

**Oracle® Solaris Cluster Data Service for
Oracle TimesTen Guide**

ORACLE®

Part No: E63548
September 2015

Part No: E63548

Copyright © 2010, 2015, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

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

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

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

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

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about 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 unless otherwise set forth in an applicable agreement between you and Oracle. 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, except as set forth in an applicable agreement between you and Oracle.

Access to Oracle Support

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

Référence: E63548

Copyright © 2010, 2015, Oracle et/ou ses affiliés. Tous droits réservés.

Ce logiciel et la documentation qui l'accompagne sont protégés par les lois sur la propriété intellectuelle. Ils sont concédés sous licence et soumis à des restrictions d'utilisation et de divulgation. Sauf stipulation expresse de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, accorder de licence, transmettre, distribuer, exposer, exécuter, publier ou afficher le logiciel, même partiellement, sous quelque forme et par quelque procédé que ce soit. Par ailleurs, il est interdit de procéder à toute ingénierie inverse du logiciel, de le désassembler ou de le décompiler, excepté à des fins d'interopérabilité avec des logiciels tiers ou tel que prescrit par la loi.

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu'elles soient exemptes d'erreurs et vous invite, le cas échéant, à lui en faire part par écrit.

Si ce logiciel, ou la documentation qui l'accompagne, est livré sous licence au Gouvernement des Etats-Unis, ou à quiconque qui aurait souscrit la licence de ce logiciel pour le compte du Gouvernement des Etats-Unis, la notice suivante s'applique :

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

Ce logiciel ou matériel a été développé pour un usage général dans le cadre d'applications de gestion des informations. Ce logiciel ou matériel n'est pas conçu ni n'est destiné à être utilisé dans des applications à risque, notamment dans des applications pouvant causer un risque de dommages corporels. Si vous utilisez ce logiciel ou ce matériel dans le cadre d'applications dangereuses, il est de votre responsabilité de prendre toutes les mesures de secours, de sauvegarde, de redondance et autres mesures nécessaires à son utilisation dans des conditions optimales de sécurité. Oracle Corporation et ses affiliés déclinent toute responsabilité quant aux dommages causés par l'utilisation de ce logiciel ou matériel pour des applications dangereuses.

Oracle et Java sont des marques déposées d'Oracle Corporation et/ou de ses affiliés. Tout autre nom mentionné peut correspondre à des marques appartenant à d'autres propriétaires qu'Oracle.

Intel et Intel Xeon sont des marques ou des marques déposées d'Intel Corporation. Toutes les marques SPARC sont utilisées sous licence et sont des marques ou des marques déposées de SPARC International, Inc. AMD, Opteron, le logo AMD et le logo AMD Opteron sont des marques ou des marques déposées d'Advanced Micro Devices. UNIX est une marque déposée de The Open Group.

Ce logiciel ou matériel et la documentation qui l'accompagne peuvent fournir des informations ou des liens donnant accès à des contenus, des produits et des services émanant de tiers. Oracle Corporation et ses affiliés déclinent toute responsabilité ou garantie expresse quant aux contenus, produits ou services émanant de tiers, sauf mention contraire stipulée dans un contrat entre vous et Oracle. En aucun cas, Oracle Corporation et ses affiliés ne sauraient être tenus pour responsables des pertes subies, des coûts occasionnés ou des dommages causés par l'accès à des contenus, produits ou services tiers, ou à leur utilisation, sauf mention contraire stipulée dans un contrat entre vous et Oracle.

Accès aux services de support Oracle

Les clients Oracle qui ont souscrit un contrat de support ont accès au support électronique via My Oracle Support. Pour plus d'informations, visitez le site <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> ou le site <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> si vous êtes malentendant.

Contents

Using This Documentation	11
1 Installing and Configuring HA for TimesTen	13
HA for TimesTen Overview	13
Overview of Installing and Configuring HA for TimesTen	14
Planning the HA for TimesTen Installation and Configuration	14
Planning the Resource Group Topology for HA for TimesTen	14
HA for TimesTen and Oracle Solaris Zones	15
HA for TimesTen and In-Memory Database Cache	15
Configuration Restrictions	15
Configuration Requirements	16
Installing and Configuring TimesTen	17
▼ How to Enable a TimesTen Database to Run in a Global Zone Configuration	18
▼ How to Install and Configure TimesTen	19
Verifying the Installation and Configuration of TimesTen	21
▼ How to Verify Installation and Configuration of TimesTen	21
Installing the HA for TimesTen Package	22
▼ How to Install the HA for TimesTen Package	22
Registering and Configuring HA for TimesTen	23
Specifying Extension Properties for the TimesTen Resource	23
Creating and Enabling Resources for TimesTen	24
Verifying the HA for TimesTen Installation and Configuration	29
▼ How to Verify the HA for TimesTen Installation for Failover Configurations	30
▼ How to Verify the HA for TimesTen Installation for Scalable or Multiple- Master Configurations	30
Define the TimesTen Replication	31
Tuning the HA for TimesTen Fault Monitor	31

Operation of the Fault Monitor for ORCL.TimesTen_server Resource Type	32
Operation of the Fault Monitor for the Other HA for TimesTen Resource Types	32
Debugging HA for TimesTen	32
▼ How to Activate Debugging for HA for TimesTen	33
A HA for TimesTen Extension Properties	35
ORCL.TimesTen_active Extension Properties	35
ORCL.TimesTen_clag Extension Properties	36
Manually Changeable Extension Properties	36
Extension Properties Changeable by TimesTen Utilities	37
ORCL.TimesTen_cldaemon Extension Properties	38
ORCL.TimesTen_dbmon Extension Properties	38
ORCL.TimesTen_server Extension Properties	39
ORCL.TimesTen_standby Extension Properties	40
B Deployment Example: Installing HA for TimesTen in a Failover Configuration	43
Target Cluster Configuration	43
Software Configuration	43
Assumptions	44
Installing and Configuring HA for TimesTen on Shared Storage	44
▼ Example: Configuring Cluster Resources for TimesTen	44
▼ Example: Preparing the Cluster for TimesTen	45
▼ Example: Installing TimesTen Software on Shared Storage	45
▼ Example: Enabling TimesTen Software to Run in the Cluster	48
C Deployment Example: Installing HA for TimesTen in an Active-Active Configuration	49
Target Cluster Configuration	49
Software Configuration	49
Assumptions	50
Installing and Configuring HA for TimesTen on Local Storage	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	53

▼ Example: Defining TimesTen Replication in an Active-Active Configuration	54
D Deployment Example: Installing HA for TimesTen in a Scalable Subscriber Configuration	57
Target Cluster Configuration	57
Software Configuration	57
Assumptions	58
Installing and Configuring HA for TimesTen on Local Storage in a Scalable Subscriber Configuration	58
▼ Example: Preparing the Cluster for HA for TimesTen	59
▼ Example: Installing TimesTen Software on Local Storage	59
▼ Example: Enabling TimesTen Software to Run in the Cluster	61
▼ Example: Defining TimesTen Replication in a Scalable Subscriber Configuration	62
Index	65

Tables

TABLE 1	Tasks for Installing and Configuring HA for TimesTen	14
TABLE 2	TimesTen and Resource Group Topology	15

Using This Documentation

- **Overview** – Describes how to install and configure the Oracle Solaris Cluster Oracle TimesTen In-Memory Database data service
- **Audience** – Technicians, system administrators, and authorized service providers
- **Required knowledge** – Advanced experience troubleshooting and replacing hardware

Product Documentation Library

Documentation and resources for this product and related products are available at <http://www.oracle.com/pls/topic/lookup?ctx=E56676-01>.

Feedback

Provide feedback about this documentation at <http://www.oracle.com/goto/docfeedback>.

◆◆◆ 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 13](#)
- [“Overview of Installing and Configuring HA for TimesTen” on page 14](#)
- [“Planning the HA for TimesTen Installation and Configuration” on page 14](#)
- [“Installing and Configuring TimesTen” on page 17](#)
- [“Verifying the Installation and Configuration of TimesTen” on page 21](#)
- [“Installing the HA for TimesTen Package” on page 22](#)
- [“Registering and Configuring HA for TimesTen” on page 23](#)
- [“Verifying the HA for TimesTen Installation and Configuration” on page 29](#)
- [“Tuning the HA for TimesTen Fault Monitor” on page 31](#)
- [“Debugging HA for TimesTen” on page 32](#)

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 4.3 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 Tasks for Installing and Configuring HA for TimesTen

Task	Instructions
Plan the installation	“Planning the HA for TimesTen Installation and Configuration” on page 14
Install and configure the TimesTen software	“Installing and Configuring TimesTen” on page 17
Verify the installation and configuration	“How to Verify Installation and Configuration of TimesTen” on page 21
Install HA for TimesTen packages	“Installing the HA for TimesTen Package” on page 22
Register and configure HA for TimesTen resources	“Registering and Configuring HA for TimesTen” on page 23
Verify the HA for TimesTen installation and configuration	“Verifying the HA for TimesTen Installation and Configuration” on page 29
Tune the HA for TimesTen fault monitor	“Tuning the HA for TimesTen Fault Monitor” on page 31
Debug HA for TimesTen	“Debugging HA for TimesTen” on page 32

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 2 TimesTen and Resource Group Topology

Replication Type	Resource group topology
No replication	Failover resource group
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

HA for TimesTen and Oracle Solaris Zones

HA for TimesTen is supported in Oracle Solaris Zones. Oracle Solaris Cluster offers the following concepts for Oracle Solaris Zones:

- A zone cluster is an Oracle Solaris non-global zone 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 zone is a non-global zone that is managed by the HA for Oracle Solaris Zones (HA for Solaris Zones) data service, and is represented by a resource of a resource group.

HA for TimesTen is configurable in a zone cluster but not in an HA zone.

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.HASStoragePlus 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_cldDaemon	An ORCL.TimesTen_clag resource is required in an active-standby configuration.
ORCL.TimesTen_dbmon	An ORCL.TimesTen_cldDaemon resource is required in an active-standby configuration.
ORCL.TimesTen_active	An ORCL.TimesTen_dbmon resource is required in an active-standby configuration.

Resource Type	Dependency
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 23](#).

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 \(http://www.oracle.com/technetwork/database/timesten/documentation/index.html\)](http://www.oracle.com/technetwork/database/timesten/documentation/index.html).

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 zone cluster:

- 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 18](#)
 - [“How to Install and Configure TimesTen” on page 19](#)
- To install and configure TimesTen in a zone cluster, complete the following tasks:
 - [“How to Install and Configure TimesTen” on page 19](#)

▼ 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 28.

Before You Begin Ensure that the `/etc/netmasks` file has IP-address subnet and netmask entries for all logical hostnames. If necessary, edit the `/etc/netmasks` file to add any missing entries.

1. **(Optional) Assume the root role and register the SUNW.HASStoragePlus resource type.**

```
# clresourcetype register SUNW.HASStoragePlus
```

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.HASStoragePlus \  
-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 global-cluster or zone-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 that runs the instance in parallel.**

```
# mkdir /tt
```

2. **Add a group for TimesTen on every node.**

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

3. **Add a user who owns the TimesTen installation on every node.**

```
# 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.**

```
# 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.odb.c.ini` file.

```
$ cat - > your-instance-dir/info/sys.odb.c.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 17](#). During the verification, you complete the TimesTen post-installation 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 data service package is available from the configured publisher and that the `solaris` and `ha-cluster` publishers are valid.

```
# pkg list -a ha-cluster/data-service/timesten
# pkg publisher
PUBLISHER                TYPE    STATUS  P  LOCATION
solaris                   origin  online  F  solaris-repository
ha-cluster                 origin  online  F  ha-cluster-repository
```

For information about setting the `solaris` publisher, see [“Adding, Modifying, or Removing Package Publishers”](#) in *Adding and Updating Software in Oracle Solaris 11.3*.

Tip - Use the `-nv` options whenever you install or update to see what changes will be made, such as which versions of which packages will be installed or updated and whether a new BE will be created.

If you do not get any error messages when you use the `-nv` options, run the command again without the `-n` option to actually perform the installation or update. If you do get error messages, run the command again with more `-v` options (for example, `-nvv`) or more of the package FMRI pattern to get more information to help you diagnose and fix the problem. For troubleshooting information, see [Appendix A, “Troubleshooting Package Installation and Update,”](#) in *Adding and Updating Software in Oracle Solaris 11.3*.

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 your software, see [Chapter 11, “Updating Your Software,” in *Oracle Solaris Cluster 4.3 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 23](#)
- [“How to Create and Enable Resources for TimesTen in a Failover Configuration” on page 24](#)
- [“How to Create and Enable Resources for TimesTen in a Multiple-Master Configuration” on page 25](#)
- [“How to Create and Enable Resources for TimesTen in a Scalable Configuration” on page 27](#)
- [“How to Create and Enable Resources for TimesTen in an Active-Standby Configuration” on page 28](#)

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 HASStoragePlus 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=hasstorageplus-resource-name*
Sets a resource dependency between tt - resource -group and the HASStoragePlus 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 will 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. Assume the `root` role on a cluster node that will 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 \(http://www.oracle.com/technetwork/database/timesten/documentation/index.html\)](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 23](#).

Standard 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 4.3 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 debug logging for HA for TimesTen is active.

```
# 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 node.

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

3. 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
#
```

4. Restart the `syslogd` daemon in the appropriate node.

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

5. 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.

◆◆◆ A P P E N D I X 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 35
- “[ORCL.TimesTen_clag Extension Properties](#)” on page 36
- “[ORCL.TimesTen_cldaemon Extension Properties](#)” on page 38
- “[ORCL.TimesTen_dbmon Extension Properties](#)” on page 38
- “[ORCL.TimesTen_server Extension Properties](#)” on page 39
- “[ORCL.TimesTen_standby Extension Properties](#)” on page 40

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.odbci.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_c1ag Extension Properties

The resource type ORCL.TimesTen_c1ag 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_c1ag extension properties:

- [“Manually Changeable Extension Properties” on page 36](#)
- [“Extension Properties Changeable by TimesTen Utilities” on page 37](#)

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_clog 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_clddaemon 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.odbci.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.odbci.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

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:

- Oracle Solaris 11 software for SPARC or x86 platforms

- Oracle Solaris Cluster 4.2 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 your 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 44](#)
- [“Example: Preparing the Cluster for TimesTen” on page 45](#)
- [“Example: Installing TimesTen Software on Shared Storage” on page 45](#)
- [“Example: Enabling TimesTen Software to Run in the Cluster” on page 48](#)

▼ Example: Configuring Cluster Resources for TimesTen

1. **Register the necessary resource types on both nodes.**

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

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 HASStoragePlus resource in the RG-TT resource group.

```
phys-schost-1# clresource create -g RG-TT -t SUNW.HASStoragePlus -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 4.3 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 non-global zone 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.odbcc.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
```


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:

- Oracle Solaris 11 software for SPARC or x86 platforms
- Oracle Solaris Cluster 4.2 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 your 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 50](#)
- [“Example: Installing TimesTen Software on Local Storage” on page 51](#)
- [“Example: Enabling TimesTen Software to Run in the Cluster” on page 53](#)
- [“Example: Defining TimesTen Replication in an Active-Active Configuration” on page 54](#)

▼ Example: Preparing the Cluster for HA for TimesTen

1. **Install and configure the cluster as instructed in [Oracle Solaris Cluster 4.3 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 4.2 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 59](#)
- [“Example: Installing TimesTen Software on Local Storage” on page 59](#)
- [“Example: Enabling TimesTen Software to Run in the Cluster” on page 61](#)
- [“Example: Defining TimesTen Replication in a Scalable Subscriber Configuration” on page 62](#)

▼ Example: Preparing the Cluster for HA for TimesTen

1. **Install and configure the cluster as instructed in [Oracle Solaris Cluster 4.3 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.odbci.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;" activephys-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, 17
commands
 clreslogicalhostname, 18
 clresource, 18
 clresourcegroup, 18, 19
 clresourcetype, 18
 clressharedaddress, 18
 ttIsql, 17
configuration requirements, 16
configuration restrictions, 15

D

debugging
 Debug_level, 33
 HA for TimesTen, 32
define
 data store, 21
dependency, component, 16
deployment examples
 TimesTen in a failover configuration, 43
 TimesTen in a scalable subscriber
 configuration, 57
 TimesTen in an active-active configuration, 49

E

enabling resource group, 19
environment variable
 LD_PRELOAD, 20
extension properties, 23, 24, 35

effect on fault monitor, 31

F

fault monitor
 tuning, 31

H

HA for Solaris Zones restriction, 15
HA for TimesTen
 debugging, 32
 fault monitors, 31
 installing, 22
 registering resources, 24
 software package, installing, 22
 verifying installation, 29

I

in-memory database cache, 15
installing
 HA for TimesTen, 22
 TimesTen, 17

L

library
 libschostr, 20

N

non-global zones, 15

O

Oracle Solaris Cluster software
 publisher, 22, 22
ORCL.TimesTen_active resource type, 35
ORCL.TimesTen_clag resource type, 36
ORCL.TimesTen_cldaemon resource type, 38
ORCL.TimesTen_dbmon resource type, 38
ORCL.TimesTen_server resource type, 39
ORCL.TimesTen_standby resource type, 40
overview
 installation, 14
 product, 13

P

package, 22
planning
 replication type, 14
 resource group topology, 14
publisher
 Oracle Solaris Cluster software, 22, 22

R

replication
 active-active, 54
 definition, 31
 scalable subscriber, 62
replication policy, 17
resource group
 enabling, 19
resource type
 ORCL.TimesTen_active, 35
 ORCL.TimesTen_clag, 36
 ORCL.TimesTen_cldaemon, 38
 ORCL.TimesTen_dbmon, 38
 ORCL.TimesTen_server, 39
 ORCL.TimesTen_standby, 40
resource types
 fault monitor, 31
resources
 TimesTen application debugging, 32

S

software package, 22
standard properties
 effect on fault monitors, 31

T

TimesTen application
 fault monitor, 31
tuning
 fault monitors, 31

V

verifying installation
 HA for TimesTen, 29
 TimesTen, 21

Z

zone cluster support, 15