Sun Cluster Data Service for Apache Tomcat Guide for Solaris OS
How to Verify the Sun Cluster HA for Apache Tomcat Installation and Configuration 27

Understanding the Sun Cluster Apache Tomcat HA Parameter File 28
  Structure of the Apache Tomcat HA parameter file 28
  Strategy to Chose the TestCmd and the ReturnString Variable 29

Understanding Sun Cluster HA for Apache Tomcat Fault Monitor 30
  Resource Properties 30
  Probing Algorithm and Functionality 30

How to debug Sun Cluster HA for Apache Tomcat 31

How to turn debug on for Sun Cluster HA for Apache Tomcat 31

Index 33
Installing and Configuring Sun Cluster HA for Apache Tomcat

Accessing Sun Documentation Online
The docs.sun.comSM Web site enables you to access Sun technical documentation online. You can browse the docs.sun.com archive or search for a specific book title or subject. The URL is http://docs.sun.com.

Ordering Sun Documentation
Sun Microsystems offers select product documentation in print. For a list of documents and how to order them, see “Buy printed documentation” at http://docs.sun.com.

Typographic Conventions
The following table describes the typographic changes used in this book.
### TABLE 1-1 Typographic Conventions

<table>
<thead>
<tr>
<th>Typeface or Symbol</th>
<th>Meaning</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>AaBbCc123</td>
<td>The names of commands, files, and directories; on-screen computer output</td>
<td>Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. <code>machine_name%</code> you have mail.</td>
</tr>
<tr>
<td>AaBbCc123</td>
<td>What you type, contrasted with on-screen computer output</td>
<td><code>machine_name% su</code> &lt;br&gt; <code>Password:</code></td>
</tr>
<tr>
<td>AaBbCc123</td>
<td>Command-line placeholder: replace with a real name or value</td>
<td>To delete a file, type <code>rm filename</code>.</td>
</tr>
<tr>
<td>AaBbCc123</td>
<td>Book titles, new words, or terms, or words to be emphasized.</td>
<td>Read Chapter 6 in <em>User's Guide</em>. These are called <em>class options</em>. You must be <em>root</em> to do this.</td>
</tr>
</tbody>
</table>

### Shell Prompts in Command Examples

The following table shows the default system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

**TABLE 1-2 Shell Prompts**

<table>
<thead>
<tr>
<th>Shell</th>
<th>Prompt</th>
</tr>
</thead>
<tbody>
<tr>
<td>C shell prompt</td>
<td><code>machine_name%</code></td>
</tr>
<tr>
<td>C shell superuser prompt</td>
<td><code>machine_name#</code></td>
</tr>
<tr>
<td>Bourne shell and Korn shell prompt</td>
<td><code>$</code></td>
</tr>
<tr>
<td>Bourne shell and Korn shell superuser prompt</td>
<td><code>#</code></td>
</tr>
</tbody>
</table>

### Installing and Configuring Sun Cluster HA for Apache Tomcat

Table 1–3 lists the tasks for installing and configuring Sun Cluster HA for Apache Tomcat. Perform these tasks in the order they are listed.
### TABLE 1-3 Task Map: Installing and Configuring Sun Cluster HA for Apache Tomcat

<table>
<thead>
<tr>
<th>Task</th>
<th>For Instructions, Go To</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Plan the installation.</td>
<td>“Planning the Sun Cluster HA for Apache Tomcat Installation and Configuration” on page 8</td>
</tr>
<tr>
<td>2 Install and configure Apache Tomcat.</td>
<td>“How to Install and Configure Apache Tomcat” on page 14</td>
</tr>
<tr>
<td>3 Verify installation and configuration.</td>
<td>“How to Verify the Installation and Configuration of Apache Tomcat” on page 17</td>
</tr>
<tr>
<td>4 Install Sun Cluster HA for Apache Tomcat Packages.</td>
<td>“Installing the Sun Cluster HA for Apache Tomcat Packages” on page 18</td>
</tr>
<tr>
<td>5.1 Register and Configure Sun Cluster HA for Apache Tomcat as a failover data service.</td>
<td>“How to Register and Configure Sun Cluster HA for Apache Tomcat as a failover data service” on page 21</td>
</tr>
<tr>
<td>5.2 Register and Configure Sun Cluster HA for Apache Tomcat as a scalable data service.</td>
<td>“How to Register and Configure Sun Cluster HA for Apache Tomcat as a Scalable or Multiple Masters Data Service” on page 24</td>
</tr>
<tr>
<td>6 Verify Sun Cluster HA for Apache Tomcat Installation and Configuration.</td>
<td>“How to Verify the Sun Cluster HA for Apache Tomcat Installation and Configuration” on page 27</td>
</tr>
<tr>
<td>7 Understanding the Sun Cluster Apache Tomcat HA parameter file.</td>
<td>“Understanding the Sun Cluster Apache Tomcat HA Parameter File” on page 28</td>
</tr>
<tr>
<td>8 Understanding the Sun Cluster HA for Apache Tomcat Fault Monitor.</td>
<td>“Understanding Sun Cluster HA for Apache Tomcat Fault Monitor” on page 30</td>
</tr>
<tr>
<td>9 How to debug Sun Cluster HA for Apache Tomcat.</td>
<td>“How to turn debug on for Sun Cluster HA for Apache Tomcat” on page 31</td>
</tr>
</tbody>
</table>

---

## Sun 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](http://jakarta.apache.org).

The Sun 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 Sun Cluster data service.

**TABLE 1-4 Protection of Components**

<table>
<thead>
<tr>
<th>Component</th>
<th>Protected by</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apache Tomcat</td>
<td>Sun Cluster HA for Apache Tomcat</td>
</tr>
</tbody>
</table>

Planning the Sun Cluster HA for Apache Tomcat Installation and Configuration

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

About Horizontal Scalability

Sun 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 loadbalancers. For a more detailed discussion of this scalable service see the *Sun Cluster Concepts Guide*.

Before using the Sun Cluster HA for Apache Tomcat in a scalable configuration, it is necessary to have a closer look at 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 it means, from the load-balancing point of view, that the source ip address is changing. This will spoil every ip based load-balancing, wether it is hardware or software.

Especially when your client accesses the server over the Internet, it is not guaranteed, that the source ip address remains the same during a session context.

Sun Cluster HA for Apache Tomcat can be configured two ways to get horizontal scalability.

1. Using Sun Cluster HA for Apache Tomcat in a purely scalable configuration.
2. Using Sun Cluster HA for Apache Tomcat in a multiple masters configuration.

Multiple masters configurations and scalable configurations are configured as a scalable data service. 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, Sun 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 the Sun Cluster Concepts Guide.

Both configuration options are discussed in the sections “Sun Cluster HA for Apache Tomcat as a scalable configuration” on page 9 and “Sun Cluster HA for Apache Tomcat as a multiple master configuration” on page 9.

Sun Cluster HA for Apache Tomcat as a scalable configuration

You can use Sun 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.

- You can achieve the session replication for Apache Tomcat 4.x by following the instructions at http://www.theserverside.com/resources/article.jsp?i=Tomcat.
- For Apache Tomcat 5.x, inbound memory session replication as a built-in feature, is on the feature list.
- For Apache Tomcat 3.x the inbound memory session replication is not configurable.

Sun Cluster HA for Apache Tomcat as a multiple master configuration

You can use Sun 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 handle the session context.
The difference between scalable and multiple masters configuration is only in the way the clients access the cluster nodes. In a scalable configuration, they access the shared address. Otherwise the clients access the physical hostnames. In items of Sun Cluster point of view there is no difference between the two configurations.

Configuration Restrictions

This section provides a list of software and hardware configuration restrictions that apply to Sun Cluster HA for Apache Tomcat only.

For restrictions that apply to all data services, see the Sun Cluster Release Notes.

- Deploy a scalable Sun 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.

- Setting the resource parameter Load_balancing_policy to LB_STICKY is strictly required, if Sun 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 the more cache hits out of your caches.

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

Configuration Requirements

These requirements apply to Sun Cluster HA for Apache Tomcat only. You must meet these requirements before you proceed with your Sun 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.

- Configure the Tomcat Home directory on local storage for a scalable or multiple masters configuration – If you intend to install Apache Tomcat in a scalable resource group, you should create the Tomcat Home directory and it’s dynamic data onto your local Storage.

  This required because Apache Tomcat uses the directory structure to store its configuration, logs, deployed applications and so on.
If your local storage is not big 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.

- **Deploy Sun Cluster HA for Apache Tomcat multiple master configuration** – If you intend to install Apache Tomcat in a multiple master configuration, an external load balancer is required.

- **Configure the Apache Tomcat Home directory on shared storage for a failover configuration** – 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.

- **Apache Tomcat component and dependencies** – You can configure the Sun 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 1–5 or in Table 1–6.

**TABLE 1–5** Apache Tomcat component and its dependencies for a failover configuration (via → symbol)

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage Resource</td>
<td>This resource can be a SUNW.HAStoragePlus or a SUNW.HAStorage resource type.</td>
</tr>
<tr>
<td>Apache Tomcat (Mandatory)</td>
<td>→ SUNW.HAStoragePlus / SUNW.HAStorage resource</td>
</tr>
<tr>
<td></td>
<td>The Storage resource manages the Apache Tomcat File System Mount points and ensures that Apache Tomcat is not started until they are mounted.</td>
</tr>
</tbody>
</table>
Logical Host (Mandatory) | The Apache Tomcat has an implicit dependency on the logical hosts of the resource group, because it is based on the generic data service. On behalf of this reason a logical Host needs to exist, but no explicit dependency is necessary.

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 are stored on the root file system. Otherwise follow Table 1–6.

### TABLE 1–6 Apache Tomcat component and its dependencies for a scalable or multiple masters configuration (via → symbol)

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage Resource</td>
<td>This resource can be a SUNW.HAStorage or a SUNW.HAStoragePlus resource type.</td>
</tr>
<tr>
<td>Apache Tomcat (Mandatory) → Storage resource</td>
<td>The Storage resource manages the Apache Tomcat File System Mount points and ensures that Apache Tomcat is not started until they are mounted.</td>
</tr>
</tbody>
</table>
| shared address (Mandatory) | The shared address resource resides in a separate resource group on which the scalable resource group depends.

Note – For more detailed information about Apache Tomcat, refer to jakarta.apache.org web page.

- Apache Tomcat component has configuration and a registration files in the directory /opt/SUNWstomcat/util — These files let you register the Apache Tomcat component with Sun Cluster.
  Within these files, the appropriate dependencies have been applied.

**EXAMPLE 1–1** Apache Tomcat configuration and registration file for Sun Cluster

```bash
# cd /opt/SUNWstomcat
# # ls -l util
total 4
-rwx--x--- 1 root other 709 Apr 29 11:57 sctomcat_config
-rwx--x--- 1 root other 561 Apr 29 11:58 sctomcat_register
# more util/*
::::::::::::::
util/sctomcat_config
```
EXAMPLE 1–1 Apache Tomcat configuration and registration file for Sun Cluster
(Continued)

::: :::::
# # Copyright 2003 Sun Microsystems, Inc. All rights reserved.
# Use is subject to license terms.
# # 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 or false for a scalable or failover resource
# # PFILE - absolute pathname 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)
#
RS=
RG=
PORT=8080
LH=
SCALABLE=false
PFILE=
HAS_RS=
::: :::::
util/sctomcat_register
::: :::::
# # Copyright 2003 Sun Microsystems, Inc. All rights reserved.
# Use is subject to license terms.
#
. `dirname $0`/sctomcat_config
scrgadm -a -j $RS -g $RG -t SUNW.gds \
-x Start_command="/opt/SUNWstomcat/bin/start_sctomcat \
-R $RS -G $RG -N $PFILE " \
-x Stop_command="/opt/SUNWstomcat/bin/stop_sctomcat \
-R $RS -G $RG -N $PFILE " \
-x Probe_command="/opt/SUNWstomcat/bin/probe_sctomcat \
-R $RS -G $RG -N $PFILE " \
-y Port_list=$PORT/tcp -y Network_resources_used=$LH \
-x Stop_signal=9 \
-y Scalable=$SCALABLE \
-y Resource_dependencies=$HAS_RS

Installing and Configuring Sun Cluster HA for Apache Tomcat 13
Installing and Configuring Apache Tomcat

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

How to Install and Configure Apache Tomcat

1. **Determine how Apache Tomcat will be deployed in Sun Cluster**
   - Determine whether you will use Apache Tomcat as a failover or a scalable data service.
     For conceptual information on scalable and failover data services, see the *Sun 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. **Mount Apache Tomcat Cluster File Systems**

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

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

4. **Switch to the appropriate user name** — If it is not root with `# 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 Sun 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 the *Sun Cluster Data Services Installation and Configuration 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 data service, install the Apache Tomcat binaries on the local storage on **every** node that will host the Apache Tomcat data service.


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 an Korn shell or a Cshell 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 users 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. 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 not be different on the various nodes.

**EXAMPLE 1-2** Korn shell script to set the environment variables for Apache Tomcat 3.x/3.3

```bash
#!/usr/bin/ksh
#
# Environment for Tomcat
#
JAVA_HOME=/usr/j2se
export JAVA_HOME
JAKARTA_HOME=/global/mnt1/jakarta-3.3
export JAKARTA_HOME
TOMCAT_HOME=$JAKARTA_HOME
export TOMCAT_HOME
```

**EXAMPLE 1-3** Cshell script to set the environment variables for Apache Tomcat 3.3/3.3

```bash
#!/usr/bin/csh
#
# Environment for Tomcat
#
```
EXAMPLE 1–3 Cshell script to set the environment variables for Apache Tomcat 3.3/3.3 (Continued)

setenv JAVA_HOME /usr/j2se
setenv JAKARTA_HOME /global/mnt1/jakarta-3.3
setenv TOMCAT_HOME $JAKARTA_HOME

EXAMPLE 1–4 Korn shell script to set the environment variables for Apache Tomcat 4.1.x

```bash
#!/usr/bin/ksh
#
# Environment for Tomcat
#
JAVA_HOME=/usr/j2se
export JAVA_HOME
CATALINA_HOME=/global/mnt1/jakarta-tomcat-4.1.18
export CATALINA_HOME
```

EXAMPLE 1–5 Cshell script to set the environment variables for Apache Tomcat 4.1.x

```csh
#!/usr/bin/csh
#
# Environment for Tomcat
#
setenv JAVA_HOME /usr/j2se
setenv CATALINA_HOME /global/mnt1/jakarta-tomcat-4.1.18
```

The environment variables are version and configuration dependent.

---

**Note** – For more information about Apache Tomcat, refer to [jakarta.apache.org](http://jakarta.apache.org) web page.

---

### 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 following example, it is root) and change to the directory where the environment script is located. In the following example the Tomcat version is 4.1.24.

   # . ./env.ksh
   # cd $CATALINA_HOME/bin
   # . /startup.sh

   Using CATALINA_BASE: /tomcat/jakarta-tomcat-4.1.24
   Using CATALINA_HOME: /tomcat/jakarta-tomcat-4.1.24
   Using CATALINA_TMPDIR: /tomcat/jakarta-tomcat-4.1.24/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 homepage everything is working correctly.

3. Stop the Apache Tomcat.

   # ./shutdown.sh

   Using CATALINA_BASE: /tomcat/jakarta-tomcat-4.1.24
   Using CATALINA_HOME: /tomcat/jakarta-tomcat-4.1.24
   Using CATALINA_TMPDIR: /tomcat/jakarta-tomcat-4.1.24/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 – It is vital for a scalable configuration that the Apache Tomcat configuration does not differ on the Tomcat hosts.
1. Modify the configuration files server.xml and the Tomcat user configuration to the desired content. For further information refer to http://jakarta.apache.org/tomcat/index.html.

2. Start Apache Tomcat as described in “How to Verify the Installation and Configuration of Apache Tomcat” on page 17.


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

5. Stop Apache Tomcat as described in “How to Verify the Installation and Configuration of Apache Tomcat” on page 17.

The port on which Apache Tomcat is serving is required for the configuration of the Sun 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 is described in “Understanding the Sun Cluster Apache Tomcat HA Parameter File” on page 28 can be get /start_page. With this procedure, you monitor Apache Tomcat and the application it is serving.

---

Installing the Sun Cluster HA for Apache Tomcat Packages

If you did not install the Sun Cluster HA for Apache Tomcat packages during your initial Sun Cluster installation, perform this procedure to install the packages. Perform this procedure on each cluster node where you are installing the Sun Cluster HA for Apache Tomcat packages. To complete this procedure, you need the Sun Java Enterprise System Accessory CD Volume 3.

If you are installing more than one data service simultaneously, perform the procedure in “Installing the Software” in Sun Cluster 3.1 10/03 Software Installation Guide.

Install the Sun Cluster HA for Apache Tomcat packages by using one of the following installation tools:

- The Web Start program
- The scinstall utility
Note – The Web Start program is not available in releases earlier than Sun Cluster 3.1 Data Services 10/03.

How to Install the Sun Cluster HA for Apache Tomcat Packages Using the Web Start Program

You can run the Web Start program with a command-line interface (CLI) or with a graphical user interface (GUI). The content and sequence of instructions in the CLI and the GUI are similar. For more information about the Web Start program, see the installer(1M) man page.

1. Become superuser on the cluster node where you are installing the Sun Cluster HA for Apache Tomcat packages.

2. (Optional) If you intend to run the Web Start program with a GUI, ensure that your DISPLAY environment variable is set.

3. Load the Sun Java Enterprise System Accessory CD Volume 3 into the CD-ROM drive.
   If the Volume Management daemon vold(IM) is running and configured to manage CD-ROM devices, it automatically mounts the CD-ROM on the /cdrom/cdrom0 directory.

4. Change to the Sun Cluster HA for Apache Tomcat component directory of the CD-ROM.
   The Web Start program for the Sun Cluster HA for Apache Tomcat data service resides in this directory.
   
   `# cd /cdrom/cdrom0/components/SunCluster_HA_APACHE_TC_3.1`

5. Start the Web Start program.
   
   `# ./installer`

6. When you are prompted, select the type of installation.
   - To install only the C locale, select Typical.
   - To install other locales, select Custom.

7. Follow instructions on the screen to install the Sun Cluster HA for Apache Tomcat packages on the node.
   After the installation is finished, the Web Start program provides an installation summary. This summary enables you to view logs that the Web Start program created during the installation. These logs are located in the /var/sadm/install/logs directory.

8. Exit the Web Start program.
   a. To ensure that the CD-ROM is not being used, change to a directory that does not reside on the CD-ROM.
   b. Eject the CD-ROM.
      
      # eject cdrom

▼ How to Install the Sun Cluster HA for Apache Tomcat Packages Using the scinstall Utility

You need the Sun Cluster Agents CD-ROM to perform this procedure. It is assumed that you did not install the data service during your initial Sun Cluster installation.

If you installed Sun Cluster HA for Apache Tomcat as part of your initial Sun Cluster installation, proceed to “Registering and Configuring Sun Cluster HA for Apache Tomcat” on page 21.

Perform this procedure, on all nodes that can run Sun Cluster HA for Apache Tomcat.

1. Load the Sun Cluster Agents CD-ROM into the CD-ROM drive.
2. Run the scinstall utility with no options.
   The scinstall utility prompts you for additional information.
3. Chose the menu option, Add Support for New Data Service to this Cluster Node
   This step starts the scinstall utility in interactive mode.
4. Provide the pathname to the Sun Cluster Agents CD-ROM.
   The utility refers to the CD as “data services cd.”
5. Chose the menu option, q) done.
6. Type yes for the question, Do you want to see more data services?.
   The utility refers to the CD as “data services cd.”
7. Specify the data service to install.
   The scinstall utility lists the data service that you selected and asks you to confirm your choice.
8. Exit the scinstall utility.
9. Unload the CD from the CD-ROM drive.
Registering and Configuring Sun Cluster HA for Apache Tomcat

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

How to Register and Configure Sun 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 Sun Cluster HA for Apache Tomcat packages, go to “Installing the Sun Cluster HA for Apache Tomcat Packages” on page 18.

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

1. Become superuser on one of the nodes in the cluster that will host Apache Tomcat.

2. Register the SUNW.gds resource type.

   ```
   # scrgadm -a -t SUNW.gds
   ```

3. Register the SUNW.HAStoragePlus resource type.

   ```
   # scrgadm -a -t SUNW.HAStoragePlus
   ```

4. Create a failover resource group.

   ```
   # scrgadm -a -g Apache-Tomcat-failover-resource-group
   ```

5. Create a resource for the Apache Tomcat Disk Storage.

   ```
   # scrgadm -a -j Apache-Tomcat-has-resource \
   > -g Apache-Tomcat-failover-resource-group \
   > -t SUNW.HAStoragePlus \
   > -x FilesystemMountPoints=Apache Tomcat- instance-mount-points
   ```

6. Create a resource for the Apache Tomcat Logical Host name.

   ```
   # scrgadm -a -L -j Apache-Tomcat-IH-resource \
   > -g Apache-Tomcat-failover-resource-group \
   > -l Apache-Tomcat-logical-hostname
   ```

7. Enable the failover resource group, which now includes the Apache Tomcat Disk Storage and Logical Hostname resources.
8. Prepare the parameter file, which is required by the Sun Cluster HA for Apache Tomcat.

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

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

```bash
#!/usr/bin/ksh
# 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. 4.x CATALINA_HOME
#               or TOMCAT_HOME for 3.x
# Host          Hostname to test Apache Tomcat
# Port          Port where Apache Tomcat is configured
# TestCmd       Apache Tomcat test command
# Startwait     Sleeping $Startwait seconds after completion of the # start command

EnvScript=
User=
Basepath=
Host=
Port=8080
TestCmd="get /index.jsp"
ReturnString="CATALINA"
Startwait=20

The following is an example for Apache Tomcat 4.1.24.

```bash
EnvScript=/global/mnt1/tomtest/env.ksh
User=tomtest
Basepath=/global/mnt1/tomtest/jakarta-tomcat-4.1.24
Host=tomcat-lh
Port=8080
TestCmd="get /index.jsp"
ReturnString="CATALINA"
Startwait=20
```

This example is from an Apache Tomcat 4.1.24 configured as standalone web server with the default start page `index.jsp`. The start page contains the string `CATALINA`.

---

**Caution** – The parameter files need to be available on every node that can host the Apache Tomcat data service. For a failover configuration store them on the shared storage. The parameter files cannot differ for an instance of Apache Tomcat on the various nodes.
Repeat Step 8 for every Apache Tomcat instance you need.

9. Create and register each required Apache Tomcat component.

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

Edit the `sctomcat_config` file and follow the comments within that file. For example:

```
# These parameters can be customized in (key=value) form
#
# Copyright 2003 Sun Microsystems, Inc. All rights reserved.
# Use is subject to license terms.
#
# 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 or false for a scalable or failover resource
# PFILE - absolute pathname 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)
#
RS=
RG=
PORT=8080
LH=
SCALABLE=false
PFILE=
HAS_RS=
```

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
HAS_RS=tomcat-hastplus-res
```

10. After editing `sctomcat_config` register the resource.

```
# ./sctomcat_register
```

Repeat Step 7, 8, and 9 for each Apache Tomcat instance you need.

11. Enable each Apache Tomcat resource.

Repeat this step for each Apache Tomcat resource created in the step 10.
How to Register and Configure Sun Cluster HA for Apache Tomcat as a Scalable or Multiple Masters Data Service

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

If you did not install the Sun Cluster HA for Apache Tomcat packages as part of your initial Sun Cluster installation, go to “Installing the Sun Cluster HA for Apache Tomcat Packages” on page 18.

1. Become superuser on one of the nodes in the cluster that will host Apache Tomcat.

2. Register the SUNW.gds resource type.
   
   # scrgadm -a -t SUNW.gds

3. Create a failover resource group for the SharedAddress resource.

   # scrgadm -a -g Apache-Tomcat-SharedAddress-resource-group

4. Create the SharedAddress resource.

   # scrgadm -a -S -j Apache-Tomcat-SharedAddress-resource \
   > -g Apache-Tomcat-SharedAddress-resource-group\ 
   > -l Apache-Tomcat-SharedAddress-hostname

5. Enable the SharedAddress resource group.

   # scswitch -Z -g Apache-Tomcat-SharedAddress-resource-group

6. Prepare the parameter file, which is required by the Sun 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
   # 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. 4.x CATALINA_HOME
The following is an example for Apache Tomcat 4.1.24.

EnvScript=/tomcat/env.ksh
User=tomcat
Basepath=/tomcat/jakarta-tomca-4.1.24
Host=localhost
Port=8080
TestCmd="get /index.jsp"
ReturnString="CATALINA"
Startwait=20

This example is from an Apache Tomcat 4.1.24 configured as standalone web server with the default start page index.jsp. The start page contains the string CATALINA. The Host parameter depends on the location of the file.

- If the file is stored on the shared storage, you need to use localhost.
- If the file is stored on the local storage, it can be either localhost or the nodes host name.

Do not use the SharedAddress here.

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

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

7. Create the resource group for the scalable service.

    # scrgadm -a -g Apache-Tomcat-Scalable-resource-group \
    > -y Maximum_primaries=2 \ 
    > -y Desired_primaries=2 \ 
    > -y RG_dependencies=Apache-Tomcat-SharedAddress-resource-group

If you need more nodes, adjust Maximum_primaries and Desired_primaries to the appropriate value.
8. Create a resource for the Apache Tomcat Disk Storage if it is not in the root file system.
   Otherwise, proceed to Step 10.

   ```
   # scrgadm -a -j Apache-Tomcat-has-resource
   > -g Apache-Tomcat-failover-resource-group
   > -t SUNW.HAStoragePlus
   > -x FilesystemMountPoints=Apache Tomcat- instance-mount-points
   ```

9. Enable the Apache-Tomcat-has-resource.

   ```
   # scstat
   # scswitch -e -j Apache- Tomcat-has-resource
   ```

10. Create and register each required Apache Tomcat component.
    ```
        # cd /opt/SUNWsctomcat/util
        # cp sctomcat_config desired place
        # cp sctomcat_register desired place
    
    Edit the sctomcat_config file and follow the comments within that file, for example:

    ```
    # These parameters can be customized in (key=value) form
    #
    # Copyright 2003 Sun Microsystems, Inc. All rights reserved.
    # Use is subject to license terms.
    #
    # 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 or false for a scalable or failover resource
    # PFILE - absolute pathname 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)
    #

    RS=
    RG=
    PORT=8080
    LH=
    SCALABLE=false
    PFILE=
    HAS_RS=
    ```

    The following is an example for Apache Tomcat.

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

Configure the parameter LH need to be configured as the name of the shared address. 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.

11. After editing sctomcat_config register the resource.

    # ./sctomcat_register

Repeat Steps 8, 9, an 10 for every Apache Tomcat instance you need.

12. Enable each Apache Tomcat resource.

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

    # scstat

    # scswitch -e -j Apache-Tomcat-resource

Verifying the Sun Cluster HA for Apache Tomcat Installation and Configuration

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

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

1. Become superuser on one of the nodes in the cluster which will host Apache Tomcat.

2. Ensure all the Apache Tomcat resources are online with scstat.

    # scstat

    For each Apache Tomcat resource which is not online, use the scswitch command as follows.

    # scswitch -e -j Apache Tomcat-resource

3. If you deployed a failover data service run the scswitch command to switch the Apache Tomcat resource group to another cluster node, such as node2 with the command described below.
Understanding the Sun Cluster Apache Tomcat HA Parameter File

Use the information to understand the contents of the Sun Cluster Apache Tomcat HA parameter file. This section describes the structure and the content of the Sun 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 Sun Cluster HA for Apache Tomcat” on page 21.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>EnvScript</td>
<td>This is a ksh script or a csh script depending on the login-shell of the user name, which 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.</td>
</tr>
<tr>
<td>User</td>
<td>The owner of the Apache Tomcat instance.</td>
</tr>
<tr>
<td>Basepath</td>
<td>Basepath is the absolute pathname to the directory where the Tomcat bin directory resides. Typically, it is the TOMCAT_HOME (3.x) or the CATALINA_HOME (4.x).</td>
</tr>
<tr>
<td>Host</td>
<td>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.</td>
</tr>
<tr>
<td>Port</td>
<td>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.</td>
</tr>
<tr>
<td>Variable</td>
<td>Explanation</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TestCmd</td>
<td>This variable represents the command which is passed to the Apache Tomcat server process to test its sanity.</td>
</tr>
<tr>
<td>ReturnString</td>
<td>The variable ReturnString represents the string which must be present in the answer to the TestCmd. It can not be Connection refused because this string is in the answer when the Apache Tomcat server process is not running.</td>
</tr>
<tr>
<td>Startwait</td>
<td>This variable stands for 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.</td>
</tr>
</tbody>
</table>

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 a Apache Tomcat resource.

- EnvScript
- User
- Basepath
- Startwait

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 Chose the TestCmd and the ReturnString Variable

Alternatives are available to chose the ReturnString:

- Take the start page of your application and set the TestCmd to get /start_page. Set the ReturnString to a string contained in the startpage. With this strategy, you are monitoring that Apache Tomcat is serving your application.
- Take the Apache Tomcat default startpage and set the TestCmd to get /default-startpage. In this case, set the ReturnString to a string contained in the startpage. This string depends on the deployed Apache Tomcat version, for 3.x it is Tomcat home and for 4.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 Sun Cluster Apache Tomcat agent) to Apache Tomcat. Set the TestCmd to get /startpage-of-the-application. In this case, set the ReturnString to a string contained in the startpage. With this strategy, you are monitoring that Apache Tomcat is serving your test application.

- If none of 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 Sun Cluster HA for Apache Tomcat Fault Monitor

This section describes the Sun Cluster HA for Apache Tomcat fault monitor’s probing algorithm and functionality, states the conditions, messages, and recovery actions associated with unsuccessful probing, and states the conditions and messages associated with unsuccessful probing.

For conceptual information on fault monitors, see the Sun Cluster Concepts Guide.

Resource Properties

The Sun 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 used.

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.
3. Connects to the Apache Tomcat via \textit{Host} and \textit{Port}. If the connection is successful it sends the \textit{TestCmd} and tests whether the \textit{ReturnString} comes back. If it fails, it is rescheduled after 5 seconds. If this fails again, then the probe will restart the Apache Tomcat.

\begin{center}
\textbf{Caution} – The \textit{ReturnString} cannot be \textbf{Connection refused} because this string will be returned if no connection is possible.
\end{center}

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 \textit{Retry\_count} within the \textit{Retry\_interval}, then a failover is initiated for the resource group onto another node. This is done if the resource property \textit{Failover\_enabled} is set to \textbf{TRUE}.

\section*{How to debug Sun Cluster HA for Apache Tomcat}

Sun Cluster HA for Apache Tomcat can be used by multiple Apache Tomcat instances. However, it is possible to turn debug on for all Apache Tomcat instances or a particular Apache Tomcat instance.

The Apache Tomcat component has a \texttt{DEBUG} file under \texttt{/opt/SUNWsctomcat/etc}.

This file allows you to switch debug on for all Apache Tomcat instances, or for a specific Apache Tomcat instance on a particular node in a Sun Cluster. If you require debug to be switched on for Sun Cluster HA for Apache Tomcat across the whole Sun Cluster, you will need to repeat this step on all nodes within Sun Cluster.

\section*{How to turn debug on for Sun Cluster HA for Apache Tomcat}

Perform this step for the Apache Tomcat that requires debug output, on each node of Sun Cluster as required.

1. \texttt{Edit /etc/syslog.conf and change daemon.notice to daemon.debug.}

\begin{verbatim}
# grep daemon /etc/syslog.conf
*.err;kern.debug;daemon.notice;mail.crit /var/adm/messages
*.alert;kern.err;daemon.err operator
\end{verbatim}
Change the `daemon.notice` to `daemon.debug` and restart `syslogd`. Note that the output below, from the command `grep daemon /etc/syslog.conf`, shows that `daemon.debug` has now been set.

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

2. Edit `/opt/SUNWsctomcat/etc/config` and change `DEBUG=` to `DEBUG=ALL` or `DEBUG=resource`

```
# cat /opt/SUNWsctomcat/etc/config
#
# Copyright 2003 Sun Microsystems, Inc. All rights reserved.
# Use is subject to license terms.
#
# Usage:
# DEBUG=<RESOURCE_NAME> or ALL
# FDEBUG=ALL
#
```

**Note** – To turn off debug, reverse the steps above.
# Index

<table>
<thead>
<tr>
<th>C</th>
<th>C locale, 19</th>
</tr>
</thead>
<tbody>
<tr>
<td>commands</td>
<td>groupadd, 14</td>
</tr>
<tr>
<td></td>
<td>scinstall, 20</td>
</tr>
<tr>
<td></td>
<td>scrgadm, 21, 24, 25</td>
</tr>
<tr>
<td></td>
<td>scstat, 23, 27</td>
</tr>
<tr>
<td></td>
<td>scswitch, 21, 23, 24, 27</td>
</tr>
<tr>
<td></td>
<td>shutdown.sh, 17</td>
</tr>
<tr>
<td></td>
<td>startup.sh, 17</td>
</tr>
<tr>
<td></td>
<td>su, 14</td>
</tr>
<tr>
<td></td>
<td>useradd, 14</td>
</tr>
<tr>
<td>components</td>
<td>failover configuration, 11</td>
</tr>
<tr>
<td></td>
<td>multiple masters configuration, 12</td>
</tr>
<tr>
<td>configurations</td>
<td>scalable configuration, 12</td>
</tr>
<tr>
<td></td>
<td>failover configuration, 11</td>
</tr>
<tr>
<td></td>
<td>multiple master configuration, 9</td>
</tr>
<tr>
<td></td>
<td>multiple masters configuration, 8</td>
</tr>
<tr>
<td></td>
<td>scalable configuration, 8, 9</td>
</tr>
</tbody>
</table>

| E | enable resource group, 21, 24 |
| | enable Tomcat resource, 23, 26, 27 |
| | environment scripts, 15 |

| F | fault monitor, 30 |
| | files, installation logs, 19 |

| I | install, 14, 20 |
| | installing |
| | Sun Cluster HA for Apache Tomcat by using Web Start program, 19 |
| | log files created, 19 |

| L | load_balancing_policy, 10 |
| | locales, 19 |
| | log files, installation, 19 |

| P | parameter file, 22, 24, 28 |
| | pfile, See parameter file |
| | probing, 30 |
protection of components, 8

R
register, 21, 23, 24, 26
registration files, 12
registration scripts, 12, 23, 26
requirements, 10
restrictions, 10

S
scalability, 8
strategy, 29

V
/var/sadm/install/logs directory, 19
verify, 17, 27

W
Web Start program, 19