Oracle® Solaris Cluster Data Service for Apache Tomcat Guide

Part No: E69330
August 2018

Oracle Solaris Cluster Data Service for Apache Tomcat Guide

Part No: E69330

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: E69330

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

Tables

Examples

Using This Documentation

- **Overview** – Describes how to install and configure the HA for Apache Tomcat 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://docs.oracle.com/cd/E69294_01`.

Feedback

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

♦ ♦ ♦ CHAPTER 1

Installing and Configuring HA for Apache Tomcat

This chapter explains how to install and configure the Oracle Solaris Cluster HA for Apache Tomcat (HA for Apache Tomcat) data service.

This chapter contains the following sections.

Installing and Configuring HA for Apache Tomcat

Table 1, "Task Map: Installing and Configuring HA for Apache Tomcat," on page 13 lists the tasks for installing and configuring HA for Apache Tomcat. Perform these tasks in the order they are listed.

TABLE 1 Task Map: Installing and Configuring HA for Apache Tomcat

Task	For Instructions, Go To
1. Plan the installation.	"Planning the HA for Apache Tomcat Installation and Configuration" on page 14

Task	For Instructions, Go To
2. Install and configure Apache Tomcat.	"How to Install and Configure Apache Tomcat" on page 25
3. Verify installation and configuration.	"How to Verify the Installation and Configuration of Apache Tomcat" on page 27
4. Install HA for Apache Tomcat packages.	"Installing the HA for Apache Tomcat Package" on page 29
5.1 Register and Configure HA for Apache Tomcat as a failover data service.	"How to Register and Configure HA for Apache Tomcat as a Failover Data Service" on page 31
5.2 Register and configure HA for Apache Tomcat as a multiple-masters data service.	"How to Register and Configure HA for Apache Tomcat as a Multiple-Masters Data Service" on page 35
5.3 Register and configure HA for Apache Tomcat as a scalable data service.	"How to Register and Configure HA for Apache Tomcat as a Scalable Data Service" on page 39
6. Verify HA for Apache Tomcat installation and configuration.	"How to Verify the HA for Apache Tomcat Installation and Configuration" on page 44
7. Understanding the Apache Tomcat HA parameter file.	"Understanding the HA for Apache Tomcat Parameter File" on page 44
8. Understanding the HA for Apache Tomcat fault monitor.	"Understanding the HA for Apache Tomcat Fault Monitor" on page 47
9. How to debug HA for Apache Tomcat.	"How to Activate Debugging for HA for Apache Tomcat" on page 48

HA for Apache Tomcat

Apache Tomcat acts as a servlet engine behind an Apache web server, or you can configure it as a standalone web server that includes the servlet engine.

Apache Tomcat is freely available under Apache Software License and can be downloaded from `http://jakarta.apache.org`.

The HA for Apache Tomcat data service provides a mechanism for orderly startup and shutdown, fault monitoring, and automatic failover of the Apache Tomcat service.

Planning the HA for Apache Tomcat Installation and Configuration

This section contains the information you need to plan your Oracle Solaris Cluster HA for Apache Tomcat installation and configuration.

Note - HA for Apache Tomcat is supported on a zone cluster, and the zone-cluster node is treated like a global-cluster node.

Apache Tomcat and Oracle Solaris Zones

HA for Apache Tomcat is supported in an Oracle Solaris Cluster zone cluster. 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.

About Horizontal Scalability

Oracle Solaris Cluster includes a concept of horizontal scalability for data services called scalable service. IP-based load-balancing algorithms are integrated in this concept. Because of this reason you can scale horizontally without using hardware load balancers. For a more detailed discussion of this scalable service, see *Concepts for Oracle Solaris Cluster 4.4*.

Before using the HA for Apache Tomcat in a scalable configuration, you should closely examine the infrastructure of the cluster and the clients.

If your clients access your application using proxies, determine whether the used proxy stays the same during a session context. This is true for an intranet.

If the proxy changes during a session context, this signifies that from the load-balancing point of view that the source IP address is changing. This will spoil every IP based load-balancing effort, whether it is hardware or software.

When the client accesses the server over the Internet, it is *not* guaranteed that the source IP address remains the same during a session context.

HA for Apache Tomcat can be configured two ways to achieve horizontal scalability:

- Using HA for Apache Tomcat in a purely scalable configuration.
- Using HA for Apache Tomcat in a multiple-masters configuration.

The difference between the two configuration is in the way of accessing the nodes:

- A client accesses the scalable configuration by its shared address. In this case, Oracle Solaris Cluster does the load-balancing.
- A client accesses a multiple-masters configuration using each node's physical address. Load-balancing must be done outside of the cluster.

For more information about scalable data services, see *Concepts for Oracle Solaris Cluster 4.4*.

Both configuration options are discussed in the sections “HA for Apache Tomcat as a Scalable Configuration” on page 16 and “HA for Apache Tomcat as a Multiple-Masters Configuration” on page 16.

HA for Apache Tomcat as a Scalable Configuration

You can use HA for Apache Tomcat in a scalable configuration if it is guaranteed that the source IP address remains the same during a session context. This guarantee is achieved, for example, in an intranet.

If the source IP address might change during a session context and a scalable configuration is required, Apache Tomcat needs to be configured with session replication. This can be done by the application using a global file system or a database.

Caution - You will experience a performance penalty from the session-replication approach.

You will obtain better performance by using Apache Tomcat's inbound memory session replication for a scalable configuration with changing source IP addresses.

HA for Apache Tomcat as a Multiple-Masters Configuration

You can use HA for Apache Tomcat in a multiple-masters configuration in the other scenarios. External load-balancing is required. A typical configuration uses the scalable Apache web server as a load balancer and configures one physical host name of an Apache Tomcat instance behind each instance of the Apache web server. Another option is to use a hardware load balancer, which handles the session context.

The difference between scalable and multiple-masters configurations is in the way the clients access the cluster nodes. In a scalable configuration, they access the shared address. Otherwise, the clients access the global zone or hostnames.

Configuration Restrictions

The configuration requirements in this section apply only to Apache Tomcat.

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

Restriction to Deploying HA for Apache Tomcat in a Scalable Configuration

Deploy a scalable HA for Apache Tomcat configuration only if either session replication or reliable source IP addresses are achieved. Otherwise, the behavior of the application becomes unpredictable.

Restriction for the `Load_balancing_policy` Property

Setting the resource property `Load_balancing_policy` to `LB_STICKY` is strictly required, if HA for Apache Tomcat is deployed in a scalable configuration with reliable source IP addresses when no session replication is configured. Otherwise, the behavior of the application becomes unpredictable. In every other scalable configuration, the Sticky `Load_balancing_policy` setting helps to get more cache hits out of your caches.

Restriction for Scalable Services and Oracle Solaris Zones

HA for Apache Tomcat can be deployed in scalable configurations in Oracle Solaris Zones only if you use the zone cluster feature of Oracle Solaris Cluster.

Restriction for Apache Tomcat 6.0.28, 6.0.29, and 7.06

Starting with Apache Tomcat versions 6.0.28, 6.0.29, and 7.06, you must use the `wget` probe algorithm. This bypasses the normal probe, enabling the `TestUrl` parameter to work correctly. You can choose to use the `wget` probe algorithm for earlier Apache Tomcat versions or continue to use the original `mconnect` probe algorithm.

Configuration Requirements

These requirements apply to HA for Apache Tomcat only. You must meet these requirements before you proceed with your HA for Apache Tomcat installation and configuration.

⚠ **Caution -** Your data service configuration might not be supported if you do not adhere to these requirements.

Location of the Tomcat Home Directory for Scalable or Multiple-Masters Configurations

If you intend to install Apache Tomcat in a scalable resource group, create the Tomcat Home directory and its dynamic data on local storage. This is required because Apache Tomcat uses the directory structure to store its configuration, logs, deployed applications, and so on.

If your local storage is not large enough, you can use a cluster file system on the shared storage.

⚠ **Caution -** In this scenario, the deployment of Tomcat applications needs to occur on every node where Apache Tomcat is hosted.

Load Balancing for Multiple-Master Configurations

If you intend to install Apache Tomcat in a multiple-masters configuration, an external load balancer is required.

Location of the Tomcat Home Directory for Failover Configurations

If you intend to install Apache Tomcat in a failover resource group, create the Tomcat Home directory on the shared storage. The location for the Tomcat Home directory can reside on a cluster file system or it can reside on a highly available local file system which uses an HAStoragePlus resource. It is best practice to store it on a highly available local file system.

This requirement is necessary because Apache Tomcat uses the directory structure to store its configuration, logs, deployed applications, and so on. It is not recommended to store the binaries local and the dynamic parts of the data on the shared storage.

Tip - Mount a cluster file system with the `/global` prefix and mount a highly available local file system with the `/local` prefix.

Location of the `wget` Command for Apache Tomcat 6.0.28, 6.0.29, and 7.06

To support the `wget` probe algorithm for Apache Tomcat 6.0.28, 6.0.29, and 7.06, the `wget` command must be available from every node of the cluster.

Apache Tomcat Component Dependencies

You can configure the HA for Apache Tomcat data service to protect one or more Apache Tomcat instances. Each instance needs to be covered by one Apache Tomcat resource. The dependencies between the Apache Tomcat resource and other needed resources are described in Table 2, "Dependencies Between HA for Apache Tomcat Components in Failover Configurations," on page 19, Table 3, "Dependencies Between HA for Apache Tomcat Components in Scalable Configurations," on page 19, Table 4, "Dependencies Between HA for Apache Tomcat Components in Multiple-Masters Configurations," on page 20, or Table 5, "Dependency Types for HA for Apache Tomcat Resources," on page 20.

TABLE 2 Dependencies Between HA for Apache Tomcat Components in Failover Configurations

Component	Dependency
Apache Tomcat resource	`SUNW.HAStoragePlus` – This dependency is required only if the configuration uses a failover file system . `SUNW.LogicalHostName`

Because of the special requirements of a scalable configuration, you need no dependencies to storage or addresses, as long as every Apache Tomcat and its parameter file `pfile` is stored on the root file system. Otherwise, follow Table 3, "Dependencies Between HA for Apache Tomcat Components in Scalable Configurations," on page 19.

TABLE 3 Dependencies Between HA for Apache Tomcat Components in Scalable Configurations

Component	Description
Storage Resource	This resource is a `SUNW.HAStoragePlus` resource type.
`(Mandatory)` Apache Tomcat	Storage resource. The Storage resource manages the Apache Tomcat file-system mount points and ensures that Apache Tomcat is not started until they are mounted.
`(Mandatory)` Shared address	The shared address resource resides in a separate resource group on which the scalable resource group depends.

A multiple-masters configuration is running on more than one nodes like a scalable configuration, but without a shared address. Because of the special requirements of a multiple-masters configuration, you need no dependencies to storage or addresses, as long as every Apache Tomcat and its parameter file `pfile` is stored on the root file system. Otherwise, follow Table 4, "Dependencies Between HA for Apache Tomcat Components in Multiple-Masters Configurations," on page 20.

TABLE 4 Dependencies Between HA for Apache Tomcat Components in Multiple-Masters Configurations

Component	Dependency
Apache Tomcat resource	`SUNW.HAStoragePlus`. This dependency is required only, if the configuration uses a failover file system.

If more elaborate dependencies are required, see the `r_properties(7)` and `rg_properties(7)` man pages for further dependencies and affinities settings.

Note - For more detailed information about Apache Tomcat, refer to the `http://jakarta.apache.org` web page.

A SUNW.HAStoragePlus resource requires a `Resource_offline_restart` dependency type, while all other resources require a strong dependency type called `Resource_dependencies`. You must define the `Resource_offline_restart` dependency for the `SUNW.HAStoragePlus` resource if the resource type version is at least version 9. See Table 5, "Dependency Types for HA for Apache Tomcat Resources," on page 20 for the dependency type needed for each resource. The following resources are examples and you should evaluate the dependency to other resources on a case-by-case basis.

TABLE 5 Dependency Types for HA for Apache Tomcat Resources

Resource Name	Dependency Type
`SUNW.HAStoragePlus`	`Resource_offline_restart`
`SUNW.HAStorage`	`Resource_dependencies`
`SUNW.LogicalHostName`	`Resource_dependencies`

HA for Apache Tomcat Configuration and Registration Files

Apache Tomcat component has configuration and registration files in the directory `/opt/SUNWsctomcat/util`. These files let you register the Apache Tomcat component with Oracle Solaris Cluster.

Within these files, the appropriate dependencies have been applied.

```
# cd /opt/SUNWsctomcat
# ls -l util
total 4
-rwxr-xr-x   1 root     bin   1619 Apr 29 11:57 sctomcat_config
-r-xr-xr-x   1 root     bin   7058 Apr 29 11:58 sctomcat_register
# more util/*g
::::::::::::::
```

```
util/sctomcat_config
::::::::::::::
#
# Copyright 2006, 2012, Oracle and/or its affiliates.  All rights reserved.
#

#ident   "@(#)sctomcat_config.ksh 1.2     01/03/12"

# This file will be sourced in by sctomcat_register and the parameters
# listed below will be used.
#
# These parameters can be customized in (key=value) form
#
#        RS - name of the resource for the application
#        RG - name of the resource group containing RS
#      PORT - name of the port number
#        LH - name of the LogicalHostname SC resource
#  SCALABLE - true for a scalable resource or false for a failover resource
#   NETWORK - false or true, false for multiple master configurations without
#             shared address, in this case SCALABLE will be ignored
#
#     PFILE - absolute path to the parameter file for the Tomcat resource
#    HAS_RS - name of the HAStoragePlus SC resource
#             (it can be a , separated list for the dependencies)
#
# The following variables need to be set only if the agent runs in a
# local zone
#      ZONE - the zone name where the Apache Tomcat should run in
#             Optional
#    ZONEBT - The resource name which controls the zone.
#             Optional
#   PROJECT - A project in the zone, that will be used for this service
#             specify it if you have an su - in the start stop or probe,
#             or to define the smf credentials. If the variable is not set,
#             it will be translated as :default for the smf manifest
#             Optional

RS=
RG=
PORT=
LH=
NETWORK=false
SCALABLE=false
PFILE=
HAS_RS=

# local zone specific options
```

```
ZONE=
ZONE_BT=
PROJECT=
```

Installing and Configuring Apache Tomcat Software

This section contains the procedures you need to install and configure Apache Tomcat software.

- "How to Enable Apache Tomcat as a Failover Data Service" on page 22
- "How to Enable Apache Tomcat as a Multiple-Masters Data Service" on page 23
- "How to Enable Apache Tomcat as a Scalable Data Service" on page 24
- "How to Install and Configure Apache Tomcat" on page 25

Note - For more information about Apache Tomcat, refer to the `http://jakarta.apache.org` web page.

Determine how Apache Tomcat will be deployed in an Oracle Solaris Cluster configuration:

- Determine which version of Apache Tomcat will be deployed.
- Determine how many Apache Tomcat instances will be deployed.
- Determine which cluster file system will be used by each Apache Tomcat instance.

▼ How to Enable Apache Tomcat as a Failover Data Service

1. **Assume the `root` role or assume a role that provides `solaris.cluster.admin` authorization on one of the nodes in the cluster that will host Apache Tomcat.**

2. **Register the `SUNW.gds` and `SUNW.HAStoragePlus` resource type.**

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

3. **Create a failover resource group.**

```
# clresourcegroup create Apache-Tomcat-failover-resource-group
```

4. **Create a resource for the Apache Tomcat disk storage.**

```
# clresource create \
-g Apache-Tomcat-failover-resource-group \
-t SUNW.HAStoragePlus \
-p FileSystemMountPoints=Apache-Tomcat- instance-mount-points  Apache-Tomcat-hasp-resource
```

5. **Create a resource for the Apache Tomcat logical hostname.**

   ```
   # clreslogicalhostname \
   -g Apache-Tomcat-failover-resource-group \
   -h Apache-Tomcat-logical-hostname  Apache-Tomcat-lh-resource
   ```

6. **Enable the failover resource group that now includes the Apache Tomcat disk storage and logical hostname resources.**

   ```
   # clresourcegroup online -eM -n current-node  Apache-Tomcat-failover-resource-group
   ```

Next Steps Go to "How to Install and Configure Apache Tomcat" on page 25.

▼ How to Enable Apache Tomcat as a Multiple-Masters Data Service

1. **Assume the `root` role or assume a role that provides `solaris.cluster.admin` authorization on one of the nodes in the cluster that will host Apache Tomcat.**

2. **Register the `SUNW.gds` and `SUNW.HAStoragePlus` resource type.**

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

3. **Create the resource group for the multiple-masters data service.**

   ```
   # clresourcegroup create \
   -p Maximum_primaries=2 \
   -p Desired_primaries=2  Apache-Tomcat-scalable-resource-group
   ```

   If you need more nodes, adjust `Maximum_primaries` and `Desired_primaries` to the appropriate value.

4. **Create a resource for the Apache Tomcat Disk Storage if it is not in the root (`/`) file system.**

   ```
   # clresource create \
   -g Apache-Tomcat-failover-resource-group \
   -t SUNW.HAStoragePlus \
   -p FileSystemMountPoints=Apache-Tomcat- instance-mount-points  Apache-Tomcat-hasp-resource
   ```

5. **Enable the multiple-masters resource group that now includes the Apache Tomcat disk storage and logical hostname resources.**

```
# clresourcegroup online -eM  Apache Tomcat-failover-resource-group
```

Next Steps Go to "How to Install and Configure Apache Tomcat" on page 25.

▼ How to Enable Apache Tomcat as a Scalable Data Service

1. **Assume the `root` role or assume a role that provides `solaris.cluster.admin` authorization on one of the nodes in the cluster that will host Apache Tomcat.**

2. **Register the `SUNW.gds` and `SUNW.HAStoragePlus` resource type.**

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

3. **Create a failover resource group for the shared address resource.**

```
# clresourcegroup create Apache-Tomcat-shared-address-resource-group
```

4. **Create the shared address resource.**

```
# clressharedaddress create \
-g Apache-Tomcat-shared-address-resource-group \
-h Apache-Tomcat-shared-address-hostname \
Apache-Tomcat-shared-address-resource
```

5. **Bring online the shared-address resource group.**

```
# clresourcegroup online -eM Apache-Tomcat-shared-address-resource-group
```

6. **Create the resource group for the scalable service.**

```
# clresourcegroup create \
-p Maximum_primaries=2 \
-p Desired_primaries=2 \
-p RG_dependencies=Apache-Tomcat-shared-address-resource-group \
Apache-Tomcat-scalable-resource-group
```

If you need more nodes, adjust `Maximum_primaries` and `Desired_primaries` to the appropriate value.

7. **Create a resource for the Apache Tomcat disk storage if it is not in the root (`/`) file system.**

```
# clresource create Apache-Tomcat-hasp-resource \
-g Apache-Tomcat-scalable-group \
-t SUNW.HAStoragePlus \
-p FileSystemMountPoints=Apache-Tomcat-instance-mount-points
```

8. **Enable the failover resource group that now includes the Apache Tomcat disk storage and logical hostname resources.**

```
# clresourcegroup online -eM  Apache-Tomcat-resource-group
```

Next Steps Go to "How to Install and Configure Apache Tomcat" on page 25.

▼ How to Install and Configure Apache Tomcat

Before You Begin Determine how Apache Tomcat will be deployed in Oracle Solaris Cluster.

- Ensure that you have enabled Apache Tomcat to run in a failover, multiple-masters, or scalable data service.

  For conceptual information on scalable and failover data services, see *Concepts for Oracle Solaris Cluster 4.4*.
- Determine which user name will run Apache Tomcat.
- Determine how many Apache Tomcat versions and instances will be deployed.
- If more than one instance of a version will be deployed, determine whether they share the binaries.
- Determine which cluster file system will be used by each Apache Tomcat instance.

1. **Assume the `root` role or assume a role that provides `solaris.cluster.modify` and `solaris.cluster.admin` authorizations.**

2. **Create a user and a group if required.**

Caution - In this scenario, the deployment of the Apache Tomcat group and user needs to occur on every node where Apache Tomcat is hosted.

If Apache Tomcat is to run under a non-root user, you have to create the appropriate user and the appropriate group. For these tasks, use the following commands.

a. **Create the group.**

```
# groupadd -g 1000 tomcat
```

b. **Create the user.**

```
# useradd -u 1000 -g 1000 -d /global/tomcat -s /bin/ksh tomcat
```

3. **If you are not logged in as the `root` role, switch to the appropriate user name.**

   ```
   # su - user-name
   ```

4. **Install Apache Tomcat.**

   If you deploy Apache Tomcat as a failover data service, install Apache Tomcat onto a shared file system within Oracle Solaris Cluster.

   You should install Apache Tomcat onto shared disks. For a discussion of the advantages and disadvantages of installing the software on a local versus a cluster file system, see "Determining the Location of the Application Binaries" in *Planning and Administering Data Services for Oracle Solaris Cluster 4.4*.

   If you will deploy Apache Tomcat as a failover data service ,install the Apache Tomcat binaries on the shared storage on **one** node. If Apache Tomcat will be deployed as a scalable or a multiple-masters data service, install the Apache Tomcat binaries on the local storage on **every** node, that will host the Apache Tomcat data service.

   Note - Refer to `http://tomcat.apache.org/index.html` for instructions about installing Apache Tomcat.

   If you deploy Apache Tomcat as a scalable data service for a scalable or a multiple-masters configuration, repeat the following step at every node that will host Apache Tomcat.

5. **Create the environment script.**

   Create a Korn shell or a C shell script (dependent on the login-shell of your Apache Tomcat user name) to set the environment variables for Apache Tomcat. You must set the environment variables in a shell script and not in the user's profile.

   With this mechanism you can install and run multiple Apache Tomcat versions and instances under one user name.

   Caution - These shell scripts must be available on every node that can host the Apache Tomcat data service, and be the same on those nodes.

   - For a failover configuration, store them on the shared storage of the node or in the target zone.
   - For a scalable or a multiple-masters configuration, store the scripts on the local file system of every node or on the shared storage.

Example 1 Korn shell script to set the environment variables for Apache Tomcat 6.0.28

```
# more env.ksh
#!/usr/bin/ksh
#
# Environment for Tomcat
#
JAVA_HOME=/usr/j2se
export JAVA_HOME
CATALINA_HOME=/global/mnt1/jakarta-tomcat-6.0.28
export CATALINA_HOME
```

Example 2 C Shell Script to Set the Environment Variables for Apache Tomcat

```
# more env.csh
#!/usr/bin/csh
#
# Environment for Tomcat
#
setenv JAVA_HOME /usr/j2se
setenv CATALINA_HOME /global/mnt1/jakarta-tomcat-6.0.28
```

The environment variables are version and configuration dependent.

Verifying the Installation and Configuration of Apache Tomcat

This section contains the procedure you need to verify the installation and configuration of Apache Tomcat.

▼ How to Verify the Installation and Configuration of Apache Tomcat

This procedure does not verify that your application is highly available because you have not installed your data service yet.

1. **Start Apache Tomcat.**

   Switch to the Apache Tomcat user name in the target node (in the following example, it is `root`) and change to the directory where the environment script is located.

Note - The output messages of the start and shutdown commands are version dependent.

```
# . ./env.ksh
# cd $CATALINA_HOME/bin
# ./startup.sh
Using CATALINA_BASE:   /tomcat/jakarta-tomcat-version
Using CATALINA_HOME:   /tomcat/jakarta-tomcat-version
Using CATALINA_TMPDIR: /tomcat/jakarta-tomcat-version/temp
Using JAVA_HOME:       /usr/j2se
```

2. **Check the Installation.**

   Start a web browser and connect to the cluster node with `http://`*nodename*`:8080`. If you see the default Apache Tomcat home page, everything is working correctly.

3. **Stop Apache Tomcat.**

```
# ./shutdown.sh
Using CATALINA_BASE:   /tomcat/jakarta-tomcat-version
Using CATALINA_HOME:   /tomcat/jakarta-tomcat-version
Using CATALINA_TMPDIR: /tomcat/jakarta-tomcat-version/temp
Using JAVA_HOME:       /usr/j2se
```

How to Deploy the Apache Tomcat Application

If you configure Apache Tomcat as a failover resource, execute the following steps on one node. If you deploy Apache Tomcat as a scalable configuration, repeat the following steps on every node.

Caution - For a scalable configuration, the Apache Tomcat configuration must be the same on all the Apache Tomcat hosts.

1. **Assume the `root` role or assume a role that provides `solaris.cluster.modify` and `solaris.cluster.admin` authorizations.**

2. **Modify the configuration files `server.xml` and the Apache Tomcat user configuration to the desired content.**

   For more information, see `http://tomcat.apache.org/index.html`.

3. **Start Apache Tomcat as described in "How to Verify the Installation and Configuration of Apache Tomcat" on page 27.**

4. **Deploy your application.**

   For more information, see http://tomcat.apache.org/index.html.

5. **Test your application.**

   For example, connect with a web browser to the Host with the appropriate port and select the start page of your application.

6. **Stop Apache Tomcat as described in "How to Verify the Installation and Configuration of Apache Tomcat" on page 27.**

   The port on which Apache Tomcat is serving is required for the configuration of the Apache Tomcat resource.

   It is best practice to use the start page of the deployed application for the Fault Monitoring. In this case, the TestCmd described in "Understanding the HA for Apache Tomcat Parameter File" on page 44 can be get /start_page. With this procedure, you monitor Apache Tomcat and the application it is serving.

Installing the HA for Apache Tomcat Package

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

▼ How to Install the HA for Apache Tomcat Package

Perform this procedure on each cluster node where you want the HA for Apache Tomcat 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/tomcat
# 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 *Updating Systems and Adding Software in Oracle Solaris 11.4*.

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 *Updating Systems and Adding Software in Oracle Solaris 11.4*.

3. **Install the HA for Apache Tomcat software package.**

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

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

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

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

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

   For instructions about updating your software, see Chapter 10, "Updating Software Packages" in *Updating Your Oracle Solaris Cluster 4.4 Environment*.

Registering and Configuring HA for Apache Tomcat

This section contains the procedures you need to configure HA for Apache Tomcat.

- "How to Register and Configure HA for Apache Tomcat as a Failover Data Service" on page 31
- "How to Register and Configure HA for Apache Tomcat as a Multiple-Masters Data Service" on page 35
- "How to Register and Configure HA for Apache Tomcat as a Scalable Data Service" on page 39

How to Register and Configure HA for Apache Tomcat as a Failover Data Service

Before You Begin This procedure assumes that you installed the data service packages.

If you did not install the HA for Apache Tomcat packages, go to "Installing the HA for Apache Tomcat Package" on page 29.

It is assumed that the file system of Apache Tomcat will be mounted as a failover file system.

1. **Prepare the parameter file, which is required by HA for Apache Tomcat.**

```
# cd /opt/SUNWsctomcat/bin
# cp pfile desired-location
```

2. **Edit the parameter file `pfile` and follow the comments within that file.**
Repeat this step for every Apache Tomcat instance you need.
For example:

```
#!/usr/bin/ksh
#
# Copyright (c) 2006, 2012, Oracle and/or its affiliates. All rights reserved.
#

#ident   "@(#)pfile.ksh 1.6     01/12/16"

# Set the Apache Tomcat specific environment variables which the start, stop
# and check fuctions will use
#
# EnvScript     Script to set runtime environment for tomcat
# User          Apache Tomcat User
# Basepath      Absolute path to Tomcat Home directory i.e. 5.x or 6.x CATALINA_HOME
# Host          Hostname to test Apache Tomcat
# Port          Port where Apache Tomcat is configured
# TestCmd       Apache Tomcat test command
#
# Some versions of Apache Tomcat do not react on the probe algorithm with the
# http get command. Because of this we provide an alternative probe method
# which uses wget. If wget is not installed on your system you must install it
# manually and provide the WgetPath variable. If the WgetPath variable is not set,
# it defaults to /usr/sfw/bin/wget.
# You can use the wget algorithm for https probe as well which are not possible
# with the http get command.
#
# Use the variables TestUrl and WgetPath as a replacement for:
# Port, Host, TestCmd
# You can specify only one set of variables.
#
# TestUrl       Url where Tomcat serves a web site. This can be done either
#               by the http or https  protocol
#               Format: http://hostname:port/startpage
#               Example: http://localhost:8080/
#               You might want to include options here like:
#               "--no-cookies http://localhost:8080/"
#               The requirement here is that wget -O - ${TestUrl} produces
#               the output containing the ReturnString.
# WgetPath      Abolute path to wget, leaving this variable empty defaults to:
#               /usr/sfw/bin/wget
# Startwait     Sleeping $Startwait seconds after completion of the
#               start command
# ReturnString  This string must be present in the output of the http get command
#               or in the output of the wget command.
EnvScript=
User=
Basepath=
Host=
Port=8080
TestCmd="get /index.jsp"
TestUrl=
WgetPath=
ReturnString="CATALINA"
Startwait=20
```

The following is an example for Apache Tomcat 5.5.27.

```
EnvScript=/global/mnt1/tomtest/env.ksh
User=tomtest
Basepath=/global/mnt1/tomtest/jakarta-tomcat-5.5.27
Host=tomcat-lh
Port=8080
TestCmd="get /index.jsp"
TestUrl=
WgetPath=
ReturnString="CATALINA"
Startwait=20
```

This example for Apache Tomcat 5.5.27 configures a standalone web server with the default start page `index.jsp`. The start page contains the string `CATALINA`.

To use the alternative `wget` probe, assuming `wget` is available under `/usr/sfw/bin/wget`, instead set the following variables:

```
EnvScript=/global/mnt1/tomtest/env.ksh
User=tomtest
Basepath=/global/mnt1/tomtest/jakarta-tomcat-6.0.28
```

```
Host=
Port=
TestCmd=
TestUrl="http://tomcat-lh:8080/"
WgetPath=
ReturnString="CATALINA"
Startwait=20
```

Caution - The parameter files must be available on every node that can host the Apache Tomcat data service. For a scalable or multiple-masters configuration, store the parameter files on the local file system of every node or on the shared storage. The parameter files cannot differ for any instance of Apache Tomcat on the various nodes.

3. **Create and register each required Apache Tomcat component.**

```
# cd /opt/SUNWsctomcat/util
# cp sctomcat_config desired-location
```

4. **Edit the `sctomcat_config` file and follow the comments within that file.**

   Repeat this step for each Apache Tomcat instance you need.

   For example:

```
#
# Copyright (c) 2006, 2012, Oracle and/or its affiliates.  All rights reserved.
#

#ident   "@(#)sctomcat_config.ksh 1.2     01/03/12"

# This file will be sourced in by sctomcat_register and the parameters
# listed below will be used.
#
# These parameters can be customized in (key=value) form
#
#        RS - name of the resource for the application
#        RG - name of the resource group containing RS
#      PORT - name of the port number
#        LH - name of the LogicalHostname SC resource
#  SCALABLE - true for a scalable resource or false for a failover resource
#   NETWORK - false or true, false for multiple master configurations without
#             shared address, in this case SCALABLE will be ignored
#
#     PFILE - absolute path to the parameter file for the Tomcat resource
#    HAS_RS - name of the HAStoragePlus SC resource
#             (it can be a , separated list for the dependencies)
#
# The following variables need to be set only if the agent runs in a
# local zone
#      ZONE - the zone name where the Apache Tomcat should run in
#             Optional
#    ZONEBT - The resource name which controls the zone.
#             Optional
#   PROJECT - A project in the zone, that will be used for this service
#             specify it if you have an su - in the start stop or probe,
#             or to define the smf credentials. If the variable is not set,
#             it will be translated as :default for the smf manifest
#             Optional

RS=
RG=
PORT=
LH=
NETWORK=false
SCALABLE=false
PFILE=
HAS_RS=

# local zone specific options

ZONE=
ZONE_BT=
PROJECT=
```

The following is an example for Apache Tomcat.

```
RS=tomcat-res
RG=tomcat-rg
PORT=8080
LH=tomcat-lh
SCALABLE=false
PFILE=/global/mnt1/pfile
NETWORK=true
HAS_RS=tomcat-hastplus-res
```

5. **After editing `sctomcat_config`, register the resource.**

   ```
   # ksh ./sctomcat_register -f desired-location/sctomcat_config
   ```

   Repeat this step for each Apache Tomcat instance you need.

6. **Enable each Apache Tomcat resource.**

   ```
   # clresource status
   # clresource enable Apache-Tomcat-resource
   ```

How to Register and Configure HA for Apache Tomcat as a Multiple-Masters Data Service

Before You Begin This procedure assumes that you installed the data service packages during your initial Oracle Solaris Cluster installation.

If you did not install the HA for Apache Tomcat packages as part of your initial Oracle Solaris Cluster installation, go to "Installing the HA for Apache Tomcat Package" on page 29.

1. **Assume the `root` role or assume a role that provides `solaris.cluster.modify` and `solaris.cluster.admin` authorizations.**

2. **Prepare the parameter file, which is required by HA for Apache Tomcat.**

```
# cd /opt/SUNWsctomcat/bin
# cp pfile desired-location
```

3. **Edit the parameter file and follow the comments within that file.**

   Repeat this step for every Apache Tomcat instance and every node you need.

   For example:

```
#!/usr/bin/ksh
#
# Copyright (c) 2006,2012, Oracle and/or its affiliates. All rights reserved.
#
#
#ident   "@(#)pfile.ksh 1.6     01/12/16"

# Set the Apache Tomcat specific environment variables which the start, stop
# and check fuctions will use
#
# EnvScript     Script to set runtime environment for tomcat
# User          Apache Tomcat User
# Basepath      Absolute path to Tomcat Home directory i.e. 5.x or 6.x CATALINA_HOME
# Host          Hostname to test Apache Tomcat
# Port          Port where Apache Tomcat is configured
# TestCmd       Apache Tomcat test command
#
# Some versions of Apache Tomcat do not react on the probe algorithm with the
# http get command. Because of this we provide an alternative probe method
# which uses wget. If wget is not installed on your system you must install it
# manually and provide the WgetPath variable. If the WgetPath variable is not set,
# it defaults to /usr/sfw/bin/wget.
# You can use the wget algorithm for https probe as well which are not possible
# with the http get command.
```

```
#
# Use the variables TestUrl and WgetPath as a replacement for:
# Port, Host, TestCmd
# You can specify only one set of variables.
#
# TestUrl       Url where Tomcat serves a web site. This can be done either
#               by the http or https  protocol
#               Format: http://hostname:port/startpage
#               Example: http://localhost:8080/
#               You might want to include options here like:
#               "--no-cookies http://localhost:8080/"
#               The requirement here is that wget -O - ${TestUrl} produces
#               the output containing the ReturnString.
# WgetPath      Abolute path to wget, leaving this variable empty defaults to:
#               /usr/sfw/bin/wget
# Startwait     Sleeping $Startwait seconds after completion of the
#               start command
# ReturnString  This string must be present in the output of the http get command
#               or in the output of the wget command.EnvScript=
EnvScript=
User=
Basepath=
Host=
Port=8080
TestCmd="get /index.jsp"
TestUrl=
WgetPath=
ReturnString="CATALINA"
Startwait=20
```

The following is an example for Apache Tomcat 5.5.27.

```
EnvScript=/tomcat/env.ksh
User=tomcat
Basepath=/tomcat/jakarta-tomca-5.5.27
Host=localhost
Port=8080
TestCmd="get /index.jsp"
TestUrl=
WgetPath=
ReturnString="CATALINA"
Startwait=20
```

This example for Apache Tomcat 5.5.27 configures a standalone web server with the default start page `index.jsp`. The start page contains the string `CATALINA`.

Depending on the selected probe method, the Host parameter or the IP alias in the `TestUrl` parameter depends on the location of the `pfile`:

- If the `pfile` is stored on the shared storage, you must use `localhost`.
- If the `pfile` is stored on the local storage, it can be either `localhost` or the node's host name.

To use the alternative `wget` probe, assuming `wget` is available under `/usr/sfw/bin/wget`, set the following variables:

```
EnvScript=/global/mnt1/tomtest/env.ksh
User=tomtest
Basepath=/global/mnt1/tomtest/jakarta-tomcat-6.0.28
Host=
Port=
TestCmd=
TestUrl="http://localhost:8080/"
WgetPath=
ReturnString="CATALINA"
Startwait=20
```

Caution - The parameter files must be available on every node that can host the Apache Tomcat data service. For a scalable or a multiple-masters configuration, store the parameter files on the local file system of every node or on the shared storage. The parameter files must be the same for every instance of Apache Tomcat on the various nodes.

4. **Create and register each required Apache Tomcat component.**

```
# cd /opt/SUNWsctomcat/util
# cp sctomcat_config desired place
```

5. **Edit the `sctomcat_config` file and follow the comments within that file.**

Repeat this step for every Apache Tomcat instance you need.

For example:

```
#
# Copyright 2006, 2012, Oracle and/or its affiliates. All rights reserved.
#

#ident   "@(#)sctomcat_config.ksh 1.2     01/03/12"

# This file will be sourced in by sctomcat_register and the parameters
# listed below will be used.
#
# These parameters can be customized in (key=value) form
#
#        RS - name of the resource for the application
#        RG - name of the resource group containing RS
#      PORT - name of the port number
#        LH - name of the LogicalHostname SC resource
#  SCALABLE - true for a scalable resource or false for a failover resource
#   NETWORK - false or true, false for multiple master configurations without
#             shared address, in this case SCALABLE will be ignored
#
#     PFILE - absolute path to the parameter file for the Tomcat resource
#    HAS_RS - name of the HAStoragePlus SC resource
#             (it can be a , separated list for the dependencies)
#
# The following variables need to be set only if the agent runs in a
# local zone
#      ZONE - the zone name where the Apache Tomcat should run in
#             Optional
#    ZONEBT - The resource name which controls the zone.
#             Optional
#   PROJECT - A project in the zone, that will be used for this service
#             specify it if you have an su - in the start stop or probe,
#             or to define the smf credentials. If the variable is not set,
#             it will be translated as :default for the smf manifest
#             Optional

RS=
RG=
PORT=
LH=
NETWORK=false
SCALABLE=false
PFILE=
HAS_RS=

# local zone specific options

ZONE=
ZONE_BT=
PROJECT=
```

The following is an example for Apache Tomcat.

```
RS=tomcat-res
RG=tomcat-rg
PORT=8080
LH=tomcat-lh
SCALABLE=false
NETWORK=false
PFILE=/local/pfile
HAS_RS=Apache-Tomcat-hasp-resource
```

Omit the LH parameter here because it will not be used. Set the parameter HAS_RS only when another file system (GFS or local file system) is used and the appropriate *Apache-Tomcat-hasp-resource* is created. The ZONE, ZONE_BT, and PROJECT variables are needed only if the multiple-masters container resources are zones managed by the HA for Oracle Solaris Zones agent.

6. **After editing sctomcat_config, register the resource.**

```
# ksh ./sctomcat_register -f desired-location/sctomcat_config
```

Repeat this step for every Apache Tomcat instance you need.

7. **Enable each Apache Tomcat resource.**
Repeat this step for each Apache Tomcat resource created.

```
# clresource status
# clresource enable Apache-Tomcat-resource
```

How to Register and Configure HA for Apache Tomcat as a Scalable Data Service

Before You Begin This procedure assumes that you installed the data service packages during your initial Oracle Solaris Cluster installation.

If you did not install the HA for Apache Tomcat packages as part of your initial Oracle Solaris Cluster installation, go to "Installing the HA for Apache Tomcat Package" on page 29.

1. **Assume the root role or assume a role that provides solaris.cluster.modify and solaris.cluster.admin authorizations.**

2. **Prepare the parameter file, which is required by the Oracle Solaris Cluster HA for Apache Tomcat.**

```
# cd /opt/SUNWsctomcat/bin
# cp pfile desired-place
```

3. **Edit the parameter file pfile and follow the comments within that file.**
Repeat this Step for every Apache Tomcat instance and every node you need.
For example:

```
#!/usr/bin/ksh
#
# Copyright (c) 2006, 2012, Oracle and/or its affiliates. All rights reserved.
```

```
#

#ident  "@(#)pfile.ksh  1.6     01/03/12"

# Set the Apache Tomcat specific environment variables which the start,
# stop and check functions will use
#
# EnvScript     Script to set runtime environment for tomcat
# User          Apache Tomcat user name
# Basepath      Absolute pathname to Tomcat Home directory i.e. 5.x or
                6.x CATALINA_HOME
# Host          Hostname to test Apache Tomcat
# Port          Port where Apache Tomcat is configured
# TestCmd       Apache Tomcat test command
#
#
# Some versions of Apache Tomcat do not react on the probe algorithm with the
# http get command. Because of this we provide an alternative probe method
# which uses wget. If wget is not installed on your system you must install it
# manually and provide the WgetPath variable. If the WgetPath variable is not
# set, it defaults to /usr/sfw/bin/wget.
# You can use the wget algorithm for https probe as well which are not possible
# with the http get command.
#
# Use the variables TestUrl and WgetPath as a replacement for:
# Port, Host, TestCmd
# You can specify only one set of variables.
#
# TestUrl       Url where Tomcat serve a web site. This can be done either
#               by the http or https  protocol
#               Format: http://hostname:port/startpage
#               Example: http://localhost:8080/
#               You might want to include options here like:
#               "--no-cookies http://localhost:8080/"
#               The requirement here is that wget -O - ${TestUrl} produces
#               the output containing the ReturnString.
# WgetPath      Abolute path to wget, leaving this variable empty defaults to:
#               /usr/sfw/bin/wget
# Startwait     Sleeping $Startwait seconds after completion of the
#               start command
# ReturnString  This string must be present in the output of the http get
#               command or in the output of the wget command.
EnvScript=
User=
Basepath=
Host=
Port=8080
TestCmd="get /index.jsp"
```

```
TestUrl=
WgetPath=
ReturnString="CATALINA"
Startwait=20
```

To use the alternative `wget` probe, assuming `wget` is available under `/usr/sfw/bin/wget`, set the following variables:

```
EnvScript=/global/mnt1/tomtest/env.ksh
User=tomtest
Basepath=/global/mnt1/tomtest/jakarta-tomcat-6.0.28
Host=
Port=
TestCmd=
TestUrl="http://localhost:8080/"
WgetPath=
ReturnString="CATALINA"
Startwait=20
```

The following example configures a standalone web server with the default start page `index.jsp`. The start page contains the string `CATALINA`.

```
EnvScript=/tomcat/env.ksh
User=tomcat
Basepath=/tomcat/jakarta-tomcat-version
Host=localhost
Port=8080
TestCmd="get /index.jsp"
TestUrl=
WgetPath=
ReturnString="CATALINA"
Startwait=20
```

Depending on the selected probe method, the Host parameter or the IP alias in the `TestUrl` parameter depends on the location of the `pfile`:

- If the pfile is stored on the shared storage, you must use `localhost`.
- If the pfile is stored on the local storage, it can be either `localhost` or the node's host name.

Note - Do **not** use the shared address here.

Caution - The parameter files must be available on every node that can host the Apache Tomcat data service. Store the parameter files on the local file system of every node or on the shared storage. The parameter files must be the same for every instance of Apache Tomcat on the various nodes.

4. **Create and register each required Apache Tomcat component.**

   Repeat this step for every Apache Tomcat instance you need.

   ```
   # cd /opt/SUNWsctomcat/util
   # cp sctomcat_config desired place
   ```

5. **Edit the `sctomcat_config` file and follow the comments within that file.**

   Repeat this step for every Apache Tomcat instance you need.

   For example:

   ```
   #
   # Copyright 2006, 2012, Oracle and/or its affiliates.  All rights reserved.
   #

   #ident   "@(#)sctomcat_config.ksh 1.2     01/03/12"

   # This file will be sourced in by sctomcat_register and the parameters
   # listed below will be used.
   #
   # These parameters can be customized in (key=value) form
   #
   #        RS - name of the resource for the application
   #        RG - name of the resource group containing RS
   #      PORT - name of the port number
   #        LH - name of the LogicalHostname SC resource
   #  SCALABLE - true for a scalable resource or false for a failover resource
   #   NETWORK - false or true, false for multiple master configurations without
   #             shared address, in this case SCALABLE will be ignored
   #
   #     PFILE - absolute path to the parameter file for the Tomcat resource
   #    HAS_RS - name of the HAStoragePlus SC resource
   #             (it can be a , separated list for the dependencies)
   #
   # The following variables need to be set only if the agent runs in a
   # local zone
   #      ZONE - the zone name where the Apache Tomcat should run in
   #             Optional
   #    ZONEBT - The resource name which controls the zone.
   #             Optional
   #   PROJECT - A project in the zone, that will be used for this service
   #             specify it if you have an su - in the start stop or probe,
   #             or to define the smf credentials. If the variable is not set,
   #             it will be translated as :default for the smf manifest
   #             Optional

   RS=
   RG=
   ```

```
PORT=
LH=
NETWORK=false
SCALABLE=false
PFILE=
HAS_RS=

# local zone specific options

ZONE=
ZONE_BT=
PROJECT=
```

The following is an example for Apache Tomcat.

```
RS=tomcat-res
RG=tomcat-rg
PORT=8080
LH=tomcat-lh
SCALABLE=true
NETWORK=true
PFILE=/tomcat/pfile
HAS_RS=Apache-Tomcat-hasp-resource
```

Configure the `LH` parameter as the name of the shared address. Set the `HAS_RS` parameter only when another file system (cluster file system or highly available local file system) is used and the appropriate `Apache-Tomcat-hasp-resource` is created.

6. **After editing `sctomcat_config`, register the resource.**

   ```
   # ksh ./sctomcat_register desired place/sctomcat_config
   ```

   Repeat this step for every Apache Tomcat instance you need.

7. **Enable each Apache Tomcat resource.**

   Repeat this step for each Apache Tomcat resource created.

   ```
   # clresource status
   # clresource enable Apache-Tomcat-resource
   ```

Verifying the HA for Apache Tomcat Installation and Configuration

This section contains the procedure to verify that you installed and configured your data service correctly.

▼ How to Verify the HA for Apache Tomcat Installation and Configuration

1. **Assume the `root` role or assume a role that provides `solaris.cluster.modify` and `solaris.cluster.admin` authorizations.**

2. **Ensure all the Apache Tomcat resources are online.**

   ```
   # cluster status
   ```

   Enable each Apache Tomcat resource that is not online.

   ```
   # clresource enable Apache-Tomcat- resource
   ```

3. **If you deployed a failover data service, switch the Apache Tomcat resource group to another cluster node, such as *node2*.**

   ```
   # clresourcegroup switch -n node2 Apache-Tomcat-failover-resource-group
   ```

Understanding the HA for Apache Tomcat Parameter File

Use this information to understand the contents of the HA for Apache Tomcat parameter file. This section describes the structure and the content of the HA for Apache Tomcat parameter file, as well as the strategy to choose some of its variables.

Structure of the HA for Apache Tomcat Parameter File

HA for Apache Tomcat uses a parameter file to pass parameters to the start, stop, and probe command. This parameter file needs to be a valid Korn shell script which sets several variables.

The structure of this file appears in Table 6, "Structure of the HA for Apache Tomcat Parameter File," on page 45. For examples of the parameter file refer to "Registering and Configuring HA for Apache Tomcat" on page 30.

TABLE 6 Structure of the HA for Apache Tomcat Parameter File

Variable	Explanation
`EnvScript`	This is a `ksh` script or a `csh` script, depending on the login-shell of the user name that owns Apache Tomcat. The purpose of this script is to set the Apache Tomcat specific environment variables, which are needed to start and stop the Apache Tomcat instance.
`User`	The owner of the Apache Tomcat instance.
`Basepath`	The absolute pathname to the directory where the Apache Tomcat `/bin` directory resides. Typically, it is `CATALINA_HOME`.
`Host`	The host to test the functionality of the Apache Tomcat server process. The Test is done via a connection to `Host:Port`.
`Port`	A port where Apache Tomcat is serving. This `Port` is used together with the `Host` to test the functionality of the Apache Tomcat server process.
`TestCmd`	This variable represents the command that is passed to the Apache Tomcat server process to test its sanity.
`ReturnString`	The variable `ReturnString` represents the string that **must** be present in the answer to the `TestCmd`. It cannot be `Connection refused` because this string is in the answer when the Apache Tomcat server process is not running.
`Startwait`	This variable represents the number of seconds to wait after the Apache Tomcat start command is completed. It lasts until the Apache Tomcat server process is fully operational. The absolute number of seconds depends on the speed and the load of the hardware. A good strategy is to start with 10 to 20 seconds.
`TestUrl`	This variable presents a complete `http` or `https` URL, which is passed to the `wget` command to test the Tomcat server's sanity.
`WgetPath`	This variable presents the absolute path to the `wget` binary, if the `wget` binary is not located under `/usr/sfw/bin/wget`.

The parameters in Table 6, "Structure of the HA for Apache Tomcat Parameter File," on page 45 can be changed at any time. The only difference is when changes take effect.

The following parameters of the HA for Apache Tomcat parameter file are used for starting and stopping Apache Tomcat. Changes to these parameters take effect at every restart or disabling and enabling of an HA for Apache Tomcat resource.

- `EnvScript`
- `User`
- `Basepath`

- `Startwait`
- `TestUrl`
- `WgetPath`

The following parameters of the HA for Apache Tomcat parameter file are used within the fault monitor. Changes to these parameters take effect at every `Thorough_probe_interval`.

- `Host`
- `Port`
- `TestCmd`
- `ReturnString`

Strategy to Choose the `TestCmd` and the `ReturnString` Variable

The following alternatives are available to choose the `ReturnString` variable:

- Take the start page of your application and set the `TestCmd` to get /*start_page*. If you picked the `wget` probe method, set `TestUrl` to `http://ipalias:port/`*start_page*. With this strategy, you are monitoring that Apache Tomcat is serving your application.
- Take the Apache Tomcat default start page and set the `TestCmd` to get `/`*default-start-page*. If you picked the `wget` method, set `TestUrl` to `http://`*ipalias*`:`*port*`/`*default-start-page*. In this case, set the `ReturnString` to a string contained in the start page. Typically this string is `CATALINA`. With this strategy, you are monitoring that Apache Tomcat is serving its default application.
- Deploy a test application (which is not provided with the HA for Apache Tomcat agent) to Apache Tomcat. Set the `TestCmd` to get *start-page_of_the_application*. If you use the `wget` method, set `TestUrl` to `http://ipalias:port/`*start-page_of_the_application*. In this case, set the `ReturnString` to a string contained in the start page. With this strategy, you are monitoring that Apache Tomcat is serving your test application.
- If none of the above is appropriate, set the `TestCmd` to `get /`*a-page-that-does-not-exist*. In this case, set the `ReturnString` to a string contained in the Error Page. With this strategy, you are monitoring that Apache Tomcat is operational, because it registers that it must deliver a page that does not exist.

You can evaluate the different pages by connecting using a browser with *hostname*`:`*port* and specifying the different pages.

Understanding the HA for Apache Tomcat Fault Monitor

This section describes the HA for Apache Tomcat fault monitor's probing algorithm and functionality, as well as the conditions, messages, and recovery actions associated with unsuccessful probing.

For conceptual information on fault monitors, see *Concepts for Oracle Solaris Cluster 4.4*.

Resource Properties

The HA for Apache Tomcat fault monitor uses the same resource properties as the resource type `SUNW.gds`. Refer to the `SUNW.gds`(7) man page for a complete list of resource properties.

Probing Algorithm and Functionality

The following steps are executed to monitor the sanity of Apache Tomcat.

1. Sleeps for `Thorough_probe_interval`.
2. Pings the `Host`, which is configured in the HA for Apache Tomcat parameter file unless the `TestUrl` parameter is specified in the HA for Apache Tomcat parameter file.
3. Connects to Apache Tomcat via `Host` and `Port`. If the connection is successful, it sends the `TestCmd` and tests whether the `ReturnString` comes back. If it fails, it is rescheduled after five seconds. If this fails again, the probe restarts Apache Tomcat.

Caution - The `ReturnString` **cannot** be `Connection refused` because this string is returned if no connection is possible.

4. If the Apache Tomcat process has died, PMF will interrupt the probe to immediately restart the Apache Tomcat.
5. If the Apache Tomcat is repeatedly restarted and subsequently exhausts the `Retry_count` within the `Retry_interval`, then a failover is initiated for the resource group onto another node. This is done if the resource property `Failover_enabled` is set to `TRUE`.

Debugging HA for Apache Tomcat

HA for Apache Tomcat has a file named `config` that enables you to activate debugging for Apache Tomcat resources. This file is in the `/opt/SUNWsctomcat/etc` directory.

▼ How to Activate Debugging for HA for Apache Tomcat

1. **Determine whether debugging for HA for Apache Tomcat is active.**

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

   If debugging is inactive, `daemon.notice` is set in the file `/etc/syslog.conf`.

2. **If debugging is inactive, edit the `/etc/syslog.conf` file to change `daemon.notice` to `daemon.debug`.**

3. **Confirm that debugging for HA for Apache Tomcat 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.**

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

5. **Edit the `/opt/SUNWsczone/sczbt/etc/config` file to change the `DEBUG=` variable to one of the following settings:**

   - `DEBUG=ALL`
   - `DEBUG=`*resource name*
   - `DEBUG=`*resource name*`,`*resource name*`,…`

   ```
   # cat /opt/SUNWsctomcat/etc/config
   #
   ```

```
# Copyright 2006, 2012, Oracle and/or its affiliates.  All rights reserved.
#
# Usage:
#       DEBUG=<RESOURCE_NAME> or ALL
#
DEBUG=ALL
#
```

Note - To deactivate debugging, repeat this procedure, changing `daemon.debug` to `daemon.notice` and changing the `DEBUG` variable to `DEBUG=`.

♦ ♦ ♦ APPENDIX A

Deployment Example: Installing Apache Tomcat

This appendix presents a complete example of how to install and configure the Apache Tomcat application and HA for Apache Tomcat data service. 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 guide.

Target Cluster Configuration

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

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

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

Software Configuration

This deployment example uses the following software products and versions:

- Oracle Solaris 11.4 software
- Oracle Solaris Cluster 4.4 core software
- HA for Apache Tomcat
- The minimum version of Apache Tomcat, 5.5.17
- The `wget` binary is available under `/usr/sfw/bin/wget`.
- 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.

Shell and User Login 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 as the `root` role or assume a role that provides `solaris.cluster.admin`, `solaris.cluster.modify`, and `solaris.cluster.read` authorization.

Installing and Configuring Apache Tomcat on Global Storage

The tasks you must perform to install and configure Apache Tomcat are as follows:

- "Example: Preparing the Cluster for Apache Tomcat" on page 52
- "Example: Configuring Cluster Resources for Apache Tomcat" on page 53
- "Example: Installing Apache Tomcat 5.5.17 Software on Shared Storage" on page 53
- "Example: Modifying the Apache Tomcat Configuration Files" on page 54
- "Example: Enabling the Apache Tomcat Software to Run in the Cluster" on page 55

▼ Example: Preparing the Cluster for Apache Tomcat

1. **Install and configure the cluster as instructed in *Installing and Configuring an Oracle Solaris Cluster 4.4 Environment*.**

   Install the following cluster software components on both nodes.

   - Oracle Solaris Cluster core software
   - Oracle Solaris Cluster data service for Apache Tomcat

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

```
phys-schost-1# groupadd -g 1000 tomcat
```

```
phys-schost-2# groupadd -g 1000 tomcat
phys-schost-1# useradd -g 1000 -d /global/mnt3/tomcat -m -s /bin/ksh tomcat
phys-schost-2# useradd -g 1000 -d /global/mnt3/tomcat -m -s /bin/ksh tomcat
```

▼ Example: Configuring Cluster Resources for Apache Tomcat

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

```
phys-schost-1# clresourcetype register SUNW.gds SUNW.HAStoragePlus
```

2. **Create the Apache Tomcat resource group.**

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

3. **Create the logical host.**

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

4. **Create the `HAStoragePlus` resource in the `RG-TOM` resource group.**

```
phys-schost-1# clresource create -g RG-TOM -t SUNW.HAStoragePlus -p AffinityOn=TRUE \
-p FileSystemMountPoints=/global/mnt3,/global/mnt4 RS-TOM-HAS
```

5. **Enable the resource group.**

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

▼ Example: Installing Apache Tomcat 5.5.17 Software on Shared Storage

These steps illustrate how to install Apache Tomcat 5.5.17 software in the directory `/global/mnt3/tomcat`. As long as only one node is mentioned it needs to be the node where your resource group is online.

1. **Install the Apache Tomcat binaries.**

```
phys-schost-1# su - tomcat
phys-schost-1# gzcat apache-tomcat-5.5.17.tar.gz|tar xvf -
phys-schost-1# gzcat apache-tomcat-5.5.17-compat.tar.gz|tar xvf -
```

2. **Create your environment script `/global/mnt3/tomcat/env.ksh` with the following contents.**

```
cat env.ksh
#!/usr/bin/ksh
JAVA_HOME=/usr/j2se
CATALINA_HOME=/global/mnt3/tomcat/apache-tomcat-5.5.17
export JAVA_HOME CATALINA_HOME
```

Example: Modifying the Apache Tomcat Configuration Files

1. **Copy the Apache Tomcat configuration file from the agent directory to its deployment location.**

```
phys-schost-1# cp /opt/SUNWsctomcat/util/sctomcat_config /global/mnt3
phys-schost-1# cp /opt/SUNWsctomcat/bin/pfile /global/mnt3
phys-schost-1# chown tomcat:tomcat /global/mnt3/pfile
```

2. **Add this cluster's information to the `sctomcat_config` configuration file.**

   The following listing shows the relevant file entries and the values to assign to each entry.

```
…
RS=RS-TOM
RG=RG-TOM
PORT=8080
LH=ha-host-1
NETWORK=true
SCALABLE=false
PFILE=/global/mnt3/pfile
HAS_RS=RS-TOM-HAS
```

3. **Save and close the file.**

4. **Add this cluster's information to the parameter file `/global/mnt3/pfile`.**

   The following listing shows the relevant file entries and the values to assign to each entry.

```
EnvScript=/global/mnt3/tomcat/env.ksh
User=tomcat
Basepath=/global/mnt3/tomcat/apache-tomcat-5.5.17
Host=ha-host-1
Port=8080
TestCmd="get /index.jsp"
ReturnString="CATALINA"
```

```
Startwait=20
```

As an alternative, you can use the following set of variables:

```
EnvScript=/global/mnt3/tomcat/env.ksh
User=tomcat
Basepath=/global/mnt3/tomcat/apache-tomcat-5.5.17
TestUrl="http://ha-host-1:8080/"
ReturnString="CATALINA"
Startwait=20
```

5. **Save and close the file.**

▼ Example: Enabling the Apache Tomcat Software to Run in the Cluster

1. **Run the `sctomcat_register` script to register the resource.**

   ```
   phys-schost-1# ksh /opt/SUNWsctomcat/util/sctomcat_register \
   -f /global/mnt3/sctomcat_config
   ```

2. **Enable the resource.**

   ```
   phys-schost-1# clresource enable RS-TOM
   ```

Index