Oracle® Solaris Cluster Data Service for Apache Tomcat Guide

Part No: E26824
March 2012, E26824–01

ORACLE

Copyright © 2010, 2012, 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, the following notice is applicable:

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

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

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 disposition de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, breveter, 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 concédé sous licence au Gouvernement des Etats-Unis, ou à toute entité qui délivre la licence de ce logiciel ou l'utilise pour le compte du Gouvernement des Etats-Unis, la notice suivante s'applique:

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

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 des dommages corporels. Si vous utilisez ce logiciel ou 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 ce type d'applications.

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 d'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. 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.

120306@25097

Contents

Preface

Oracle Solaris Cluster Data Service for Apache Tomcat Guide explains how to install and configure Oracle Solaris Cluster data services.

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

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

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

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

Using UNIX Commands

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

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

Typographic Conventions

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

TABLE P–1 Typographic Conventions

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

Shell Prompts in Command Examples

The following table shows the default UNIX system prompt and superuser prompt for shells that are included in the Oracle Solaris OS. Note that the default system prompt that is displayed in command examples varies, depending on the Oracle Solaris release.

TABLE P–2 Shell Prompts

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

Related Documentation

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

Topic	Documentation
Hardware installation and administration	*Oracle Solaris Cluster 4.0 Hardware Administration Manual* Individual hardware administration guides
Concepts	*Oracle Solaris Cluster Concepts Guide*
Software installation	*Oracle Solaris Cluster Software Installation Guide*
Data service installation and administration	*Oracle Solaris Cluster Data Services Planning and Administration Guide* and individual data service guides
Data service development	*Oracle Solaris Cluster Data Services Developer's Guide*
System administration	*Oracle Solaris Cluster System Administration Guide* *Oracle Solaris Cluster Quick Reference*
Software upgrade	*Oracle Solaris Cluster Upgrade Guide*
Error messages	*Oracle Solaris Cluster Error Messages Guide*
Command and function references	*Oracle Solaris Cluster Reference Manual* *Oracle Solaris Cluster Data Services Reference Manual* *Oracle Solaris Cluster Geographic Edition Reference Manual* *Oracle Solaris Cluster Quorum Server Reference Manual*

Access to Oracle Support

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

Getting Help

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

- Your name and email address (if available)
- Your company name, address, and phone number
- The model number and serial number of your systems
- The release number of the operating environment (for example, Oracle Solaris 11)
- The release number of Oracle Solaris Cluster (for example, Oracle Solaris Cluster 4.0)

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

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

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

◆ ◆ ◆ CHAPTER 1

Installing and Configuring Oracle Solaris Cluster HA for Apache Tomcat

This chapter explains how to install and configure HA for Apache Tomcat.

This chapter contains the following sections.

Installing and Configuring Oracle Solaris Cluster HA for Apache Tomcat

Table 1–1 lists the tasks for installing and configuring Oracle Solaris Cluster HA for Apache Tomcat. Perform these tasks in the order they are listed.

TABLE 1–1 Task Map: Installing and Configuring Oracle Solaris Cluster HA for Apache Tomcat

Task	For Instructions, Go To
1. Plan the installation.	"Planning the Oracle Solaris Cluster HA for Apache Tomcat Installation and Configuration" on page 13

TABLE 1–1 Task Map: Installing and Configuring Oracle Solaris Cluster HA for Apache Tomcat *(Continued)*

Task	For Instructions, Go To
2. Install and configure Apache Tomcat.	"How to Install and Configure Apache Tomcat in the Global Zone" on page 24
3. Verify installation and configuration.	"How to Verify the Installation and Configuration of Apache Tomcat" on page 29
4. Install Oracle Solaris Cluster HA for Apache Tomcat packages.	"Installing the HA for Apache Tomcat Package" on page 31
5.1 Register and Configure Oracle Solaris Cluster HA for Apache Tomcat as a failover data service.	"How to Register and Configure Oracle Solaris Cluster HA for Apache Tomcat as a Failover Data Service" on page 32
5.2 Register and Configure Oracle Solaris Cluster HA for Apache Tomcat as a multiple masters data service.	"How to Register and Configure Oracle Solaris Cluster HA for Apache Tomcat as a Multiple Masters Data Service" on page 35
5.3 Register and Configure Oracle Solaris Cluster HA for Apache Tomcat as a scalable data service.	"How to Register and Configure Oracle Solaris Cluster HA for Apache Tomcat as a Scalable Data Service" on page 39
5.4 Register and configure Oracle Solaris Cluster HA for Apache Tomcat in a failover zone.	"Configuring Oracle Solaris Cluster HA for Apache Tomcat in a Failover Zone" on page 43
6. Verify Oracle Solaris Cluster HA for Apache Tomcat Installation and Configuration.	"How to Verify the Oracle Solaris Cluster HA for Apache Tomcat Installation and Configuration" on page 46
7. Understanding the Oracle Solaris Cluster Apache Tomcat HA parameter file.	"Understanding the Oracle Solaris Cluster Apache Tomcat HA Parameter File" on page 46
8. Understanding the Oracle Solaris Cluster HA for Apache Tomcat Fault Monitor.	"Understanding Oracle Solaris Cluster HA for Apache Tomcat Fault Monitor" on page 48
9. How to debug Oracle Solaris Cluster HA for Apache Tomcat.	"How to Activate Debugging for HA for Apache Tomcat" on page 49

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

The following table describes the relation between the application components and the related Oracle Solaris Cluster data service.

TABLE 1–2 Protection of Components

Component	Protected by
Apache Tomcat	Oracle Solaris Cluster HA for Apache Tomcat
Failover Zone	Oracle Solaris Cluster HA for Zones

Planning the Oracle Solaris Cluster 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 – Oracle Solaris Cluster HA for Apache Tomcat is supported on a zone cluster, and the zone cluster node is treated like a physical node.

Apache Tomcat and Solaris Zones

Oracle Solaris Cluster HA for Apache Tomcat is supported in Solaris Zones, and Oracle Solaris Cluster supports two concepts for Solaris Zones.

- Zones are containers that are running after a reboot of the node. These containers, combined with resource groups having the nodename *nodename:zonename* as a valid "nodename" in the resource groups nodename list.
- Failover Zone containers are managed by the Solaris Zones agent, and are represented by a resource of a resource group.

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 *Oracle Solaris Cluster Concepts Guide*.

Before using the Oracle Solaris Cluster 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.

Oracle Solaris Cluster HA for Apache Tomcat can be configured two ways to get horizontal scalability:

1. Using Oracle Solaris Cluster HA for Apache Tomcat in a purely scalable configuration.
2. Using Oracle Solaris Cluster 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 *Oracle Solaris Cluster Concepts Guide*.

Both configuration options are discussed in the sections"Oracle Solaris Cluster HA for Apache Tomcat as a scalable configuration" on page 14 and "Oracle Solaris Cluster HA for Apache Tomcat as a multiple master configuration" on page 15.

Oracle Solaris Cluster HA for Apache Tomcat as a scalable configuration

You can use Oracle Solaris Cluster 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 in 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 get a performance penalty from this approach.

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

- For Apache Tomcat 5.x and 6.x, inbound memory session replication as a built-in feature is on the feature list.

Oracle Solaris Cluster HA for Apache Tomcat as a multiple master configuration

You can use Oracle Solaris Cluster HA for Apache Tomcat in a multiple master 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 configuration is in the way the clients access the cluster nodes. In a scalable configuration, they access the shared address. Otherwise, the clients access the physical 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 deploy Oracle Solaris Cluster HA for Apache Tomcat in a scalable configuration

Deploy a scalable Oracle Solaris Cluster 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

Setting the resource parameter Load_balancing_policy to LB_STICKY is strictly required, if Oracle Solaris Cluster 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 helps to get more cache hits out of your caches.

Restriction for Scalable Services and Solaris Zones

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

Restriction for the Apache Tomcat smf Service Name in a Failover Zone

The Apache Tomcat configuration in a failover zone uses the `smf` component of Oracle Solaris Cluster HA for Solaris Zones. The registration of the Apache Tomcat data service in a failover zone defines an `smf` service to control the Apache Tomcat database. The name of this `smf` service is generated in this naming scheme:
`svc:/application/sczone-agents:`*resource-name*. No other `smf` service with exactly this name can exist.

The associated `smf` manifest is automatically created during the registration process in this location and naming scheme:
`/var/svc/manifest/application/sczone-agents/`*resource-name*`.xml`. No other manifest can coexist with this name.

Restriction for Apache Tomcat 5.5.28, 6.0.28, 6.0.29, and 7.06

Starting with Apache Tomcat versions 5.5.28, 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 Oracle Solaris Cluster HA for Apache Tomcat only. You must meet these requirements before you proceed with your Oracle Solaris Cluster 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 Global 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 master 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 Global File System (GFS) or it can reside on a Failover File System (FFS) with an HAStoragePlus resource. It is best practice to store it on a FFS.

This is required 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.

Note – It is best practice to mount Global File Systems with the `/global` prefix and to mount Failover File Systems with the `/local` prefix.

Location of the wget Command for Apache Tomcat 5.5.28, 6.0.28, 6.0.29, and 7.06

To support the `wget` probe algorithm for Apache Tomcat 5.5.28, 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 Oracle Solaris Cluster 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 1–3, Table 1–4, Table 1–5, or Table 1–6.

TABLE 1–3 Dependencies Between HA for Apache Tomcat Components in Failover Configurations

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

TABLE 1–3 Dependencies Between HA for Apache Tomcat Components in Failover Configurations *(Continued)*

Component	Dependency
Apache Tomcat resource in a failover zone	Oracle Solaris Cluster HA for the Solaris Zone boot resource. `SUNW.HAStoragePlus` `SUNW.LogicalHostName` – This dependency is required only if the zones boot resource does not manage the zone's IP address.

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 1–4.

TABLE 1–4 Dependencies Scalable

Component	Description
Storage Resource	This resource can be a `SUNW.HAStorage` or a `SUNW.HAStoragePlus` resource type.
Apache Tomcat`(Mandatory)`	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.
Shared address`(Mandatory)`	The shared address resource resides in a separate resource group on which the scalable resource group depends.

A multiple master 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 1–5.

TABLE 1–5 Dependencies Between HA for Apache Tomcat Components in Failover Configurations

Component	Dependency
Apache Tomcat resource in the global zone	`SUNW.HAStoragePlus`. This dependency is required only, if the configuration uses a failover file system.
Apache Tomcat resource in a failover zone	Oracle Solaris Cluster HA for the Solaris Zone boot resource. `SUNW.HAStoragePlus`

If more elaborate dependencies are required, see the `r_properties(5)` and `rg_properties(5)` 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 1–6 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 1–6 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
-r-xr-xr-x   1 root     bin        3752 Apr 29 11:57 sctomcat_smf_register
-r-xr-xr-x   1 root     bin        1350 Apr 29 11:58 sctomcat_smf_remove
# 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

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

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

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

- 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.

- Determine the type of the target zone where you will install Apache Tomcat. Valid zone types include the global zone or the failover zone.

To install and configure Apache Tomcat in a *global zone* configuration, complete the following tasks:

- "How to Enable Apache Tomcat to run in a Global Zone Configuration" on page 21
- "How to Install and Configure Apache Tomcat in the Global Zone" on page 24

To install and configure Apache Tomcat in a *failover zone* configuration, complete the following tasks:

- "How to Enable Apache Tomcat to run in Failover Zone Configuration" on page 26
- "How to Install and Configure Apache Tomcat in a Failover Zone" on page 27

You will find installation examples for each zone type in:

- Appendix A, "Deployment Example: Installing Apache Tomcat in the Global Zone"
- Appendix B, "Deployment Example: Installing Apache Tomcat in a Failover Zone"

How to Enable Apache Tomcat to run in a Global Zone Configuration

1 **Become superuser or assume a role that provides solaris.cluster.admin RBAC 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.**

Note – Perform this step only if you create a failover data service.

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

4 **(Optional) Create a resource for the Apache Tomcat Disk Storage.**

Note – Perform this step only if you create a failover data service.

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

5 **(Optional) Create a resource for the Apache Tomcat Logical Hostname.**

Note – Perform this step for a failover data service.

```
# 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.**

Note – Perform this step for a failover data service.

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

7 **(Optional) Create the resource group for the multiple masters data service.**

Note – Perform this step only if you create a 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.

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

Note – Perform this step only if you create a multiple masters data service.

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

9 **Enable the multiple master resource group that now includes the Apache Tomcat Disk Storage and Logical Hostname resources.**

Note – Perform this step only if you create a multiple masters data service.

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

10 **Create a failover resource group for the SharedAddress resource.**

Note – Perform this step only if you create a scalable data service.

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

11 Create the SharedAddress resource.

Note – Perform this step only if you create a scalable data service.

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

12 Online the SharedAddress resource group.

Note – Perform this step only if you create a scalable data service.

```
# clresourcegroup online -emM   Apache-Tomcat-SharedAddress-resource-group
```

13 Create the resource group for the scalable service.

Note – Perform this step only if you create a scalable data service.

```
# clresourcegroup create \
-p Maximum_primaries=2 \
-p Desired_primaries=2 \
-p RG_dependencies=Apache-Tomcat-SharedAddress-resource-group \
Apache-Tomcat-Scalable-resource-group
```

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

14 Create a resource for the Apache Tomcat Disk Storage if it is not in the root file system.

Note – Perform this step only if you create a scalable data service.

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

15 Enable the failover resource group that now includes the Apache Tomcat Disk Storage and Logical Hostname resources.

Note – Perform this step only if you create a scalable data service.

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

How to Install and Configure Apache Tomcat in the Global Zone

1 Determine how Apache Tomcat will be deployed in Oracle Solaris Cluster

- Determine whether you will use Apache Tomcat as a failover or a multiple master, or a scalable data service.

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

2 Become superuser or assume a role that provides `solaris.cluster.modify` and `solaris.cluster.admin` RBAC authorizations.

3 Create a user and a group if required.

Caution – In this scenario, the deployment of the 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
```

4 If you are not logged in as root, switch to the appropriate user name.

```
# su – user-name
```

5 **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 *Oracle Solaris Cluster Data Services Planning and Administration Guide*.

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 master 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.

6 **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. 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 them on the local file system of every node or on the shared storage. These scripts must be the same on the various nodes.

Example 1–1 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 1–2 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.

How to Enable Apache Tomcat to run in Failover Zone Configuration

1. **Become superuser or assume a role that provides solaris.cluster.admin RBAC authorization on one of the nodes in the cluster that will host Apache Tomcat.**

2. **As superuser register the `SUNW.HAStoragePlus` and the `SUNW.gds` resource types.**

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

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

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

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

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

5. **(Optional) If you want protection against a total adapter failure for your public network, create a resource for the Apache Tomcat's logical hostname.**

   ```
   # clreslogicalhostname create -g Apache Tomcat-resource-group \
   -h logical-hostname Apache Tomcat-logical-hostname-resource-name
   ```

6. **Place the resource group in the managed state.**

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

7. **Install the zone.**

   Install the zone according to the Oracle Solaris Cluster HA for Solaris Zones agent documentation, assuming that the resource name is *Apache Tomcat-zone-rs* and that the zone name is *Apache Tomcat-zone*.

8. **Verify the zone's installation.**

   ```
   # zoneadm -z Apache Tomcat-zone boot
   # zoneadm -z Apache Tomcat-zone halt
   ```

9 **Register the zone's boot component.**

a. **Copy the zone resource boot component configuration file.**

```
# cp /opt/SUNWsczone/sczbt/util/sczbt_config zones-target-configuration-file
```

b. **Use a text editor to set the following variables:**

```
RS=Apache Tomcat-zone-rs
RG=Apache Tomcat-resource-group
PARAMETERDIR=Apache Tomcat-zone-parameter-directory
SC_NETWORK=true|false
SC_LH=Apache Tomcat-logical-hostname-resource-name
FAILOVER=true|false
HAS_RS=Apache Tomcat-has-resource
Zonename=Apache Tomcat-zone
Zonebootopt=zone-boot-options
Milestone=zone-boot-milestone
Mounts=
```

c. **Create the parameter directory for your zone's resource.**

```
# mkdir Apache Tomcat-zone-parameter-directory
```

d. **Execute the Oracle Solaris Cluster HA for Solaris Zone's registration script.**

```
# /opt/SUNWsczone/sczbt/util/sczbt_register -f zones-target-configuration-file
```

e. **Enable the Solaris Zone resource.**

```
# clresource enable Apache Tomcat-zone-rs
```

10 **Enable the resource group.**

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

▼ How to Install and Configure Apache Tomcat in a Failover Zone

1 **Determine how Apache Tomcat will be deployed in Oracle Solaris Cluster**

- Determine whether you will use Apache Tomcat as a failover or a multiple master, or a scalable data service.

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

2 **Become superuser or assume a role that provides `solaris.cluster.modify` and `solaris.cluster.admin` RBAC authorizations.**

3 **Create user and group if required.**

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
```

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

```
# su – user-name
```

5 **Install Apache Tomcat.**

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

It is recommended that you 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 *Oracle Solaris Cluster Data Services Planning and Administration Guide*.

If you plan to deploy Apache Tomcat as a failover data service, install the Apache Tomcat binaries on the shared storage on **one** node or in the failover zone. If Apache Tomcat will be deployed as a multiple master data service, install the Apache Tomcat binaries on the local storage in **every** target zone 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 multiple masters data service, repeat the following step at every node or target zone that will host Apache Tomcat.

6 **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 zone that can host the Apache Tomcat data service. For a failover configuration, store them on the shared storage of the node or in the target zone. For a multiple masters configuration, store them on the local file system or on the shared storage of the target zone. These scripts must be the same on the various zones. For an installation in a failover zone, any place where the Tomcat user has access is appropriate.

Example 1–3 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 1–4 C shell script to set the environment variables for Apache Tomcat 6.0.28

```
# 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 the Apache Tomcat.

Switch to the Tomcat user name in the target zone (in the following example, it is root) and change to the directory where the environment script is located. In the following example the Tomcat version is 6.0.28.

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-6.0.28
Using CATALINA_HOME:   /tomcat/jakarta-tomcat-6.0.28
Using CATALINA_TMPDIR: /tomcat/jakarta-tomcat-6.0.28/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 Tomcat home page, everything is working correctly.

3 **Stop the Apache Tomcat.**

```
 # ./shutdown.sh
Using CATALINA_BASE:   /tomcat/jakarta-tomcat-6.0.28
Using CATALINA_HOME:   /tomcat/jakarta-tomcat-6.0.28
Using CATALINA_TMPDIR: /tomcat/jakarta-tomcat-6.0.28/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 Tomcat hosts.

1 **Become superuser or assume a role that provides `solaris.cluster.modify` and `solaris.cluster.admin` RBAC authorizations.**

2 **Modify the configuration files `server.xml` and the 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 29.**

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 29.**

The port on which Apache Tomcat is serving is required for the configuration of the Oracle Solaris Cluster 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 Oracle Solaris Cluster Apache Tomcat HA Parameter File" on page 46 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, become superuser.**

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

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

For information about setting the `solaris` publisher, see "Set the Publisher Origin To the File Repository URI" in *Copying and Creating Oracle Solaris 11 Package Repositories*.

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

Registering and Configuring Oracle Solaris Cluster HA for Apache Tomcat

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

▼ How to Register and Configure Oracle Solaris Cluster HA for Apache Tomcat as a Failover Data Service

This procedure assumes that you installed the data service packages.

If you did not install the Oracle Solaris Cluster HA for Apache Tomcat packages, go to "Installing the HA for Apache Tomcat Package" on page 31.

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 the Oracle Solaris Cluster HA for Apache Tomcat.**

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

Edit the parameter file `pfile` and follow the comments within that file. 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.

Repeat Step 1 for every Apache Tomcat instance you need.

2 Create and register each required Apache Tomcat component.

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

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

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

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

Repeat these steps for each Apache Tomcat instance you need.

4 **Enable each Apache Tomcat resource.**

Repeat this step for each Apache Tomcat resource created in the step 10.

```
# clresource status

# clresource enable Apache-Tomcat-resource
```

How to Register and Configure Oracle Solaris Cluster HA for Apache Tomcat as a Multiple Masters Data Service

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

If you did not install the Oracle Solaris Cluster 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 31.

1 **Become superuser or assume a role that provides `solaris.cluster.modify` and `solaris.cluster.admin` RBAC 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
```

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

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

3 Create and register each required Apache Tomcat component.

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

Edit the `sctomcat_config` file and follow the comments within that file. 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-has-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-has-resource is created. The ZONE, ZONE_BT, and PROJECT variables are needed only if the multiple masters container resources are zones managed by the zone agent.

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

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

Repeat Steps 7, 8, and 9 for every Apache Tomcat instance you need.

5 **Enable each Apache Tomcat resource.**

Repeat this step for each Apache Tomcat resource created in the previous step.

```
# clresource status

# clresource enable Apache-Tomcat-resource
```

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

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

If you did not install the Oracle Solaris Cluster 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 31.

1 **Become superuser or assume a role that provides `solaris.cluster.modify` and `solaris.cluster.admin` RBAC 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
```

Edit the parameter file pfile and follow the comments within that file. 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
```

This example for Apache Tomcat 6.0.28 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-6.0.28
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.

Do not use the SharedAddress here.

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.

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

3 Create and register each required Apache Tomcat component.

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

Edit the `sctomcat_config` file and follow the comments within that file, 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-has-resource
```

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

4 After editing `sctomcat_config`, register the resource.

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

Repeat these steps for every Apache Tomcat instance you need.

5 **Enable each Apache Tomcat resource.**

Repeat this step for each Apache Tomcat resource created in the previous step.

```
# clresource status

# clresource enable Apache-Tomcat-resource
```

Configuring Oracle Solaris Cluster HA for Apache Tomcat in a Failover Zone

This section describes the additional steps that are necessary to configure Oracle Solaris Cluster HA for Apache Tomcat in a failover zone.

Configuring Oracle Solaris Cluster HA for Apache Tomcat in a Failover Zone

▼ How to Register Oracle Solaris Cluster HA for Apache Tomcat in a Failover Zone

This procedure uses the configuration file to register Oracle Solaris Cluster HA for Apache Tomcat in a failover zone.

Before You Begin Before you register Oracle Solaris Cluster HA for Apache Tomcat in a failover zone, perform the following tasks:

- Install Oracle Solaris Cluster HA for Zones and Oracle Solaris Cluster HA for Apache Tomcat data service packages in the global zone of each cluster node that you want to run the failover zone.
- Create the failover zone.
- Configure the boot component to control your failover zone.

Follow the procedures in the *Oracle Solaris Cluster Data Service for Oracle Solaris Zones Guide*.

1 **Become superuser or assume a role that provides `solaris.cluster.modify` and `solaris.cluster.admin` RBAC authorizations.**

2 **Make a copy of the `/opt/SUNWsctomcat/util/sctomcat_config` file to use as your configuration file.**

You can place this configuration file in any directory that is accessible to the failover zone.

3 **Register Oracle Solaris Cluster HA for Apache Tomcat with SMF.**

```
# ksh /opt/SUNWsctomcat/util/sctomcat_register -f filename
```

`-f` Specifies the configuration file name.

filename The configuration file name.

The registration of Oracle Solaris Cluster HA for Apache Tomcat generates an SMF manifest and registers an SMF service in the failover zone that is specified by the `ZONE` variable in the configuration file. The manifest is named *resourcename*`.xml`, where *resource* is the name that is specified by the `RS` variable in the configuration file. This manifest is placed in the `/var/svc/manifest/application/sczone-agents/` directory. The SMF service is named `svc:/application/sczone-agents/`*resource*.

The Oracle Solaris Cluster HA for Apache Tomcat data service uses the SMF component of the Oracle Solaris Cluster HA for Zones data service to control and probe this SMF service of the Oracle Solaris Cluster HA for Apache Tomcat. It registers an Oracle Solaris Cluster HA for Zones SMF component on top of the already-created SMF service. The name of the script that starts and stops the manifest, as well as probes the SMF component, is `control_sctomcat`.

How to Modify Parameters in the Oracle Solaris Cluster HA for Apache Tomcat Manifest

Perform this task to change parameters in the Oracle Solaris Cluster HA for Apache Tomcat manifest and to validate the parameters in the failover zone. Parameters for the Oracle Solaris Cluster HA for Apache Tomcat manifest are stored as properties of the SMF service. To modify parameters in the manifest, change the related properties in the SMF service and then validate the parameter changes.

1. **Become superuser or assume a role that provides `solaris.cluster.modify` and `solaris.cluster.admin` RBAC authorizations on the zones console.**

2. **Change the Solaris Service Management Facility (SMF) properties for the Oracle Solaris Cluster HA for Apache Tomcat manifest.**

   ```
   # svccfg svc:/application/sczone-agents:resource
   ```

   For more information, see the `svccfg(1M)` man page.

3. **Validate the parameter changes.**

   ```
   # /opt/SUNWsctomcat/bin/control_sctomcat validate resource
   ```

   Messages for this command are stored in the `/var/adm/messages/` directory of the failover zone.

4. **Disconnect from the failover zone's console.**

▼ How to Remove an Oracle Solaris Cluster HA for Apache Tomcat Resource From a Failover Zone

1. **Become superuser or assume a role that provides `solaris.cluster.modify` and `solaris.cluster.admin` RBAC authorizations.**

2. **Disable and remove the resource that is used by the Oracle Solaris Cluster HA for Apache Tomcat data service.**

   ```
   # clresource disable resource
   # clresource delete resource
   ```

3. **Log in as superuser to the failover zone's console.**

4. **Unregister Oracle Solaris Cluster HA for Apache Tomcat from the Solaris Service Management Facility (SMF) service.**

   ```
   # /opt/SUNWsctomcat/util/sctomcat_smf_remove -f filename
   ```

   `-f` Specifies the configuration file name.

   filename The name of the configuration file that you used to register Oracle Solaris Cluster HA for Apache Tomcat with the SMF service.

   Note – If you no longer have the configuration file that you used to register Oracle Solaris Cluster HA for Apache Tomcat with the SMF service, create a replacement configuration file:

   a. Make a copy of the default file, `/opt/SUNWsctomcat/util/sctomcat_config`.

   b. Set the `ZONE` and `RS` parameters with the values that are used by the data service.

   c. Run the `sctomcat_smf_remove` command and use the `-f` option to specify this configuration file.

   ---

5. **Disconnect from the failover zone's console.**

Verifying the Oracle Solaris Cluster 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 Oracle Solaris Cluster HA for Apache Tomcat Installation and Configuration

1 **Become superuser or assume a role that provides `solaris.cluster.modify` and `solaris.cluster.admin` RBAC 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 Oracle Solaris Cluster Apache Tomcat HA Parameter File

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

Structure of the Apache Tomcat HA parameter file

Apache Tomcat HA 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 1–7. For examples of the parameter file refer to "Registering and Configuring Oracle Solaris Cluster HA for Apache Tomcat" on page 32.

TABLE 1–7 Structure of the Apache Tomcat HA 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.

TABLE 1–7 Structure of the Apache Tomcat HA Parameter File *(Continued)*

Variable	Explanation
Basepath	Basepath is the absolute pathname to the directory where the Tomcat bin directory resides. Typically, it is CATALINA_HOME for 5.x or 6.x.
Host	The Host variable is 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 10-20 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 1–7 can be changed at any time. The only difference is when changes take effect.

The following parameters of the Apache Tomcat HA 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 Apache Tomcat resource.

- `EnvScript`
- `User`
- `Basepath`
- `Startwait`
- `TestUrl`
- `WgetPath`

The following parameters of the Apache Tomcat HA 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-startpage`. If you picked the `wget` method, set `TestUrl` to `http://ipalias:port/default-startpage`. In this case, set the `ReturnString` to a string contained in the start page. This string depends on the deployed Apache Tomcat version; for 5.x and 6.x it 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 `startpage_of_the_application`. If you use the `wget` method, set `TestUrl` to `http://ipalias:port/startpage_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-which-does-not-exists`. 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 Oracle Solaris Cluster HA for Apache Tomcat Fault Monitor

This section describes the Oracle Solaris Cluster 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 *Oracle Solaris Cluster Concepts Guide*.

Resource Properties

The Oracle Solaris Cluster HA for Apache Tomcat fault monitor uses the same resource properties as the resource type `SUNW.gds`. Refer to the `SUNW.gds(5)` 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 Apache Tomcat HA parameter file unless the `TestUrl` parameter is specified in the Tomcat parameter file.
3. Connects to the 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 the 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 you are in a global zone or in a failover zone configuration.

If your operating system is Oracle Solaris 11 and your Apache Tomcat resource is dependent on a Solaris Zone boot component resource, you are in a failover zone configuration.

2 **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 `/etc/syslog.conf` file of the appropriate zone.

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

4 **Confirm that debugging for HA for Apache Tomcat is active.**

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

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

5 **Restart the `syslogd` daemon in the appropriate zone.**

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

6 **Edit the `/opt/SUNWsczone/sczbt/etc/config` file to change the `DEBUG=` variable according to one of the examples:**

- `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 steps 1–6, changing `daemon.debug` to `daemon.notice` and changing the `DEBUG` variable to `DEBUG=`.

◆ ◆ ◆ APPENDIX A

Deployment Example: Installing Apache Tomcat in the Global Zone

This appendix presents a complete example of how to install and configure the Apache Tomcat application and data service in the global zone. 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 Apache Tomcat installation in a failover zone, see Appendix B, "Deployment Example: Installing Apache Tomcat in a Failover Zone."

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 software for SPARC or x86 platforms
- Oracle Solaris Cluster 4.0 core software
- HA for Apache Tomcat
- A minimum of Apache Tomcat version 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.

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 superuser or assume a role that provides `solaris.cluster.admin`, `solaris.cluster.modify`, and `solaris.cluster.read` RBAC authorization.

Installing and Configuring Apache Tomcat on Global Storage in the Global Zone

The tasks you must perform to install and configure Apache Tomcat in the global zone are as follows:

▼ Example: Preparing the Cluster for Apache Tomcat

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

Install the following cluster software components on both nodes.

- Oracle Solaris Cluster core software
- 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 -emM RG-TOM
```

Example: Installing the Apache Tomcat Software on Shared Storage

These steps illustrate how to install the Apache Tomcat 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
```

◆ ◆ ◆ APPENDIX B

Deployment Example: Installing Apache Tomcat in a Failover Zone

This appendix presents a complete example of how to install and configure the Apache Tomcat application and data service in a failover zone. 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 Apache Tomcat installation in a global zone, see Appendix A, "Deployment Example: Installing Apache Tomcat in the Global Zone."

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
- `solarisfz1` – The zone to be failed over

Software Configuration

This deployment example uses the following software products and versions:

- Oracle Solaris 11 software for SPARC or x86 platforms
- Oracle Solaris Cluster 4.0 core software
- HA for Apache Tomcat
- Oracle Solaris Cluster HA for Zones
- A minimum of Apache Tomcat version 5.5.17
- The `wget` binary is available under `/usr/sfw/bin/wget`.
- Your preferred text editor

ZFS is the only supported file system for a failover zone in Solaris 11. This example assumes that you have already installed and established your cluster. The example 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 as superuser or assume a role that provides `solaris.cluster.admin`, `solaris.cluster.modify`, and `solaris.cluster.read` RBAC authorization.

Installing and Configuring Apache Tomcat on Global Storage in the Failover Zone

The tasks you must perform to install and configure Apache Tomcat in the failover zone are as follows:

▼ Example: Preparing the Cluster for Apache Tomcat

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

  Install the following cluster software components on both nodes.

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

▼ Example: Configuring Cluster Resources for Apache Tomcat

1. **Register the necessary resource types.**

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

2 Create the failover resource group.

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

3 Create a zpool on a shared disk to host the zonepath for the failover zone.

```
phys-schost-1# zpool create -m /ha-zones ha-zones \
/dev/rdsk/c0t60080E500017B52C00002B9D4EB40DB5d0
phys-schost-1# zpool export ha-zones
```

In a failover configuration, the zone's zonepath must be on a highly available file system.

4 Register the zpool with the `HAStoragePlus` resource and create a resource (of type HASP) for the zones' disk storage (the zpool).

```
phys-schost-1# clresource create -g zone-rg -t SUNW.HAStoragePlus -p \
zpools=ha-zones ha-zones-hasp-rs
phys-schost-1# clresourcegroup online -emM -n phys-schost-1 zone-rg
phys-schost-1# zfs create ha-zones/solaris
```

▼ Example: Creating and Configuring the Failover Zone

1 Create and configure the zone on all nodes that can host this failover zone.

The zpool hosting the zonepath must be on a shared disk. For a two-node cluster, the zone configuration must be executed on both nodes. Following is an example using the phys-schost-1 node. Perform the same actions on the phys-schost-2 node.

```
phys-schost-1# zonecfg -z solarisfz1 \
'create -b;
set zonepath=/ha-zones/solaris/solarisfz1;
set autoboot=false; set ip-type=shared;
add attr; set name=osc-ha-zone; set type=boolean; set value=true; end;
add net; set address=zone-hostname ; set physical=sc_ipmp0; end;'
```

2 Verify that the node is configured on both nodes.

```
phys-schost-1# zoneadm list -cv
 ID NAME             STATUS       PATH                           BRAND    IP
  0 global           running       /                             solaris  shared
  - solarisfz1       configured    /ha-zones/solaris/solarisfz1  solaris  shared

phys-schost-2# zoneadm list -cv
 ID NAME             STATUS       PATH                           BRAND    IP
  0 global           running       /                             solaris  shared
  - solarisfz1       configured    /ha-zones/solaris/solarisfz1  solaris  shared
```

3 Install the zone on phys-schost-1, which is where the ha-zones zpool is online.

```
phys-schost-1:~# zoneadm -z solarisfz1 install
Progress being logged to /var/log/zones/zoneadm.20030401T184050Z.solarisfz1.install
    Image:       Preparing at /ha-zones/solaris/solarisfz1/root.
    Install Log: /system/volatile/install.3349/install_log
    AI Manifest: /tmp/manifest.xml.QGa4Gg
    SC Profile:  /usr/share/auto_install/sc_profiles/enable_sci.xml
```

```
    Zonename:    solarisfz1
Installation:    Starting ...

    Creating IPS image
    Installing packages from:
       solaris
         origin:  http://pkg.oracle.com/solaris/release/
       ha-cluster
         origin:  http://localhost:1008/ha-cluster/2c76b8fe7512dde39 \
                  c04c11f28f6be4603f39c66/
DOWNLOAD                                  PKGS       FILES    XFER (MB)
Completed                               167/167 32062/32062  175.8/175.8$<3>

PHASE                                        ACTIONS
Install Phase                              44313/44313

PHASE                                          ITEMS
Package State Update Phase                   167/167
Image State Update Phase                         2/2
Installation: Succeeded

        Note: Man pages can be obtained by installing pkg:/system/manual. Done.
        Done: Installation completed in 550.217 seconds.
Next Steps: Boot the zone, then log into the zone console (zlogin -C)to complete
the configuration process.
Log saved as /ha-zones/solaris/solarisfz1/
root/var/log/zones/zoneadm.20030401T184050Z.solarisfz1.install.
```

4 Verify that the zone was successfully installed and can boot up on phys-schost-1.

a. Verify that the zone was installed.

```
phys-schost-1# zoneadm list -cv
  ID NAME             STATUS      PATH                           BRAND    IP
   0 global           running       /                            solaris  shared
   - solarisfz1       installed     /ha-zones/solaris/solarisfz1 solaris  shared
```

b. In a different window (for example, from an ssh, rlogin, or telnet window), log into the zone's console and boot the zone.

```
phys-schost-1# zlogin -C solarisfz1
phys-schost-1# zoneadm -z solarisfz1 boot
```

c. Follow the prompts in the interactive screens to configure the zone.

d. Shut down the zone and switch the resource group to another node in the resource group nodelist.

```
phys-schost-1# zoneadm -z solarisfz1 shutdown
phys-schost-1# clresourcegroup switch -n phys-schost-2 zone-rg
phys-schost-1# zoneadm -z solarisfz1 detach -F
phys-schost-1# zoneadm list -cv
  ID NAME             STATUS      PATH                           BRAND    IP
   0 global           running       /                            solaris  shared
   - solarisfz1       configured    /ha-zones/solaris/solarisfz1 solaris  shared
```

5. **Assign the universally unique identifier (UUID) for the active boot environment (BE) from the first node, phys-schost-1, to the active BE on the second node, phys-schost-2.**

   a. **Get the UUID for the active BE on phys-schost-1.**

   ```
   phys-schost-1:~# beadm list -H
   b175b-fresh;70db96a2-5006-c84e-da77-f8bd430ba914;;;64512;static;1319658138
   s11_175b;b5d7b547-180d-467e-b2c4-87499cfc1e9d;NR;/;8000659456;static;1319650094
   s11_175b-backup-1;aba7a813-feb9-e880-8d7b-9d0e5bcd09af;;;166912;static;1319658479

   phys-schost-2:~# beadm list -H
   b175b-fresh;c37d524b-734a-c1e2-91d9-cf460c94110e;;;65536;static;1319471410
   s11_175b;1d0cca6d-8599-e54a-8afa-beb518b1d87a;NR;/;8096948224;static;1319293680
   s11_175b-backup-1;db2b581a-ea82-6e8c-9a3d-c1b385388fb7;;;167936;static;1319472971
   ```

   b. **Set the UUID for the active BE of the global zone on phys-schost-2 to be the same as phys-schost-1. The active BE has flag *N* in the third field separated by a semicolon. The UUID is set on the data set of the BE. You can get the UUID by running `df -b /`.**

   ```
   phys-schost-2:~# df -b /
   Filesystem                      avail
   rpool/ROOT/s11_175b   131328596
   root@vzoolah3a:/#
   phys-schost-2:~# zfs set org.opensolaris.libbe:uuid=b5d7b547-180d-467e-b2c4 \
   -87499cfc1e9d \
   rpool/ROOT/s11_175b
   ```

6. **Attach the zone and verify the zone can boot on the second node.**

   a. **Attach the zone.**

   ```
   phys-schost-2# zoneadm -z solarisfz1 attach -F
   ```

   b. **From another session, connect to the zone console.**

   ```
   phys-schost-2# zlogin -C solarisfz1
   ```

   c. **Boot the zone and observe the boot messages on the console.**

   ```
   phys-schost-2# zoneadm -z solarisfz1 boot
   ```

7. **If the boot up succeeded, shut down and detach the zone.**

   ```
   phys-schost-2# zoneadm -z solarisfz1 shutdown
   phys-schost-2# zoneadm -z solarisfz1 detach -F
   ```

8. **On both nodes, install the failover container agent if it is not already installed.**

   The following example shows how to install the agent on phys-schost-1.

   ```
   phys-schost-1# pkg install ha-cluster/data-service/ha-zones
   ```

9 Create the resource from any one node and set the parameters on both nodes.

Steps a and b show these steps performed on phys-schost-1.

a. Register the resource

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

b. On both nodes, edit the `sczbt_config` file and set the parameters.

```
phys-schost-1# cd /opt/SUNWsczone/sczbt/util
phys-schost-1# cp -p sczbt_config sczbt_config.solarisfz1-rs
phys-schost-1# vi sczbt_config.solarisfz1-rs
RS=solarisfz1-rs
RG=zone-rg
PARAMETERDIR=/ha-zones/solaris/solarisfz1/params
SC_NETWORK=false
SC_LH=
FAILOVER=true
HAS_RS=ha-zones-hasp-rs
Zonename="solarisfz1"
Zonebrand="solaris"
Zonebootopt=""
Milestone="svc:/milestone/multi-user-server"
LXrunlevel="3"
SLrunlevel="3"
Mounts=""
```

c. On phys-schost-2, create the `params` directory that appears in the `sczbt_config` file.

```
phys-schost-2# mkdir /ha-zones/solaris/solarisfz1/params
```

d. On one node, configure the zone-boot resource.

The resource is configured with the parameters that you set in the `sczbt_config` file.

```
phys-schost-2# ./sczbt_register -f ./sczbt_config.solarisfz1-rs
```

e. On one node, enable the failover zone resource that was created.

```
phys-schost-2# clresource enable solarisfz1-rs
```

f. On one node, check the status of the resource groups and resources.

```
phys-schost-2# clresource status -g zone-rg
=== Cluster Resources ===

Resource Name         Node Name      State      Status Message
-------------------   -------------  -----      -------------------
solarisfz1-rs         phys-schost-1  Offline    Offline
                      phys-schost-2  Online     Online

ha-zones-hasp-rs      phys-schost-1  Offline    Offline
                      phys-schost-2  Online     Online
```

g. Verify that the zone successfully boots up and then switch to the other node to test the switchover capability.

```
phys-schost-2# clresourcegroup switch -n phys-schost-1 zone-rg
phys-schost-2# clresource status -g zone-rg
```

```
=== Cluster Resources ===

Resource Name         Node Name      State       Status Message
-------------------   -------------  -----       -------------------
solarisfz1-rs         phys-schost-1  Online      Online
                      phys-schost-2  Offline     Offline

ha-zones-hasp-rs      phys-schost-1  Online      Online
                      phys-schost-2  Offline     Offline
```

h. Verify that the zone successfully switched over to the other node.

```
phys-schost-1# zlogin -C solarisfz1
```

▼ Example: Installing the Apache Tomcat Software on Shared Storage

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

1 Log in to the zone.

```
phys-schost-1# zlogin solarisfz1
```

2 Add the `tomcat` group and user.

```
zone# groupadd -g 1000 tomcat
zone# useradd -g 1000 -d /tomcat -m -s /bin/ksh tomcat
```

3 Install the Apache Tomcat binaries.

```
zone# su - tomcat
zone# gzcat apache-tomcat-5.5.17.tar.gz|tar xvf -
zone# gzcat apache-tomcat-5.5.17-compat.tar.gz|tar xvf -
```

4 Create your environment script `/tomcat/env.ksh` with the following contents.

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

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

```
zone# cp /opt/SUNWsctomcat/bin/pfile /tomcat
zone# chown tomcat:tomcat /tomcat/pfile
```

6 Add this cluster's information to the parameter file `/tomcat/pfile`.

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

```
EnvScript=/tomcat/env.ksh
User=tomcat
Basepath=/tomcat/apache-tomcat-5.5.17
```

```
Host=zone-hostname
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
```

7 Save and close the file.

8 Leave the zone.

▼ Example: Modifying the Apache Tomcat Configuration Files

1 Ensure that the Apache Tomcat data service package was installed.

2 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
```

3 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=zone-rg
PORT=8080
LH=zone-hostname
NETWORK=true
SCALABLE=false
PFILE=/tomcat/pfile
HAS_RS=ha-zones-hasp-rs
ZONE=solarisfz1
ZONE_BT=solarisfz1-rs
PROJECT=
```

4 Save and close the file.

▼ 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

O

P

R

S

T