11. Machine names shown with the Bash shell prompt are displayed for clarity.

Using UNIX Commands

This document contains information about commands that are specific to installing and configuring Oracle Solaris Cluster data services. The document does *not* contain comprehensive information about basic UNIX commands and procedures, such as shutting down the system, booting the system, and configuring devices. Information about basic UNIX commands and procedures is available from the following sources:

- Online documentation for the Oracle Solaris Operating System
- Oracle Solaris Operating System man pages
- Other software documentation that you received with your system

Typographic Conventions

The following table describes the typographic conventions that are used in this book.

TABLE P–1 Typographic Conventions

Typeface	Description	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`
`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`** `Password:`
aabbcc123	Placeholder: replace with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the *User's Guide*. A *cache* is a copy that is stored locally. Do *not* save the file. **Note:** Some emphasized items appear bold online.

Shell Prompts in Command Examples

The following table shows UNIX system prompts and superuser prompts for shells that are included in the Oracle Solaris OS. In command examples, the shell prompt indicates whether the command should be executed by a regular user or a user with privileges.

TABLE P–2 Shell Prompts

Shell	Prompt
Bash shell, Korn shell, and Bourne shell	`$`
Bash shell, Korn shell, and Bourne shell for superuser	`#`
C shell	`machine_name%`
C shell for superuser	`machine_name#`

Related Documentation

Information about related Oracle Solaris Cluster topics is available in the documentation that is listed in the following table. All Oracle Solaris Cluster documentation is available at http://www.oracle.com/technetwork/indexes/documentation/index.html.

Topic	Documentation
Hardware installation and administration	*Oracle Solaris Cluster 4.1 Hardware Administration Manual* Individual hardware administration guides
Concepts	*Oracle Solaris Cluster Concepts Guide*
Software installation	*Oracle Solaris Cluster Software Installation Guide*
Data service installation and administration	*Oracle Solaris Cluster Data Services Planning and Administration Guide* and individual data service guides
Data service development	*Oracle Solaris Cluster Data Services Developer's Guide*
System administration	*Oracle Solaris Cluster System Administration Guide* *Oracle Solaris Cluster Quick Reference*
Software upgrade	*Oracle Solaris Cluster Upgrade Guide*
Error messages	*Oracle Solaris Cluster Error Messages Guide*
Command and function references	*Oracle Solaris Cluster Reference Manual* *Oracle Solaris Cluster Data Services Reference Manual* *Oracle Solaris Cluster Geographic Edition Reference Manual* *Oracle Solaris Cluster Quorum Server Reference Manual*
Compatible software	Oracle Solaris Cluster Compatibility Guide available at the Oracle Solaris Cluster Technical Resources page

Access to Oracle Support

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

Getting Help

If you have problems installing or using Oracle Solaris Cluster, contact your service provider and provide the following information.

- Your name and email address (if available)
- Your company name, address, and phone number
- The model number and serial number of your systems
- The release number of the operating environment (for example, Oracle Solaris 11)
- The release number of Oracle Solaris Cluster (for example, Oracle Solaris Cluster 4.1)

Use the following commands to gather information about your system for your service provider.

Command	Function
`prtconf -v`	Displays the size of the system memory and reports information about peripheral devices
`psrinfo -v`	Displays information about processors
`pkg list`	Reports which packages are installed
`prtdiag -v`	Displays system diagnostic information
`/usr/cluster/bin/clnode show-rev -v`	Displays Oracle Solaris Cluster release and package version information for each node

Also have available the contents of the `/var/adm/messages` file.

◆ ◆ ◆ CHAPTER 1

Installing and Configuring HA for TimesTen

This chapter explains how to install and configure HA for TimesTen and contains the following sections:

- "HA for TimesTen Overview" on page 11
- "Overview of Installing and Configuring HA for TimesTen" on page 12
- "Planning the HA for TimesTen Installation and Configuration" on page 12
- "Installing and Configuring TimesTen" on page 15
- "Verifying the Installation and Configuration of TimesTen" on page 18
- "Installing the HA for TimesTen Package" on page 19
- "Registering and Configuring HA for TimesTen" on page 20
- "Verifying the HA for TimesTen Installation and Configuration" on page 25
- "Tuning the HA for TimesTen Fault Monitor" on page 27
- "Debugging HA for TimesTen" on page 28

HA for TimesTen Overview

Oracle Solaris Cluster HA for Oracle TimesTen In-Memory Database (HA for TimesTen) enables Oracle Solaris Cluster software to manage Oracle TimesTen In-Memory Database (TimesTen) by providing components to perform the orderly startup, shutdown, and fault monitoring of TimesTen.

You can configure HA for TimesTen as a failover, scalable, or multiple-master service. The type of deployment you configure depends of the deployment of the TimesTen data stores.

When a TimesTen database cluster is managed by the HA for TimesTen data service, the TimesTen instance becomes a failover, scalable, or multiple-master TimesTen resource across the cluster nodes. The control of the TimesTen instance is managed by the HA for TimesTen data service.

For conceptual information about failover data services, multiple-masters data services, and scalable data services, see *Oracle Solaris Cluster Concepts Guide*.

Overview of Installing and Configuring HA for TimesTen

The following table summarizes the tasks for installing and configuring HA for TimesTen and provides cross-references to detailed instructions for performing these tasks. Perform the tasks in the order that they are listed in the table.

TABLE 1–1 Tasks for Installing and Configuring HA for TimesTen

Task	Instructions
Plan the installation	"Planning the HA for TimesTen Installation and Configuration" on page 12
Install and configure the TimesTen software	"Installing and Configuring TimesTen" on page 15
Verify the installation and configuration	"How to Verify Installation and Configuration of TimesTen" on page 19
Install HA for TimesTen packages	"Installing the HA for TimesTen Package" on page 19
Register and configure HA for TimesTen resources	"Registering and Configuring HA for TimesTen" on page 20
Verify the HA for TimesTen installation and configuration	"Verifying the HA for TimesTen Installation and Configuration" on page 25
Tune the HA for TimesTen fault monitor	"Tuning the HA for TimesTen Fault Monitor" on page 27
Debug HA for TimesTen	"Debugging HA for TimesTen" on page 28

Planning the HA for TimesTen Installation and Configuration

This section contains the information you need to plan your HA for TimesTen installation and configuration.

Planning the Resource Group Topology for HA for TimesTen

TimesTen can be configured in the following ways, classified by the TimesTen replication configuration.

TABLE 1–2 TimesTen and Resource Group Topology

Replication Type	Resource group topology
No replication	Failover resource group

TABLE 1–2 TimesTen and Resource Group Topology *(Continued)*

Replication Type	Resource group topology
Master and one subscriber	Two failover resource groups
Master and multiple subscribers	One failover resource group plus one scalable or multiple-master resource group
Active-active	One multiple-master resource group for TimesTen plus one failover resource group for a logical host and potential applications.
Active-standby	One multiple-master resource group for the managing the server plus two failover resource groups for the active and the standby service

TimesTen and Solaris Containers

HA for TimesTen is supported in Solaris Containers, also called Solaris non-global zones. Oracle Solaris Cluster offers the following concepts for Solaris Containers:

- A zone cluster is a Solaris container of brand `cluster` that is created by using the `clzonecluster` command. A zone cluster forms a complete and separate cluster across the global-cluster nodes.
- An HA container is a non-global zone that is managed by the HA-Solaris Containers agent, and is represented by a resource of a resource group.

HA for TimesTen is configurable in all types of Solaris containers except an HA container.

HA for TimesTen and In-Memory Database Cache

TimesTen offers the option to configure data stores as in-memory database caches. This is a data store property and it is transparent to the HA for TimesTen agent. The underlying Oracle database can be made highly available in the same cluster as the in-memory database caches or in a different cluster.

Configuration Restrictions

The following configuration restrictions apply to HA for TimesTen.

Caution – Your data service configuration might not be supported if you do not observe these restrictions.

- **Combining data stores** - The TimesTen database can manage multiple data stores with a single instance. However, you can only mix the data-store topology if the data stores match your resource-group topology. For example, if one data store requires a failover resource and the other requires a multiple-master resource, you cannot mix the data stores in one single TimesTen server resource.
- **Automatic startup** - The TimesTen software installation provides an option to start the TimesTen server on system reboot. In an HA for TimesTen configuration, do *not* configure this option. HA for TimesTen controls the TimesTen server startup.
- **Recovery actions** - The TimesTen server resource type does not perform TimesTen internal error recovery actions, such as the reduplication of data stores. Only an active-standby configuration uses this type of recovery. For all other configurations, these type of actions are manual procedures.
- **Node lists in an active-standby configuration** - All resource groups that are part of an active-standby configuration must share the same node list.

Configuration Requirements

The following are configuration requirements for HA for TimesTen.

Caution – If your data service configuration does not conform to these requirements, the data service configuration might not be supported.

- **Dependencies between resource types** – The dependencies between the HA for TimesTen resource types are described in the following table.

Resource Type	Dependency
`ORCL.TimesTen_server`	`SUNW.HAStoragePlus` is required if the configuration uses a highly available local file system.
	`SUNW.Logical_Hostname` is required for a failover resource.
	`SUNW.Shared_Address` is required for a scalable subscriber.
`ORCL.TimesTen_cldaemon`	An `ORCL.TimesTen_clag` resource is required in an active-standby configuration.
`ORCL.TimesTen_dbmon`	An `ORCL.TimesTen_cldaemon` resource is required in an active-standby configuration.
`ORCL.TimesTen_active`	An `ORCL.TimesTen_dbmon` resource is required in an active-standby configuration.
`ORCL.TimesTen_standby`	An `ORCL.TimesTen_dbmon` resource is required in an active-standby configuration.

You set these dependencies when you register and configure HA for TimesTen. For more information, see "Registering and Configuring HA for TimesTen" on page 20.

If more elaborate dependencies are required, see the `r_properties(5)` and `rg_properties(5)` man pages for further dependencies and affinities settings.

- **Replication policies and cache policies** – Except for active-standby configurations, in all configurations that use replication and in-memory database cache, the startup policy for the replication and the cache *must* be set to `always`.

Installing and Configuring TimesTen

This section explains only the special requirements for installing TimesTen for use with HA for TimesTen. For complete information about installing and configuring TimesTen, see TimesTen documentation (`http://www.oracle.com/technetwork/database/timesten/documentation/index.html`). For complete information about installing and configuring a Solaris Container, see *Oracle Solaris 11.1 Administration: Oracle Solaris Zones, Oracle Solaris 10 Zones, and Resource Management*.

To determine which TimesTen version is installed, run the following commands.

```
# su - non-root-user
$ ttIsql --version
```

For each TimesTen instance that you are installing and configuring, choose the following tasks depending on whether you will configure HA for TimesTen to run in a global zone or in a non-global zone:

- To install and configure TimesTen in a global zone configuration, complete the following tasks:
  - "How to Enable a TimesTen Database to Run in a Global Zone Configuration" on page 15
  - " How to Install and Configure TimesTen" on page 17
- To install and configure TimesTen in a non-global zone configuration, complete the following tasks:
  - " How to Install and Configure TimesTen" on page 17

▼ How to Enable a TimesTen Database to Run in a Global Zone Configuration

This procedure creates the cluster infrastructure, such as resource groups, storage resources, and IP resources, according to your TimesTen configuration. For complete deployment examples, see Appendix B, "Deployment Example: Installing HA for TimesTen in a Failover

Configuration," Appendix C, "Deployment Example: Installing HA for TimesTen in an Active-Active Configuration," or Appendix D, "Deployment Example: Installing HA for TimesTen in a Scalable Subscriber Configuration."

Note – If you are deploying an active-standby configuration, do not perform this procedure. Instead, install TimesTen software and proceed to "How to Create and Enable Resources for TimesTen in an Active-Standby Configuration" on page 24.

---

1 (Optional) Assume the `root` role and register the `SUNW.HAStoragePlus` resource type.

```
# clresourcetype register SUNW.HAStoragePlus
```

2 Create a failover or multiple-master resource group.

- **For a failover resource group, perform the following command:**

  ```
  # clresourcegroup create TimesTen-resource-group
  ```

- **For a multiple-master resource group, perform the following command:**

  ```
  # clresourcegroup create -p maximum_primaries=2 \
  -p desired_primaries=2 \
  TimesTen-resource-group
  ```

3 (Optional) For a failover configuration, create a resource for the TimesTen disk storage.

```
# clresource create -t SUNW.HAStoragePlus \
-p FileSystemMountPoints=TimesTen-instance-mount-points \
TimesTen-has-resource
```

4 Create a logical-hostname or shared-address resource group.

- **To access the database from a logical host, perform the following command:**

  ```
  # clreslogicalhostname create -g TimesTen-resource-group \
  TimesTen-logical-hostname-resource-name
  ```

- **To access the database from a shared address, perform the following commands:**

  ```
  # clresourcegroup create TimesTen-access-group
  # clressharedaddress create -g TimesTen-access-group \
  TimesTen-shared-address-resource-name
  ```

5 Enable the resource groups.

```
# clresourcegroup online -eM TimesTen-resource-group
# clresourcegroup online -eM TimesTen-access-group
```

How to Install and Configure TimesTen

Note – For complete information about installing TimesTen, go to `http://www.oracle.com/technetwork/database/timesten/documentation/index.html`.

For complete deployment examples, see Appendix B, "Deployment Example: Installing HA for TimesTen in a Failover Configuration," Appendix C, "Deployment Example: Installing HA for TimesTen in an Active-Active Configuration," or Appendix D, "Deployment Example: Installing HA for TimesTen in a Scalable Subscriber Configuration."

Before You Begin Determine the following requirements for the deployment of TimesTen with Oracle Solaris Cluster:

- Verify that the TimesTen version you need is already installed on each cluster node. Search the most probable root paths where you find `bin/ttIsql`.

/your-path	Fully customized root path for TimesTen. This is where to place the binaries on the shared or local storage. A known convention is `/path/TimesTen/`*instancename*

- Determine the number of TimesTen instances to deploy.
- Determine the number of TimesTen data stores to deploy.
- Determine which local or cluster file systems will be used by each TimesTen resource.

The following assumptions are made:

- The TimesTen database software will be installed on shared or local storage in the directory *TimesTen* in the file system */tt*.
- The TimesTen database content will be installed in the same file system as the database software, in the directory */tt/data*.
- The home directory of the *timesten* user is */tt*.

1 Assume the `root` role and create the home directory for the TimesTen user on each node or non-global zone that runs the instance in parallel.

```
#  mkdir /tt
```

2 Add a group for TimesTen on every node or non-global zone.

```
# groupadd -g 1000 timesten
```

3. **Add a user who owns the TimesTen installation on every node or non-global zone.**

```
# useradd -u 1000 -g timesten -d /tt -s /usr/bin/ksh tt
# chown -R timesten:timesten /tt
```

4. **Create the `/etc/TimesTen` directory on every node or non-global zone.**

```
# mkdir /etc/TimesTen
# chgrp -R timesten /etc/TimesTen
# chmod -R 775 /etc/TimesTen
```

5. **Switch to the TimesTen user.**

```
# su - timesten
$
```

6. **If you are configuring a TimesTen failover service, set the LD_PRELOAD variable.**

```
$ export LD_PRELOAD_32=$LD_PRELOAD_32:/usr/cluster/lib/libschost.so.1
$ export LD_PRELOAD_64=$LD_PRELOAD_64:/usr/cluster/lib/64/libschost.so.1
$ export SC_LHOSTNAME=logical host ip alias
```

7. **(Optional) Add export statements for your LD_PRELOAD variables to your shells profile.**

8. **Install TimesTen software.**

```
$ cd your-timesten-install-dir
$ ./setup.sh
```

9. **Configure your data stores.**

   Add the following entries to your `sys.odbc.ini` file.

```
$ cat - > your-instance-dir/info/sys.odbc.ini
[test]
DataStore=/tt/data/test
PermSize=64
DatabaseCharacterSet=WE8MSWIN1252
```

10. **Create the directory to contain the data stores.**

```
$ mkdir /tt/data
```

Verifying the Installation and Configuration of TimesTen

Before you install the HA for TimesTen packages, verify that each TimesTen instance that you created is correctly configured to run in a cluster. The instance is the TimesTen database processes together with the associated data store processes. This verification does not confirm that the TimesTen databases are highly available, because the HA for TimesTen data service is not yet configured.

▼ How to Verify Installation and Configuration of TimesTen

Perform this procedure for each TimesTen instance that you created in "Installing and Configuring TimesTen" on page 15. During the verification, you complete the TimesTen postinstallation steps.

Before You Begin Determine whether you are in a local zone or in the global zone.

1. **Switch to the TimesTen user, if necessary.**

   ```
   # su - timesten
   ```

2. **Connect to your data store and exit.**

   ```
   $ ttIsql test -e "exit;"
   ```

3. **Stop the TimesTen server.**

   ```
   $ your-instance-dir/startup/tt_instancename stop
   ```

4. **If you are in a non-global zone, exit the zone to return to the global zone.**

Installing the HA for TimesTen Package

If you did not install the HA for TimesTen package during your initial Oracle Solaris Cluster installation, perform this procedure to install the package.

▼ How to Install the HA for TimesTen Package

Perform this procedure on each cluster node where you want the HA for TimesTen software to run.

1. **On the cluster node where you are installing the data service package, assume the `root` role.**

2. **Ensure that the `solaris` and `ha-cluster` publishers are valid.**

   ```
   # pkg publisher
   PUBLISHER                             TYPE     STATUS   URI
   solaris                               origin   online   solaris-repository
   ha-cluster                            origin   online   ha-cluster-repository
   ```

   For information about setting the `solaris` publisher, see "Set the Publisher Origin to the File Repository URI" in *Copying and Creating Oracle Solaris 11.1 Package Repositories*.

3. **Install the HA for TimesTen software package.**

   ```
   # pkg install ha-cluster/data-service/timesten
   ```

4 **Verify that the package installed successfully.**

```
$ pkg info ha-cluster/data-service/timesten
```

Installation is successful if output shows that `State` is `Installed`.

5 **Perform any necessary updates to the Oracle Solaris Cluster software.**

For instructions on updating single or multiple packages, see Chapter 11, "Updating Your Software," in *Oracle Solaris Cluster System Administration Guide*.

Registering and Configuring HA for TimesTen

Before you perform the procedures in this section, ensure that the HA for TimesTen data service packages are installed.

This section covers the following main topics:

- "Specifying Extension Properties for the TimesTen Resource" on page 20
- "How to Create and Enable Resources for TimesTen in a Failover Configuration" on page 21
- "How to Create and Enable Resources for TimesTen in a Multiple-Master Configuration" on page 22
- "How to Create and Enable Resources for TimesTen in a Scalable Configuration" on page 23
- "How to Create and Enable Resources for TimesTen in an Active-Standby Configuration" on page 24

Specifying Extension Properties for the TimesTen Resource

HA for TimesTen provides multiple resource types for configuring TimesTen in the various TimesTen topologies. Except for the `ORCL.TimesTen_clag` resource type, TimesTen allows you to specify extension properties for a resource type either at resource creation or at resource modification.

For the `ORCL.TimesTen_clag` resource type, some of its properties you modify with the `clresource` command. However, other properties you must *not* modify manually with the `clresource` command. Instead, you *must* use the TimesTen utility `ttCWadmin`.

For a complete description of resource types and properties, including requirements for modifying each property, see Appendix A, "HA for TimesTen Extension Properties."

Creating and Enabling Resources for TimesTen

▼ How to Create and Enable Resources for TimesTen in a Failover Configuration

Before You Begin Ensure that the following prerequisites are met.

- The TimesTen software is installed on shared storage.
- The TimesTen data store is configured.
- The required resource group, storage resource, and logical-hostname resource are created.
- The resources in the failover resource group are enabled and the resource group is managed and online.

1 Assume the `root` role on a node in the cluster that will host TimesTen.

2 Register the resource type for the TimesTen server resource.

```
# clresourcetype register ORCL.TimesTen_server
```

3 Create the resource for the TimesTen resource group.

```
# clresource create -g tt-resource-group \
-t TimesTen_server \
-p Base_directory=TimesTen-base-directory \
-p Instance=TimesTen-instance-name \
-p Datastore=TimesTen-data-store-name \
-p Table=TimesTen-table-name \
-p Hostname=TimesTen-logical-host-name \
-p Resource_dependencies=logical-host-name \
-p Resource_dependencies_offline_restart=hastorageplus-resource-name \
TimesTen-resource-name
```

Note – The resource is created in the enabled state.

–g *resource-group*
Specifies the resource group name into which the resource is to be placed

–t ORCL.TimesTen_server
Specifies the resource type for the TimesTen resource

-p Base_directory=*TimesTen-base-directory*
Specifies the directory where TimesTen instances are to be installed

-p Instance=*TimesTen-instance-name*
Specifies the TimesTen instance name

-p Datastore=*TimesTen-data-store-name*
Specifies the TimesTen data-store name that will be used for server monitoring

`-p Table=`*TimesTen-table-name*
Specifies the table name that is about to be manipulated in the data store for monitoring purpose

`-p Hostname=`*TimesTen-logical-hostname*
Specifies the IP alias name of the logical-hostname resource where the hostname must point to for TimesTen failover configurations

`-p Resource_dependencies=`*logical-hostname*
Specifies the list of Oracle Solaris Cluster resources on which TimesTen depends

`-p Resource_dependencies_offline_restart=`*hastorageplus-resource-name*
Sets a resource dependency between *tt-resource-group* and the `HAStoragePlus` resource you created in Step 3

Note – The dependency list must include the logical-hostname resource and the storage resource for failover configurations.

▼ How to Create and Enable Resources for TimesTen in a Multiple-Master Configuration

Before You Begin Ensure that the following prerequisites are met.

- The TimesTen software is installed on all participating nodes.
- The TimesTen data store is configured.
- The required resource group is created.
- The resources in the failover resource group are enabled and the resource group is managed.

1 Assume the `root` role on a node in the cluster that will host TimesTen.

2 Register the resource type for the TimesTen server resource.

```
# clresourcetype register ORCL.TimesTen_server
```

3 Create the resource for the TimesTen resource group.

```
# clresource create -g tt-resource-group \
-t TimesTen_server \
-p Base_directory=TimesTen-base-directory \
-p Instance=TimesTen-instance-name \
-p Datastore=TimesTen-data-store-name \
-p Table=TimesTen-table-name \
TimesTen-resource-name
```

Note – The resource is created in the enabled state.

–g *resource-group*
Specifies the resource group name into which the resource is to be placed

–t ORCL.TimesTen_server
Specifies the resource type for the TimesTen resource

-p Base_directory=*TimesTen-base-directory*
Specifies the directory where TimesTen instances are to be installed

-p Instance=*TimesTen-instance-name*
Specifies the TimesTen instance name

-p Datastore=*TimesTen-data-store-name*
Specifies the TimesTen data-store name that will be used for server monitoring

-p Table=*TimesTen-table-name*
Specifies the table name that is about to be manipulated in the data store for monitoring purpose

▼ How to Create and Enable Resources for TimesTen in a Scalable Configuration

Before You Begin Ensure that the following prerequisites are met.

- The TimesTen software is installed on all participating nodes.
- The TimesTen data store is configured.
- The required TimesTen resource group, the failover resource group, and the required shared address resource are created.
- The shared address resource in the failover resource group is enabled and all resource groups are managed.

1 Assume the `root` role on a node in the cluster that will host TimesTen.

2 Register the resource type for the TimesTen server resource.

```
# clresourcetype register ORCL.TimesTen_server
```

3 Create the resource for the TimesTen resource group.

```
# clresource create -g tt-resource-group \
-t TimesTen_server \
-p Scalable=True \
-p Port_List=TimesTen-server-port/tcp \
-p Base_directory=TimesTen-base-directory \
-p Instance=TimesTen-instance-name \
-p Datastore=TimesTen-data-store-name \
-p Table=TimesTen-table-name \
-p Resource_dependencies=shared-address-resource-name \
-p Resource_dependencies_offline_restart=hastorageplus-resource-name \
TimesTen-resource-name
```

Note – The resource is created in the enabled state.

`–g` *resource-group*
Specifies the resource group name into which the resource is to be placed

`–t ORCL.TimesTen_server`
Specifies the resource type for the TimesTen resource

`–p Scalable=true`
Specifies that the resource is a scalable resource

`-p Port_list=`*TimesTen-base-directory*
Specifies the port for incoming sql connections on which the TimesTenserver listens

`-p Base_directory=`*TimesTen-base-directory*
Specifies the directory where TimesTen instances are to be installed

`-p Instance=`*TimesTen-instance-name*
Specifies the TimesTen instance name

`-p Datastore=`*TimesTen-data-store-name*
Specifies the TimesTen data-store name that will be used for server monitoring

`-p Table=`*TimesTen-table-name*
Specifies the table name that is about to be manipulated in the data store for monitoring purpose

`-p Resource_dependencies=`*logical-hostname*
Specifies the list of Oracle Solaris Cluster resources on which TimesTen depends

`-p Resource_dependencies_offline_restart=`*hastorageplus-resource-name*
Sets a resource dependency between `tt-resource-group` and the `HAStoragePlus` resource you created.

Note – The dependency list must include the shared-address resource.

▼ How to Create and Enable Resources for TimesTen in an Active-Standby Configuration

Before You Begin Ensure that the following prerequisites are met.

- The TimesTen software is installed on all participating nodes.
- The TimesTen data stores are configured.

1 Assume the `root` role on a node in the cluster that will host TimesTen.

2 **Create the multiple-master resource group,**

This resource group contains the HA for TimesTen daemon resource together with the HA for TimesTen cluster agent and the TimesTen database monitor resources.

```
# clresourcegroup create -p maximum_primaries=2 \
-p desired_primaries=2  \
-n nodelist \
server-resource-group-name
```

3 **Create the failover resource group to contain the active service resource.**

```
# clresourcegroup create  \
-n nodelist \
active-resource-group-name
```

Note – Ensure that this node list is identical to the node list for the multiple-master resource group.

---

4 **Create the failover resource group to contain the standby service resource.**

```
# clresourcegroup create  \
-n nodelist \
standby-resource-group-name
```

Note – Ensure that this node list is identical to the node list for the multiple-master resource group.

---

5 **Ensure that all resource groups are in the managed online state.**

```
# clresourcegroup online -M server-resource-group-name
# clresourcegroup online -M active-resource-group-name
# clresourcegroup online -M standby-resource-group-name
```

6 **Proceed with the TimesTen configuration, using the TimesTen utility `ttCWadmin`.**

Follow procedures in your TimesTen documentation.

Verifying the HA for TimesTen Installation and Configuration

After you install, register, and configure HA for TimesTen, verify this installation and configuration to determine whether the HA for TimesTen data service makes your TimesTen database highly available.

▼ How to Verify the HA for TimesTen Installation for Failover Configurations

1. **Assume the `root` role on a cluster node that is to host the TimesTen component.**

2. **Ensure that all the TimesTen resources are online.**
   For each resource, perform the following steps:

   a. **Determine whether the resource is online.**
      `# clresource status` *TimesTen-rs*

   b. **If the resource is not online, bring the resource online.**
      `# clresource enable` *TimesTen-rs*

3. **Switch the resource group to another cluster node, such as *node2*.**
   `# clresourcegroup switch -h` *node2 TimesTen-rg*

4. **Confirm that the resource is now online on *node2*.**
   `# clresource status` *TimesTen-rs*

▼ How to Verify the HA for TimesTen Installation for Scalable or Multiple-Master Configurations

1. **Become superuser on a cluster node that is to host the TimesTen component.**

2. **Ensure that all TimesTen resources are online.**
   Perform these steps for each resource.

   a. **Determine whether the resource is online.**
      `# clresource status` *TimesTen-rs*

   b. **If the resource is not online, bring the resource online.**
      `# clresource enable` *TimesTen-rs*

   c. **If the resource is online, bring the resource offline, and online again.**
      `# clresource disable` *TimesTen-rs*
      `# clresource enable` *TimesTen-rs*

3. **Confirm that the resource is now online on all nodes in the node list.**
   `# clresource status` *TimesTen-rs*

Define the TimesTen Replication

For active-active configurations and master and subscriber configurations, configure your replication between data stores after you have verified the HA for TimesTen configuration. For further information, consult the TimesTen documentation (`http://www.oracle.com/technetwork/database/timesten/documentation/index.html`).

Tuning the HA for TimesTen Fault Monitor

The HA for TimesTen fault monitor verifies that the data service is running in a healthy condition.

An HA for TimesTen fault monitor is contained in each resource that represents a TimesTen instance or that manages a separate TimesTen daemon. You created these resources when you registered and configured HA for TimesTen. For more information, see "Registering and Configuring HA for TimesTen" on page 20.

System properties and extension properties of the TimesTen resources control the behavior of the fault monitor. The default values of these properties determine the preset behavior of the fault monitor. Because the preset behavior should be suitable for most Oracle Solaris Cluster installations, tune the HA for TimesTen fault monitor *only* if you need to modify this preset behavior.

Tuning the HA for TimesTen fault monitor involves the following tasks:

- Setting the interval between fault monitor probes
- Setting the timeout for fault monitor probes
- Defining the criteria for persistent faults
- Specifying the failover behavior of a resource

To achieve these behaviors, consider the standard resource properties `retry_interval` and `thorough_probe_interval`.

For more information, see "Tuning Fault Monitors for Oracle Solaris Cluster Data Services" in *Oracle Solaris Cluster Data Services Planning and Administration Guide*.

Operation of the Fault Monitor for ORCL.TimesTen_server Resource Type

The fault monitor for HA for TimesTen ensures that all the requirements for the `ORCL.TimesTen_server` resource type to run are met in one of the following ways:

- The `ORCL.TimesTen_server` resource type answers on http requests.

If the http request is not answered, the fault monitor restarts the TimesTen database server. If the fault persists, the fault monitor fails over the resource group that contains the resource for TimesTen.

- Connections to the TimesTen database server are possible, the database catalog is accessible, and a test table can be manipulated.

  If the any of the above fails, the fault monitor triggers a restart or a failover of the TimesTen database-server resource.

Operation of the Fault Monitor for the Other HA for TimesTen Resource Types

The fault monitor for HA for TimesTen resource types that are used for active-standby configurations ensures that all the requirements for the TimesTen resource types to run are met.

- Each command to start one of the required daemons has a check option. The fault monitor calls this command with the check option. If this call is unsuccessful, the fault monitor triggers a restart or a failover of the TimesTen resource.

Debugging HA for TimesTen

HA for TimesTen resource types have an extension property, `debug_level`, that enables you to activate debugging for TimesTen resources.

▼ How to Activate Debugging for HA for TimesTen

Perform this procedure to activate debugging.

Note – To deactivate debugging, repeat this procedure with the following changes:

- Change `daemon.debug` to `daemon.notice`.
- Change the `debug_level` property to `0`.

---

1 Determine whether you are in the global zone or in a non-global zone configuration.

If your TimesTen resource group node list contains non-global zones, you are in a non–global zone configuration. In any other case, you are in a global zone configuration.

2 Determine whether debug logging for HA for TimesTen is active in your node or zone.

```
# grep daemon /etc/syslog.conf
*.err;kern.debug;daemon.notice;mail.crit        /var/adm/messages
```

```
*.alert;kern.err;daemon.err                     operator
#
```

If debug logging is inactive, `daemon.notice` is set in the file `/etc/syslog.conf` of the appropriate zone.

3 If debug logging is inactive, edit the `/etc/syslog.conf` file in the appropriate node or zone to change `daemon.notice` to `daemon.debug`.

4 Confirm that debug logging for HA for TimesTen is active.

If debugging is active, `daemon.debug` is set in the file `/etc/syslog.conf`.

```
# grep daemon /etc/syslog.conf
*.err;kern.debug;daemon.debug;mail.crit        /var/adm/messages
*.alert;kern.err;daemon.err                    operator
#
```

5 Restart the `syslogd` daemon in the appropriate node or zone.

```
# svcadm refresh svc:/system/system-log:default
```

6 Set the `debug_level` property to either `1` or `2`.

```
# clresource set -p debug_level=intended value
```

Note – Setting the `debug_level` property to 2 will result in a Korn shell trace of the validate command.

◆ ◆ ◆ APPENDIX A

HA for TimesTen Extension Properties

The extension properties that you can set for each HA for TimesTen resource type are listed in the following sections:

- "ORCL.TimesTen_active Extension Properties" on page 31
- "ORCL.TimesTen_clag Extension Properties" on page 32
- "ORCL.TimesTen_cldaemon Extension Properties" on page 34
- "ORCL.TimesTen_dbmon Extension Properties" on page 34
- "ORCL.TimesTen_server Extension Properties" on page 34
- "ORCL.TimesTen_standby Extension Properties" on page 36

See the r_properties(5) and rg_properties(5) man pages for details about all of the system-defined properties.

ORCL.TimesTen_active Extension Properties

`Daemon_home_directory` (String)
Specifies the TimesTen directory that contains the `sys.odbc.ini`.

Default: None

Tunable: When disabled

Node dependent: Yes

`Data_store` (String)
Specifies a TimesTen data store used to identify the TimesTen active service.

Default: None

Tunable: When disabled

`Debug_level` (Integer)
Specifies a debug flag to enable and control debugging.

Default: 0

Maximum value: 2

Tunable: Any time

ORCL.TimesTen_clag Extension Properties

The resource type `ORCL.TimesTen_clag` has two sets of extension properties.

- Extension properties that are validated by the validate method of the resource type. You can manually change this set of extension properties by using the `clresource set` command.
- Extension properties that are validated only by the TimesTen utility `ttCWadmin`. Do *not* use the `clresource` command to change this set of extension properties.

The following sections describe both sets of `ORCL.TimesTen_clag` extension properties:

- "Manually Changeable Extension Properties" on page 32
- "Extension Properties Changeable by TimesTen Utilities" on page 33

Manually Changeable Extension Properties

The following extension properties can be changed manually by using the `clresource` command.

`Base_directory` (String)
Specifies the directory where TimesTen instances are installed.

Default: None

Range Minimum: 1

Tunable: When disabled

`Debug_level` (Integer)
Specifies a debug flag to enable and control debugging.

Default: 0

Maximum value: 2

Tunable: Any time

`Envscript` (String)
Specifies a Korn shell script which sets environment variables for the TimesTen application.

Default: None

Tunable: Any time

`Instance` (String)
Specifies the TimesTen instance name.

Default: None

Range Minimum: 1

Tunable: When disabled

Extension Properties Changeable by TimesTen Utilities

The following `ORCL.TimesTen_clag` extension properties must be changed only by using the TimesTen utility `ttCWadmin`:

- `AUTORECOVER`
- `BACKUPC`
- `BACKUPDIR`
- `BACKUPP`
- `CACHE`
- `CACHEPWD`
- `CACHEUSER`
- `CHARACTERSET`
- `CLUSTERTYPE`
- `DSN`
- `GRIDIP`
- `GRIDIPCONF`
- `GRIDPORT`
- `HALEVEL`
- `HOSTLIST`
- `HOSTS`
- `NUMAPPS`
- `NUMSUBHOSTS`
- `ORANETSERVICENAME`
- `ORAPWD`
- `ORAUSER`
- `PPHRASE`
- `REPCHECKSUM`
- `REPDDL`
- `STDBY_SCRIPTTIMEOUT`
- `SUBSCRIBERHOSTS`

ORCL.TimesTen_cldaemon Extension Properties

`Debug_level` (Integer)
Specifies a debug flag to enable and control debugging.

Default: 0

Maximum value: 2

Tunable: Any time

ORCL.TimesTen_dbmon Extension Properties

`Daemon_home_directory` (String)
Specifies the TimesTen directory that contains the `sys.odbc.ini`.

Default: None

Tunable: When disabled

Node dependent: Yes

`Data_store` (String)
Specifies a TimesTen data store used to identify the TimesTen database monitor.

Default: None

Tunable: When disabled

`Debug_level` (Integer)
Specifies a debug flag to enable and control debugging.

Default: 0

Maximum value: 2

Tunable: Any time

ORCL.TimesTen_server Extension Properties

`Base_directory` (String)
Specifies the directory where TimesTen instances are installed.

Default: None

Range Minimum: 1

Tunable: When disabled

`Daemon_home_directory` (String)
Specifies the TimesTen directory that contains the `sys.odbc.ini`.

Default: None

Tunable: When disabled

Node dependent: Yes

`Datastore` (String)
Specifies a TimesTen data store that is used to monitor the TimesTen server.

Default: None

Tunable: When disabled

`Debug_level` (Integer)
Specifies a debug flag to enable and control debugging.

Default: 0

Maximum value: 2

Tunable: Any time

`Envscript` (String)
Specifies a Korn shell script that sets environment variables for the TimesTen application.

Default: None

Tunable: Any time

`Hostname` (String)
Specifies an IP alias to point to an IP alias that is managed by a logical host. This IP alias works as the hostname for TimesTen in failover configurations.

Default: None

Tunable: When disabled

`Instance` (String)
Specifies the TimesTen instance name.

Default: None

Range Minimum: 1

Tunable: When disabled

`Table` (String)
Specifies a TimesTen table that is used to monitor the TimesTen server.

Default: None

Tunable: When disabled

ORCL.TimesTen_standby Extension Properties

`Daemon_home_directory` (String)
Specifies the TimesTen directory that contains the `sys.odbc.ini`.

Default: None

Tunable: When disabled

Node dependent: Yes

`Data_store` (String)
Specifies a TimesTen data store used to identify the TimesTen standby service.

Default: None

Tunable: When disabled

`Debug_level` (Integer)
Specifies a debug flag to enable and control debugging.

Default: 0

Maximum value: 2

Tunable: Any time

◆ ◆ ◆ APPENDIX B

Deployment Example: Installing HA for TimesTen in a Failover Configuration

This appendix presents a complete example of how to install and configure the TimesTen application and data service in a failover configuration. It presents a simple two-node cluster configuration. If you need to install the TimesTen application in the global zone, perform all steps only in the global zone.

If you need to install the application in any other configuration, refer to the general-purpose procedures presented elsewhere in this manual. For an example of an HA for TimesTen installation in a scalable subscriber configuration, see Appendix D, "Deployment Example: Installing HA for TimesTen in a Scalable Subscriber Configuration." For an active-active configuration, see Appendix C, "Deployment Example: Installing HA for TimesTen in an Active-Active Configuration."

Target Cluster Configuration

This example uses a two-node cluster with the following node names:

- `phys-schost-1` (a physical node, that owns the file system)
- `phys-schost-2` (a physical node)

This configuration also uses the logical hostname `ha-host-1`.

Software Configuration

This deployment example uses the following software products and versions:

- Solaris 11 software for SPARC or x86 platforms
- Oracle Solaris Cluster 4.1 core software
- HA for TimesTen
- TimesTen software version 11.2.2 tar file
- Your preferred text editor

This example assumes that you have already installed and established your cluster. It illustrates installation and configuration of the data service application only.

Assumptions

The instructions in this example were developed with the following assumptions:

- **Shell environment:** All commands and the environment setup in this example are for the Korn shell environment. If you use a different shell, replace any Korn shell-specific information or instructions with the appropriate information for you preferred shell environment.
- **User login:** Unless otherwise specified, perform all procedures by assuming a role that provides `solaris.cluster.admin`, `solaris.cluster.modify`, and `solaris.cluster.read` RBAC authorization.

Installing and Configuring HA for TimesTen on Shared Storage

The tasks you must perform to install and configure HA for TimesTen in the global zone are as follows:

- "Example: Configuring Cluster Resources for TimesTen" on page 38
- "Example: Preparing the Cluster for TimesTen" on page 39
- "Example: Installing TimesTen Software on Shared Storage" on page 39
- "Example: Enabling TimesTen Software to Run in the Cluster" on page 41

▼ Example: Configuring Cluster Resources for TimesTen

1 Register the necessary resource types on both nodes.

```
phys-schost-1# clresourcetype register ORCL.TimesTen_server SUNW.HAStoragePlus
```

2 Create the TimesTen resource group `RG-TT`.

```
phys-schost-1# clresourcegroup create RG-TT
```

3 Create the logical host `ha-host-1`.

```
phys-schost-1# clreslogicalhostname create -g RG-TT ha-host-1
```

4 Create the `HAStoragePlus` resource in the `RG-TT` resource group.

```
phys-schost-1# clresource create -g RG-TT -t SUNW.HAStoragePlus -p AffinityOn=TRUE \
-p FilesystemMountPoints=/global/mnt3 TT-HSP-RS
```

5 Enable the `RG-TT` resource group.

```
phys-schost-1# clresourcegroup online -eM RG-TT
```

▼ Example: Preparing the Cluster for TimesTen

1 **Install and configure the cluster as instructed in *Oracle Solaris Cluster Software Installation Guide*.**

Install the following cluster software components on both nodes.

- Oracle Solaris Cluster core software
- HA for TimesTen data service software

2 **Beginning on the node that owns the file system, add the `timesten` user.**

```
phys-schost-1# groupadd -g 1000 timesten
phys-schost-1# useradd -g 1000 -d /global/mnt3/timesten -s /bin/ksh timesten

phys-schost-2# groupadd -g 1000 timesten
phys-schost-2# useradd -g 1000 -d /global/mnt3/timesten -s /bin/ksh timesten
```

▼ Example: Installing TimesTen Software on Shared Storage

These steps illustrate how to install TimesTen software on shared storage.

1 **Create the home directory for the TimesTen user.**

```
phys-schost-1# mkdir /global/mnt3/timesten
```

2 **Change the ownership of the `timesten` directory.**

```
phys-schost-1# chown -R timesten:timesten /global/mnt3/timesten
```

3 **Create the `/etc/TimesTen` directory and set the permission for the `timesten` user.**

```
phys-schost-1# mkdir /etc/TimesTen
phys-schost-1# chgrp -R timesten /etc/TimesTen
phys-schost-1# chmod 775 /etc/TimesTen

phys-schost-2# mkdir /etc/TimesTen
phys-schost-2# chgrp -R timesten /etc/TimesTen
phys-schost-2# chmod 775 /etc/TimesTen
```

4 **Log in as the TimesTen user.**

```
phys-schost-1# su - timesten
phys-schost-2# su - timesten
```

5 **Create the data directory on shared storage.**

```
phys-schost-1$ mkdir /global/mnt3/timesten/data
```

6 Create a `.profile` file in the TimesTen user's home directory.

Add the following lines to the `.profile` file to set the necessary environment for logical hostname `ha-host-1`.

```
phys-schost-1# vi .profile
export LD_PRELOAD_32=$LD_PRELOAD_32:/usr/cluster/lib/libschost.so.1
export LD_PRELOAD_64=$LD_PRELOAD_64:/usr/cluster/lib/64/libschost.so.1
export SC_LHOSTNAME=ha-host-1
```

7 Check the setting.

```
phys-schost-1# su - timesten
phys-schost-1$ hostname
```

8 Change to your software directory and install the TimesTen software.

```
phys-schost-1$ cd /global/mnt3/repository/
phys-schost-1$ tar xf timesten1122.tar
phys-schost-1$ cd ./solx8664
phys-schost-1$ ./setup.sh
```

Provide the following answers to the questions. Most prompts in this example accept the default.

```
Please choose an instance name for this installation? [ tt1122 ]
Instance name will be 'tt1122'. Is this correct? [ yes ]

   Of the three components:

[1] Client/Server and Data Manager
[2] Data Manager Only
[3] Client Only
Which would you like to install? [ 1 ]

 Of the following options :

[1] /export/tt
[2] /my-data/solx8664
[3] Specify a location
[q] Quit the installation

Where would you like to install the tt1122 instance of TimesTen? [ 1 ] 3
Please specify a directory to install TimesTen? [ /export/tt ] /global/mnt3/timesten
Where would you like to create
the daemon home directory? [ /global/mnt3/timesten/TimesTen/tt1122/info ]
Would you like to specify a different location for the daemon logs? [ no ]
Do you want to use the default port number for the TimesTen daemon? [ yes ]
Restrict access to the TimesTen installation to the group 'timesten'? [ yes ]
Please enter a value for TNS_ADMIN (s=skip)? [  ] s
What is the TCP/IP port number that you want
the TimesTen Server to listen n? [ 53389 ]
Do you want to install QuickStart and the TimesTen Documentation? [ no ]
Would you like to install the documentation (without QuickStart)? [ yes ]
Where would you like to create
the doc directory (s=skip)? [ /global/mnt3/TimesTen/tt1122/doc ]
```

9 **Adjust the PATH variable to contain the TimesTen binaries, for convenience.**

```
phys-schost-1$ PATH=$PATH:/global/mnt3/TimesTen/tt1122/bin
phys-schost-1$ export PATH
```

10 **Create a data store.**

Add the following lines to the `/global/mnt3/TimesTen/tt1122/info` `sys.odbc.ini` file.

```
[test]
DataStore=/global/mnt3/data/test
PermSize=64
DatabaseCharacterSet=WE8MSWIN1252
```

11 **Connect to the data store.**

```
phys-schost-1$ ttIsql -e "exit;" test
```

12 **Stop the TimesTen server.**

```
phys-schost-1$ /global/mnt3/TimesTen/tt1122/startup/tt_tt1122 stop
```

13 **Leave the TimesTen user and exit the non-global zone.**

▼ Example: Enabling TimesTen Software to Run in the Cluster

This procedure creates the TimesTen resource `TT-RS`.

- **Create the TimesTen server resource `TT-RS`.**

```
phys-host-1# clrs create -g RG-TT -t TimesTen_server \
-p Base_directory=/global/mnt3/TimesTen \
-p Instance=tt1122 \
-p datastore=test \
-p table=sctest \
-p Hostname=ha-host-1 \
-p resource_dependencies=ha-host-1 \
-p resource_dependencies_offline_restart=TT-HSP-RS TT-RS
```

◆ ◆ ◆ APPENDIX C

Deployment Example: Installing HA for TimesTen in an Active-Active Configuration

This appendix presents a complete example of how to install and configure the TimesTen application and data service in an active-active configuration. It presents a simple two-node cluster configuration. If you need to install the TimesTen application in the global zone, perform all steps only in the global zone.

If you need to install the application in any other configuration, refer to the general-purpose procedures presented elsewhere in this manual. For an example of TimesTen in a failover configuration, see Appendix B, "Deployment Example: Installing HA for TimesTen in a Failover Configuration," for scalable subscriber configuration see Appendix D, "Deployment Example: Installing HA for TimesTen in a Scalable Subscriber Configuration."

Target Cluster Configuration

This example uses a two-node cluster with the following node names:

- `phys-schost-1` (a physical node)
- `phys-schost-2` (a physical node)

Software Configuration

This deployment example uses the following software products and versions:

- Solaris 11 software for SPARC or x86 platforms
- Oracle Solaris Cluster 4.1 core software
- HA for TimesTen data service software
- TimesTen software version 11.2.2 tar file
- Your preferred text editor

This example assumes that you have already installed and established your cluster. It illustrates installation and configuration of the data service application only.

Assumptions

The instructions in this example were developed with the following assumptions:

- **Shell environment:** All commands and the environment setup in this example are for the Korn shell environment. If you use a different shell, replace any Korn shell-specific information or instructions with the appropriate information for you preferred shell environment.
- **User login:** Unless otherwise specified, perform all procedures by assuming a role that provides `solaris.cluster.admin`, `solaris.cluster.modify`, and `solaris.cluster.read` RBAC authorization.

Installing and Configuring HA for TimesTen on Local Storage

These instructions assume that you are installing TimesTen software as the `timesten` user in a local directory.

The tasks you must perform to install and configure TimesTen local storage are as follows:

- "Example: Preparing the Cluster for HA for TimesTen" on page 44
- "Example: Installing TimesTen Software on Local Storage" on page 45
- "Example: Enabling TimesTen Software to Run in the Cluster" on page 46
- "Example: Defining TimesTen Replication in an Active-Active Configuration" on page 47

▼ Example: Preparing the Cluster for HA for TimesTen

1 Install and configure the cluster as instructed in *Oracle Solaris Cluster Software Installation Guide*.

Install the following cluster software components on both nodes.

- Oracle Solaris Cluster core software
- HA for TimesTen data service software

2 Add the `timesten` group and user.

```
phys-schost-1# groupadd -g 1000 timesten
phys-schost-1# useradd -g 1000 -m -d /timesten -s /bin/ksh timesten

phys-schost-2# groupadd -g 1000 timesten
phys-schost-2# useradd -g 1000 -m -d /timesten -s /bin/ksh timesten
```

▼ Example: Installing TimesTen Software on Local Storage

This example illustrates how to install TimesTen software on local storage. Perform this procedure on both nodes.

1 Create the `/etc/TimesTen` directory and set the permission for the `timesten` user.

```
phys-schost-1# mkdir /etc/TimesTen
phys-schost-1# chgrp -R timesten /etc/TimesTen
phys-schost-1# chmod 775 /etc/TimesTen

phys-schost-2# mkdir /etc/TimesTen
phys-schost-2# chgrp -R timesten /etc/TimesTen
phys-schost-2# chmod 775 /etc/TimesTen
```

2 Log in as the TimesTen user.

```
phys-schost-1# su - timesten
phys-schost-2# su - timesten
```

3 Create the data directory `/timesten/data` on local storage.

```
phys-schost-1$ mkdir /timesten/data
phys-schost-2$ mkdir /timesten/data
```

4 Change to your software directory and install the TimesTen software on both nodes.

```
phys-schost-1$ cd /repository/
phys-schost-1$ tar xf timesten1122.tar
phys-schost-1$ cd ./solx8664
phys-schost-1$ ./setup.sh
```

Provide the following answers to the questions. Most prompts in this example accept the default.

```
Please choose an instance name for this installation? [ tt1122 ]
Instance name will be 'tt1122'. Is this correct? [ yes ]

   Of the three components:

[1] Client/Server and Data Manager
[2] Data Manager Only
[3] Client Only  Which would you like to install? [ 1 ]

 Of the following options :

[1] /export/tt
[2] /my-data/solx8664
[3] Specify a location
[q] Quit the installation

Where would you like to install the tt1122 instance of TimesTen? [ 1 ] 3
Please specify a directory to install TimesTen? [ /export/tt ] /timesten
Where would you like to create
the daemon home directory? [ /timesten/TimesTen/tt1122/info ]
```

```
Would you like to specify a different location for the daemon logs? [ no ]
Do you want to use the default port number for the TimesTen daemon? [ yes ]
Restrict access to the TimesTen installation to the group 'timesten'? [ yes ]
Please enter a value for TNS_ADMIN (s=skip)? [  ] s
What is the TCP/IP port number that you want
the TimesTen Server to listen n? [ 53389 ]
Do you want to install QuickStart and the TimesTen Documentation? [ no ]
Would you like to install the documentation (without QuickStart)? [ yes ]
Where would you like to create
the doc directory (s=skip)? [ /timesten/TimesTen/tt1122/doc ]
```

Repeat this step on the other node.

5 Adjust the PATH variable to contain the TimesTen binaries, for convenience.

```
phys-schost-1$ PATH=$PATH:/timesten/TimesTen/tt1122/bin
phys-schost-1$ export PATH
phys-schost-2$ PATH=$PATH:/timesten/TimesTen/tt1122/bin
phys-schost-2$ export PATH
```

6 Add the following lines to the `/timesten/TimesTen/tt1122/info/sys.odbc.ini` file on both nodes.

```
[test]
DataStore=/timesten/data/test
PermSize=64
DatabaseCharacterSet=WE8MSWIN1252
```

7 Connect to the data store.

```
phys-schost-1$ ttIsql -e "exit;" test
phys-schost-2$ ttIsql -e "exit;" test
```

8 Stop the TimesTen server on both nodes.

```
phys-schost-1$ /timesten/TimesTen/tt1122/startup/tt_tt1122 stop
phys-schost-2$ /timesten/TimesTen/tt1122/startup/tt_tt1122 stop
```

9 Leave the TimesTen user.

▼ Example: Enabling TimesTen Software to Run in the Cluster

This procedure creates the TimesTen resource.

1 Register the TimesTen resource type.

```
phys-schost-1# clresourcetype register ORCL.TimesTen_server
```

2 Create the failover resource group for the logical host.

```
phys-schost-1# clresourcegroup create -app-rg
```

3 **Create the logical-hostname resource `ha-host-1`.**

```
phys-schost-1# clreslogicalhostname create -g app-rg ha-host-1
```

4 **Enable the resource group `app-rg`.**

```
phys-schost-1# clresourcegroup online -eM app-rg
```

5 **Create the resource group to contain the TimesTen server resource `server-rg`.**

```
phys-schost-1# clresourcegroup create \
-p maximum_primaries=2 \
-p desired_primaries=2 \
server-rg
```

6 **Create the TimesTen server resource `tt-rs`.**

```
phys-host-1# clrs create -g server-rg  \
-t TimesTen_server \
-p Base_directory=/timesten/TimesTen \
-p Instance=tt1122 \
-p datastore=test \
-p table=sctest \
-p resource_dependencies=ha-host-1 tt-rs \
```

7 **Enable the resource group `server-rg`.**

```
phys-schost-1# clresourcegroup online -eM server-rg
```

▼ Example: Defining TimesTen Replication in an Active-Active Configuration

This procedure defines the TimesTen replication.

1 **Define the replicated data stores.**

On both nodes, add the following lines to the `/timesten/TimesTen/info/sys.odbc.ini` file.

```
[active]
DataStore=/timesten/data/active
PermSize=64
DatabaseCharacterSet=WE8MSWIN1252
```

2 **Define the datastore content, including a user with administrative permissions.**

```
phys-schost-1$ ttIsql active
Command> create table
t1 (col1 integer not null, col2 varchar(30), primary key (col1));
Command> CREATE REPLICATION repscheme1
       > ELEMENT complete DATASTORE
       > MASTER active on "phys-schost-2"
       > subscriber active on "phys-schost-1";
Command> CREATE REPLICATION repscheme2
       > ELEMENT complete DATASTORE
       > MASTER active on "phys-schost-1"
       > subscriber active on "phys-schost-2";
```

```
Command> CALL ttRepStart;
Command> create user repl identified by 'repl';
Command> grant admin to repl;
```

3 Duplicate the data store `active` to the remote node and start the replication agent.

```
phys-schost-2$ ttRepAdmin -duplicate -from active \
> -host phys-schost-1 -uid repl -pwd repl "dsn=active"
phys-schost-2$ ttIsql -e "call ttrepstart;exit;" active
```

4 Define the replication policy on all hosts.

```
phys-schost-1$ ttAdmin -repPolicy always active
phys-schost-2$ ttAdmin -repPolicy always active
```

◆ ◆ ◆ APPENDIX D

Deployment Example: Installing HA for TimesTen in a Scalable Subscriber Configuration

This appendix presents a complete example of how to install and configure the TimesTen application and data service in a scalable subscriber configuration. It presents a simple two-node cluster configuration.

If you need to install the application in any other configuration, refer to the general-purpose procedures presented elsewhere in this manual. For an example of TimesTen in a failover configuration, see Appendix B, "Deployment Example: Installing HA for TimesTen in a Failover Configuration," for an active-active configuration, see Appendix C, "Deployment Example: Installing HA for TimesTen in an Active-Active Configuration."

Target Cluster Configuration

This example uses a two-node cluster with the following node names:

- `phys-schost-1` (a physical node)
- `phys-schost-2` (a physical node)
- `master-host` (a logical host that is configured as a highly available master

Software Configuration

This deployment example uses the following software products and versions:

- Solaris 11 software for SPARC or x86 platforms
- Oracle Solaris Cluster 4.1 core software
- HA for TimesTen data service software
- TimesTen software version 11.2.2 tar file
- Your preferred text editor

This example assumes that you have already installed and established your cluster, and that you already made your TimesTen master highly available . It illustrates installation and configuration of the data service application only.

Assumptions

The instructions in this example were developed with the following assumptions:

- **Shell environment:** All commands and the environment setup in this example are for the Korn shell environment. If you use a different shell, replace any Korn shell-specific information or instructions with the appropriate information for you preferred shell environment.
- **User login:** Unless otherwise specified, perform all procedures by assuming a role that provides `solaris.cluster.admin`, `solaris.cluster.modify`, and `solaris.cluster.read` RBAC authorization.

Installing and Configuring HA for TimesTen on Local Storage in a Scalable Subscriber Configuration

These instructions assume that you are installing TimesTen software as the `timesten` user in a local directory. It is also assumed that you already installed the master in a highly available configuration. This can be done either on the same or on a different cluster.

The tasks you must perform to install and configure HA for TimesTen in a scalable configuration are as follows:

- "Example: Preparing the Cluster for HA for TimesTen" on page 50
- "Example: Installing TimesTen Software on Local Storage" on page 51
- "Example: Enabling TimesTen Software to Run in the Cluster" on page 52
- "Example: Defining TimesTen Replication in a Scalable Subscriber Configuration" on page 53

▼ Example: Preparing the Cluster for HA for TimesTen

1 Install and configure the cluster as instructed in *Oracle Solaris Cluster Software Installation Guide*.

Install the following cluster software components on both nodes.

- Oracle Solaris Cluster core software
- HA for TimesTen data service software

2 Add the `timesten` group and user.

```
phys-schost-1# groupadd -g 1000 timesten
phys-schost-1# useradd -g 1000 -m -d /timesten -s /bin/ksh timesten

phys-schost-2# groupadd -g 1000 timesten
phys-schost-2# useradd -g 1000 -m -d /timesten -s /bin/ksh timesten
```

▼ Example: Installing TimesTen Software on Local Storage

This example illustrates how to install TimesTen software on local storage. Perform this procedure on both nodes.

1 Create the `/etc/TimesTen` directory and set the permission for the `timesten` user.

```
phys-schost-1# mkdir /etc/TimesTen
phys-schost-1# chgrp -R timesten /etc/TimesTen
phys-schost-1# chmod 775 /etc/TimesTen

phys-schost-2# mkdir /etc/TimesTen
phys-schost-2# chgrp -R timesten /etc/TimesTen
phys-schost-2# chmod 775 /etc/TimesTen
```

2 Log in as the TimesTen user.

```
phys-schost-1# su - timesten
phys-schost-2# su - timesten
```

3 Create the data directory `/timesten/data` on local storage.

```
phys-schost-1$ mkdir /timesten/data
phys-schost-2$ mkdir /timesten/data
```

4 Change to your software directory and install TimesTen software on both nodes.

```
phys-schost-1$ cd /repository/
phys-schost-1$ tar xf timesten1122.tar
phys-schost-1$ cd ./solx8664
phys-schost-1$ ./setup.sh
```

Provide the following answers to the questions. Most prompts in this example accept the default.

```
Please choose an instance name for this installation? [ tt1122 ]
Instance name will be 'tt1122'. Is this correct? [ yes ]

   Of the three components:

[1] Client/Server and Data Manager
[2] Data Manager Only
[3] Client Only  Which would you like to install? [ 1 ]

 Of the following options :

[1] /export/tt
[2] /my-data/solx8664
[3] Specify a location
[q] Quit the installation

Where would you like to install the tt1122 instance of TimesTen? [ 1 ] 3
Please specify a directory to install TimesTen? [ /export/tt ] /timesten
Where would you like to create
the daemon home directory? [ /timesten/TimesTen/tt1122/info ]
```

```
Would you like to specify a different location for the daemon logs? [ no ]
Do you want to use the default port number for the TimesTen daemon? [ yes ]
Restrict access to the TimesTen installation to the group 'timesten'? [ yes ]
Please enter a value for TNS_ADMIN (s=skip)? [  ] s
What is the TCP/IP port number that you want
the TimesTen Server to listen n? [ 53389 ]
Do you want to install QuickStart and the TimesTen Documentation? [ no ]
Would you like to install the documentation (without QuickStart)? [ yes ]
Where would you like to create
the doc directory (s=skip)? [ /timesten/TimesTen/tt1122/doc ]
```

Repeat this step on the other node.

5 Adjust the PATH variable to contain the TimesTen binaries, for convenience.

```
phys-schost-1$ PATH=$PATH:/timesten/TimesTen/tt1122/bin
phys-schost-1$ export PATH

phys-schost-2$ PATH=$PATH:/timesten/TimesTen/tt1122/bin
phys-schost-2$ export PATH
```

6 Add the following lines to the `/timesten/TimesTen/tt1122/info` `sys.odbc.ini` file on both nodes.

```
[test]
DataStore=/timesten/data/test
PermSize=64
DatabaseCharacterSet=WE8MSWIN1252
```

7 Connect to this data store.

```
phys-schost-1$ ttIsql -e "exit;" test
phys-schost-2$ ttIsql -e "exit;" test
```

8 Stop the TimesTen server on both nodes.

```
phys-schost-1$ /timesten/TimesTen/tt1122/startup/tt_tt1122 stop
phys-schost-2$ /timesten/TimesTen/tt1122/startup/tt_tt1122 stop
```

9 Leave the TimesTen user.

▼ Example: Enabling TimesTen Software to Run in the Cluster

This example creates the TimesTen resource.

1 Register the TimesTen resource type.

```
phys-schost-1# clresourcetype register ORCL.TimesTen_server
```

2 Create the failover resource group `access-rg` for the shared-address resource.

```
phys-schost-1# clresourcegroup create access-rg
```

3 **Create the shared-address resource `ha-host-1`.**

```
phys-schost-1# clressharedaddress create -g access-rg ha-host-1
```

4 **Enable the resource group `access-rg`**

```
phys-schost-1# clresourcegroup online -eM access-rg
```

5 **Create the resource group `server-rg` to contain the TimesTen server resource.**

```
phys-schost-1# clresourcegroup create -p maximum_primaries=2 \
-p desired_primaries=2 \
server-rg
```

6 **Create the TimesTen server resource `tt-sub-rs`.**

```
phys-host-1# clrs create -g server-rg -t TimesTen_server \
-p Scalable=true \
-p Port_list=53889/tcp \
-p Base_directory=/timesten/TimesTen \
-p Instance=tt1122 \
-p datastore=test \
-p table=sctest \
-p resource_dependencies=ha-host-1 \
tt-sub-rs
```

7 **Enable the resource group `server-rg`.**

```
phys-schost-1# clresourcegroup online -eM server-rg
```

▼ Example: Defining TimesTen Replication in a Scalable Subscriber Configuration

This procedure defines the TimesTen replication.

1 **Define the replicated data stores.**

On the master node and on both cluster nodes that host the subscriber service, add the following lines to the `/timesten/TimesTen/info/sys.odbc.ini` file.

```
[replicated]
DataStore=/timesten/data/replicated
PermSize=64
DatabaseCharacterSet=WE8MSWIN1252
```

2 **Define the datastore content on the node that hosts the master server, including a user with administrative permissions.**

```
phys-schost-1$ ttIsql replicated
Command> create table
t1 (col1 integer not null, col2 varchar(30), primary key (col1));
Command> CREATE REPLICATION repscheme
       > ELEMENT complete DATASTORE
       > MASTER replicated on "master-host"
       > subscriber replicated on "phys-schost-1"
```

```
       > subscriber replicated on "phys-schost-2";

Command> CALL ttRepStart;
Command> create user repl identified by 'repl';
Command> grant admin to repl;
```

3 Duplicate the data store `replicated` to the subscriber node and start the replication agent.

```
phys-schost-1$ ttRepAdmin -duplicate -from replicated \
> -host master-host -uid repl -pwd repl "dsn=replicated"
phys-schost-1$  ttIsql -e "call ttrepstart;exit;" active

phys-schost-2$ ttRepAdmin -duplicate -from replicated \
> -host master-host -uid repl -pwd repl "dsn=replicated"
phys-schost-2$  ttIsql -e "call ttrepstart;exit;" active
```

4 Define the replication policy on all hosts.

```
phys-schost-1$ ttAdmin -repPolicy always active
phys-schost-2$ ttAdmin -repPolicy always active
```

Index

C

cache policy, 15
commands
  `cat`, 18, 19
  `chgrp`, 18
  `chmod`, 18
  `chown`, 18
  `clreslogicalhostname`, 15–16
  `clresource`, 15–16
  `clresourcegroup`, 15–16, 16
  `clresourcetype`, 15–16
  `clressharedaddress`, 15–16
  `groupadd`, 17
  `mkdir`, 18
  `su`, 18
  `ttIsql`, 15
  `useradd`, 18
configuration requirements, 14–15
configuration restrictions, 13–14

D

debugging
  `debug_level`, 29
  HA for TimesTen, 28–29
define, `data store`, 18
dependency, component, 14
deployment examples
  TimesTen in a failover configuration, 37–41
  TimesTen in a scalable subscriber configuration, 49–54
deployment examples *(Continued)*
  TimesTen in an active-active configuration, 43–48

E

enabling resource group, 16
environment variable, `LD_PRELOAD`, 18
extension properties, 20, 21, 31
  effect on fault monitor, 27

F

fault monitor, tuning, 27–28

H

HA for TimesTen
  debugging, 28–29
  fault monitors, 27–28
  installing, 19–20
  registering resources, 21–22
  software package, installing, 19–20
  verifying installation, 25–27
help, 10

I

in-memory database cache, 13

installing
HA for TimesTen, 19–20
TimesTen, 15–18

L
library, `libschost`, 18

N
non-global zones, 13

O
Oracle Solaris Cluster software, publisher, 19
Oracle Solaris software, publisher, 19
`ORCL.TimesTen_active` resource type, 31–32
`ORCL.TimesTen_clag` resource type, 32–33
`ORCL.TimesTen_cldaemon` resource type, 34
`ORCL.TimesTen_dbmon` resource type, 34
`ORCL.TimesTen_server` resource type, 34–36
`ORCL.TimesTen_standby` resource type, 36
overview
installation, 12
product, 11

P
package, 19–20
planning
replication type, 12–15
resource group topology, 12–15
publisher
Oracle Solaris Cluster software, 19
Oracle Solaris software, 19

R
replication
active-active, 47–48
replication *(Continued)*
definition, 27
scalable subscriber, 53–54
replication policy, 15
resource group, enabling, 16
resource type
`ORCL.TimesTen_active`, 31–32
`ORCL.TimesTen_clag`, 32–33
`ORCL.TimesTen_cldaemon`, 34
`ORCL.TimesTen_dbmon`, 34
`ORCL.TimesTen_server`, 34–36
`ORCL.TimesTen_standby`, 36
resource types, fault monitor, 27–28
resources, TimesTen application debugging, 28–29

S
software package, 19–20
system properties, effect on fault monitors, 27

T
technical support, 10
TimesTen application, fault monitor, 27–28
tuning, fault monitors, 27–28

V
verifying installation
HA for TimesTen, 25–27
TimesTen, 18–19