

Oracle® Solaris Cluster Data Service for Apache Guide

Copyright © 2009, 2010, 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 software 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 which 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.

AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. 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. UNIX is a registered trademark licensed through X/Open Company, Ltd.

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.

Contents

Preface	5
1 Installing and Configuring Oracle Solaris Cluster HA for Apache	11
Planning the Installation and Configuration	11
Overview of the Installation and Configuration Process for Oracle Solaris Cluster HA for Apache	16
Installing and Configuring Apache	17
Installing a Non-Secure Apache Webserver	17
Installing a Secure Apache Webserver	21
Installing the HA for Apache Packages	25
▼ How to Install the HA for Apache Packages	25
Registering and Configuring Oracle Solaris Cluster HA for Apache	27
Setting Oracle Solaris Cluster HA for Apache Extension Properties	27
Monitoring Arbitrary URIs	28
Tools for Registering and Configuring HA for Apache	29
▼ How to Register and Configure the Oracle Solaris Cluster HA for Apache by Using <code>clsetup</code>	29
▼ How to Register and Configure Oracle Solaris Cluster HA for Apache by Using Oracle Solaris Cluster Command Line Interface (CLI)	32
How to Configure <code>SUNW.HASStoragePlus</code> Resource Type	38
How to Verify Data Service Installation and Configuration	38
Tuning the Oracle Solaris Cluster HA for Apache Fault Monitor	39
Operations by the Oracle Solaris Cluster HA for Apache Fault Monitor	39
Actions in Response to Faults	41
Upgrading the <code>SUNW.apache</code> Resource Type	41
Information for Registering the New Resource Type Version	41
Information for Migrating Existing Instances of the Resource Type	42

A HA for Apache Extension Properties	43
Index	45

Preface

Oracle Solaris Cluster Data Service for Apache Guide explains how to install and configure HA for Apache.

Note – This Oracle Solaris Cluster release supports systems that use the SPARC and x86 families of processor architectures: UltraSPARC, SPARC64, AMD64, and Intel 64. In this document, x86 refers to the larger family of 64-bit 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.

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	Meaning	Example
AaBbCc123	The names of commands, files, and directories, and onscreen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. <code>machine_name% you have mail.</code>
AaBbCc123	What you type, contrasted with onscreen computer output	<code>machine_name% su</code> Password:
<i>aabbcc123</i>	Placeholder: replace with a real name or value	The command to remove a file is <code>rm filename</code> .
<i>AaBbCc123</i>	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the <i>User's Guide</i> . <i>A cache</i> is a copy that is stored locally. Do <i>not</i> 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://docs.sun.com>.

Topic	Documentation
Data service administration	<i>Oracle Solaris Cluster Data Services Planning and Administration Guide</i> Individual data service guides
Concepts	<i>Oracle Solaris Cluster Concepts Guide</i>
Overview	<i>Oracle Solaris Cluster Overview</i>
Software installation	<i>Oracle Solaris Cluster Software Installation Guide</i>
System administration	<i>Oracle Solaris Cluster System Administration Guide</i>
Hardware administration	<i>Oracle Solaris Cluster 3.3 Hardware Administration Manual</i> Individual hardware administration guides
Data service development	<i>Oracle Solaris Cluster Data Services Developer's Guide</i>
Error messages	<i>Oracle Solaris Cluster Error Messages Guide</i>
Command and function reference	<i>Oracle Solaris Cluster Reference Manual</i>

For a complete list of Oracle Solaris Cluster documentation, see the release notes for your release of Oracle Solaris Cluster at <http://docs.sun.com>.

Related Third-Party Web Site References

Third-party URLs that are referenced in this document provide additional related information.

Note – Oracle is not responsible for the availability of third-party web sites mentioned in this document. Oracle does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Oracle will not be responsible or liable for any actual or alleged damage or loss caused or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Documentation, Support, and Training

See the following web sites for additional resources:

- [Documentation \(http://docs.sun.com\)](http://docs.sun.com)
- [Support \(http://www.oracle.com/us/support/systems/index.html\)](http://www.oracle.com/us/support/systems/index.html)
- [Training \(http://education.oracle.com\)](http://education.oracle.com) – Click the Sun link in the left navigation bar.

Oracle Welcomes Your Comments

Oracle welcomes your comments and suggestions on the quality and usefulness of its documentation. If you find any errors or have any other suggestions for improvement, go to <http://docs.sun.com> and click Feedback. Indicate the title and part number of the documentation along with the chapter, section, and page number, if available. Please let us know if you want a reply.

[Oracle Technology Network \(http://www.oracle.com/technetwork/index.html\)](http://www.oracle.com/technetwork/index.html) offers a range of resources related to Oracle software:

- Discuss technical problems and solutions on the [Discussion Forums \(http://forums.oracle.com\)](http://forums.oracle.com).
- Get hands-on step-by-step tutorials with [Oracle By Example \(http://www.oracle.com/technology/obe/start/index.html\)](http://www.oracle.com/technology/obe/start/index.html).
- Download [Sample Code \(http://www.oracle.com/technology/sample_code/index.html\)](http://www.oracle.com/technology/sample_code/index.html).

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 Oracle Solaris Operating System (for example, Oracle Solaris 10)
- The release number of Oracle Solaris Cluster (for example, Oracle Solaris Cluster 3.3)

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

Command	Function
<code>prtconf -v</code>	Displays the size of the system memory and reports information about peripheral devices
<code>psrinfo -v</code>	Displays information about processors
<code>showrev -p</code>	Reports which patches are installed
<code>prtdiag -v</code>	Displays system diagnostic information
<code>/usr/cluster/bin/clnode show-rev</code>	Displays Oracle Solaris Cluster release and package version information

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

Installing and Configuring Oracle Solaris Cluster HA for Apache

This chapter describes the steps to install and configure Oracle Solaris Cluster HA for Apache on your Oracle Solaris Cluster servers. You can also use the same steps to install and configure the Oracle Solaris Cluster HA for Apache Proxy Server.

This chapter contains the following sections.

- “Planning the Installation and Configuration” on page 11
- “Overview of the Installation and Configuration Process for Oracle Solaris Cluster HA for Apache” on page 16
- “Installing and Configuring Apache” on page 17
- “Installing the HA for Apache Packages” on page 25
- “Registering and Configuring Oracle Solaris Cluster HA for Apache” on page 27
- “Tuning the Oracle Solaris Cluster HA for Apache Fault Monitor” on page 39
- “Upgrading the `SUNW.apache` Resource Type” on page 41

You can configure Oracle Solaris Cluster HA for Apache as a failover or a scalable data service. See Chapter 1, “Planning for Oracle Solaris Cluster Data Services,” in *Oracle Solaris Cluster Data Services Planning and Administration Guide* and the *Solaris Cluster Concepts Guide for Solaris OS* document for an overview of failover and scalable data services.

Planning the Installation and Configuration

Before you install Oracle Solaris Cluster HA for Apache, update the following information in the Apache configuration file `httpd.conf`.

Note – The location of the `httpd.conf` file varies according to installation. System administrators typically install the `httpd.conf` file on the cluster file system. The default installation places the `httpd.conf` file in the `/usr/local/apache/conf` directory. When installing Apache packages bundled with Solaris, the file is located in the `/etc/apache` directory.

- **The `ServerName` directive that contains the hostname** – For Oracle Solaris Cluster HA for Apache to be highly available, you must set this directive to the name of the network address (logical hostname or shared address) that is used to access the server. You should have set up the logical hostname or shared address when you installed the cluster. See the *Solaris Cluster Concepts Guide for Solaris OS* document for details on network resources.
- **The `BindAddress` directive, which you must set to the logical host or shared address** – You can configure Apache to bind to `INADDR_ANY`. However, each resource must bind to a unique combination of network resource and port number. For example, if you run multiple resources, you can use `INADDR_ANY` provided that the port number for each resource is different.
- **The `ServerType` directive** – This directive must be set to `standalone`, the default.
- **Multiple instances of Apache** – If you have multiple instances of Apache, you must manage each instance with a separate resource. Furthermore, each separate resource must have a unique `Bin_dir` setting. Under the specified `Bin_dir` property that starts the particular instance of Apache, an `apachectl` script must exist.

Note – Different Apache resources can share the same `httpd` binary, that is, the `apachectl` scripts for different resources can specify the path to the same `httpd` binary. However, you must modify each `apachectl` script to use a different configuration file for specific Apache resources. To do so, use the `-f` option of the `httpd` command to specify a specific `httpd.conf` file.

- **The `DocumentRoot` directive that specifies the location of the documentation root directory** – This directive is a pointer to a location on the cluster file system, where the HTML documents are installed.
- **The `ScriptAlias` directive that contains the location on a cluster file system of the `cgi-bin` directory** – This directive is a pointer to a location on the cluster file system, where the `cgi-bin` files are installed.

Note – You must follow certain conventions when you configure URL mappings for the web server. For example, when setting the CGI directory, locate the CGI directory on the cluster file system to preserve availability. For example, you might map your CGI directory to `/global/diskgroup/ServerRoot/cgi-bin`, where *diskgroup* is the disk device group that contains the Apache software. In situations where the CGI programs access “back-end” servers, such as an RDBMS, ensure that the Oracle Solaris Cluster software controls the “back-end” server. If the server is an RDBMS that the Oracle Solaris Cluster software supports, use one of the highly available RDBMS packages. Alternatively, you can use the APIs that the *Oracle Solaris Cluster Data Services Developer's Guide for Solaris OS* documents to put the server under Oracle Solaris Cluster control.

- **The lock file** – If you use a lock file, set the value of the `LockFile` directive in your `httpd.conf` file to a local file.
- **The `PidFile` directive** – Point this directive to a local file, as in the following example.

```
PidFile /usr/local/apache/log/httpd.pid
```

Note – While using Apache 2.0 with HA for Apache, ensure that the directory to which the `PidFile` directive in the configuration file references exists and proper permissions are assigned. Before you install the HA for Apache package, verify that the Apache software is properly installed and configured to run on a cluster.

In Apache 1.x, the `PidFile` directive points to `/var/run/httpd.pid`, which is created by default when Solaris is installed. But, in Apache 2.0, the `PidFile` directive has been modified to point to `/var/run/apache2/httpd.pid`. Since this directory does not exist by default, you are required to create it manually, else the resource will not start.

- **The Port directive setting that the server port or ports access** – The defaults are set in each node's `httpd.conf` file. The `Port_list` resource property must include all of the ports that the `httpd.conf` files specify.

The `Port_list` property assumes that the web server serves all combinations of ports and IP addresses from the network resources as defined in the `Network_resources_used` property.

```
Port_list="80/tcp,443/tcp,8080/tcp"
```

The preceding `Port_list` configuration, for example, probes the following IP-port combinations.

Host	Port	Protocol
<i>node1</i>	80	tcp
<i>node1</i>	443	tcp

<i>node1</i>	8080	tcp
<i>node2</i>	80	tcp
<i>node2</i>	443	tcp
<i>node2</i>	8080	tcp

However, if *node1* serves ports 80 and 443 only and *node2* serves ports 80 and 8080 only, you can configure the `Port_list` property for Apache as follows.

```
Port_list=node1/80/tcp,node1/443/tcp,node2/80/tcp,node2/8080/tcp
```

Consider the following rules.

- You must specify hostnames or IP addresses (not network resource names) for *node1* and *node2*.
- If Apache serves *nodeN/port* for every *nodeN* in the `Network_resources_used` property, you can use a short form to replace the combination of, for example, *node1/port1*, *node2/port2*, and so on. See the following examples.

Example One

```
Port_list="80/tcp,node1/443/tcp,node2/8080/tcp"
Network_resources_used=node1,node2
```

This example probes the following IP-port combinations.

Host	Port	Protocol
<i>node1</i>	80	tcp
<i>node1</i>	443	tcp
<i>node2</i>	80	tcp
<i>node2</i>	8080	tcp

Example Two

```
Port_list="node1/80/tcp,node2/80/tcp"
Network_resources_used=net-1,net-2
#net-1 contains node1.
#net-2 contains node2 and node3.
```

This example probes the following IP-port combinations.

Host	Port	Protocol
<i>node1</i>	80	tcp

<i>node2</i>	80	tcp
--------------	----	-----

- All of the hostnames (IP addresses) that the `Port_list` property specifies must not belong to a network resource that is specified in any other scalable resource's `Network_resources_used` property. Otherwise, as soon as a scalable service detects that another scalable resource already uses an IP address, creation of the Apache resource fails.

Note – The Apache resource can use an already created shared address resource provided they listen to a port or port list that is not being used on that shared address.

- **Apache Proxy Server** – Add the following lines of code to the `httpd.conf` file if you choose to configure the Apache software as a proxy server.

```
# Proxy Server Directives.
<IfModule mod_proxy.c>
  ProxyRequests On
  <Directory proxy:*>
    Order deny,allow
    Deny from all
    Allow from IP_ADDRESS
  </Directory>
  ProxyVia On
</IfModule>
# End of Proxy Server Directives.
```

Note – If you run Oracle Solaris Cluster HA for Apache and another HTTP server, configure the HTTP servers to listen on different ports. Otherwise, a port conflict can occur between the two servers.

To register and configure Oracle Solaris Cluster HA for Apache, you must consider or provide information on the following points.

- Decide whether to run Oracle Solaris Cluster HA for Apache as a failover or scalable data service.
- Decide whether to install a secure or non secure version of the Apache webserver.
- Decide which fault monitoring resource properties to set. In most cases, the default values suffice. See *Solaris Cluster Data Services Planning and Administration Guide for Solaris OS* for information about the standard properties and [Appendix A, “HA for Apache Extension Properties,”](#) for information about the extension properties.
- Provide the name of the resource type for Oracle Solaris Cluster HA for Apache. This name is `SUNW.apache`.
- Provide the names of the cluster nodes that will master the data service.

- Provide the logical hostname (failover services) or shared address (scalable services) that clients use to access the data service. You typically set up this IP address when you install the cluster. See the *Solaris Cluster Concepts Guide for Solaris OS* document for details on network resources.
- Provide the path to the application binaries. You can install the binaries on the local disks or on the cluster file system. See “[Configuration Guidelines for Oracle Solaris Cluster Data Services](#)” in *Oracle Solaris Cluster Data Services Planning and Administration Guide* for a discussion of the advantages and disadvantages of each location.
- Modify each copy of `apachectl` to use the appropriate `httpd.conf` configuration file.
- Exercise caution when you change the `Load_balancing_weights` property for an online scalable service that has the `Load_balancing_policy` property set to `LB_STICKY` or `LB_STICKY_WILD`. Changing these properties while the service is online can cause existing client affinities to be reset, hence a different node might service a subsequent client request even if another cluster member previously serviced the client.

Similarly, when a new instance of the service is started on a cluster, existing client affinities might be reset.
- Determine the entry for the `Port_list` property. The `Port_list` property can have multiple entries. See “[How to Register and Configure Oracle Solaris Cluster HA for Apache by Using Oracle Solaris Cluster Command Line Interface \(CLI\)](#)” on page 32 for details.
- Determine whether to utilize the `Monitor Uri List` extension property. This extension property enables you to monitor an arbitrary list of URIs. Arbitrary monitoring of URIs is beneficial if you require the HA for Apache agent probe to monitor any applications (URIs) deployed on the HA for Apache server. Use of the `Monitor Uri List` extension property is not supported with secure instances of HA for Apache. You must install release 3.1 10/03 or 3.2 of HA for Apache to use this property. If you are upgrading HA for Apache from a previous version, you must perform a resource type upgrade procedure to use the new property. For instructions, see “[Upgrading a Resource Type](#)” in *Oracle Solaris Cluster Data Services Planning and Administration Guide*. See “[Monitoring Arbitrary URIs](#)” on page 28 for detailed information about using the `Monitor Uri List` extension property.

Overview of the Installation and Configuration Process for Oracle Solaris Cluster HA for Apache

The table below lists the sections that describe the installation and configuration tasks.

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

Task	Instructions
Install the Apache software	“ Installing and Configuring Apache ” on page 17

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

Task	Instructions
Install the Oracle Solaris Cluster HA for Apache packages	“How to Install the HA for Apache Packages” on page 25
Configure and start Oracle Solaris Cluster HA for Apache	“How to Register and Configure Oracle Solaris Cluster HA for Apache by Using Oracle Solaris Cluster Command Line Interface (CLI)” on page 32
Tune the Oracle Solaris Cluster HA for Apache fault monitor	“Tuning the Oracle Solaris Cluster HA for Apache Fault Monitor” on page 39

Installing and Configuring Apache

The Apache webserver can be installed and set up as either a non secure or a secure webserver. This section provides procedures for both types of installations. To install a non secure version of the webserver, see one of the following procedures.

- “How to Install and Configure the Apache Software From the Solaris CD-ROM” on page 18
- “How to Install and Configure the Apache Software from the Apache Web Site” on page 19

To install a secure version of the webserver, see one of the following procedures.

- “How to Install and Configure the Apache Software Using `mod_ssl`” on page 21
- “How to Install and Configure the Apache Software Using `apache-ssl`” on page 23

Oracle Solaris Cluster HA for Apache works with the Apache software configured as either a web server or a proxy server.

See Apache documentation at <http://www.apache.org> for standard installation instructions. Contact your Sun sales representative for a complete list of Apache versions that are supported with the Oracle Solaris Cluster software.

Note – The Oracle Solaris Cluster HA for Apache can be configured to run in a whole root or a sparse root non-global zone, if required.

Installing a Non-Secure Apache Webserver

This section provides procedures for installing a non-secure Apache webserver. For procedures for installing a secure Apache webserver, see “Installing a Secure Apache Webserver” on page 21.

▼ How to Install and Configure the Apache Software From the Solaris CD-ROM

This procedure installs a non secure version of the Apache webserver. For procedures for installing a secure Apache webserver, see “[Installing a Secure Apache Webserver](#)” on page 21.

The Apache binaries are included in three packages—SUNWapchr, SUNWapchu, and SUNWapchd—that form the SUNWCapache package metacluster. You must install the SUNWapchr package before you install the SUNWapchu package.

Place the Web server binaries on the local file system on each of your cluster nodes or on a cluster file system.

Note – If you are using the `clsetup` utility to configure Oracle Solaris Cluster HA for Apache, skip Step 2 in this procedure. The `clsetup` utility automates Step 2.

1 Run the `pkginfo(1)` command to determine if the Apache packages SUNWapchr, SUNWapchu, and SUNWapchd have been installed.

If these packages have not been installed, install them as follows.

- Starting with Solaris 9, run the following command.


```
# pkgadd -d Solaris-product-directory SUNWapchr SUNWapchu SUNWapchd
```
- Starting with Solaris 10, run the following command.


```
# pkgadd -G -d Solaris-product-directory SUNWapchr SUNWapchu SUNWapchd
```

The output from the command is as follows.

```
...
Installing Apache Web Server (root) as SUNWapchr
...
[ verifying class initd ]
/etc/rc0.d/K16apache linked-pathname
/etc/rc1.d/K16apache linked-pathname
/etc/rc2.d/K16apache linked-pathname
/etc/rc3.d/S50apache linked-pathname
/etc/rcS.d/K16apache linked-pathname
...
```

2 Disable the START and STOP run control scripts that were just installed as part of the SUNWapchr package.

This step is necessary because Oracle Solaris Cluster HA for Apache starts and stops the Apache application after you have configured the data service. Perform the following steps.

- a. List the Apache run control scripts.
- b. Rename the Apache run control scripts.

c. Verify that all the Apache-related scripts have been renamed.

The following example changes the first letter in the name of the run control script from uppercase to lowercase. However, you can rename the scripts to be consistent with your normal administration practices.

```
# ls -l /etc/rc?.d/*apache
/etc/rc0.d/K16apache
/etc/rc1.d/K16apache
/etc/rc2.d/K16apache
/etc/rc3.d/S50apache
/etc/rcS.d/K16apache

# mv /etc/rc0.d/K16apache /etc/rc0.d/k16apache
# mv /etc/rc1.d/K16apache /etc/rc1.d/k16apache
# mv /etc/rc2.d/K16apache /etc/rc2.d/k16apache
# mv /etc/rc3.d/S50apache /etc/rc3.d/s50apache
# mv /etc/rcS.d/K16apache /etc/rcS.d/k16apache

# ls -l /etc/rc?.d/*apache
/etc/rc0.d/k16apache
/etc/rc1.d/k16apache
/etc/rc2.d/k16apache
/etc/rc3.d/s50apache
/etc/rcS.d/k16apache
```

▼ How to Install and Configure the Apache Software from the Apache Web Site

This procedure installs a non secure version of the Apache webserver. For procedures for installing a secure Apache webserver, see “Installing a Secure Apache Webserver” on page 21.

Place the web server binaries on the local file system on each of your cluster nodes or on a cluster file system.

- 1 On a cluster member, become superuser or assume a role that provides `solaris.cluster.admin` RBAC authorization.**
- 2 Install the Apache software using the installation procedures found in the Apache installation documentation.**

Install the Apache software using the Apache installation documentation you received with your Apache software or see the installation instructions at <http://www.apache.org>.

- 3 Update the `httpd.conf` configuration file.**
 - Set the `ServerName` directive. (In Version 2.0 of Apache, the `ServerName` directive specifies the hostname and the port.)
 - Set the `BindAddress` directive (optional). (The `BindAddress` directive only exists in versions prior to Apache 2.0. For Apache 2.0, see the following bullet for the `Listen` directive.)

- Set the `Listen` directive. The `Listen` directive must use the address of the logical host or shared address. (The `Listen` directive only exists in Apache 2.0 and beyond. For Apache versions prior to Apache 2.0, see the previous bullet for the `BindAddress` directive.)
- Set the `ServerType`, `ServerRoot`, `DocumentRoot`, `ScriptAlias`, and `LockFile` directives.

Note – The `ServerType` directive does not exist in Apache 2.0.

- Set the `Port` directive to the same number as the `Port_list` standard resource property. See [Step 4](#) for more information.
- Add the following lines of code to the `httpd.conf` configuration file if you choose to configure the Apache software as a proxy server.

```
# Proxy Server Directives.
<IfModule mod_proxy.c>
  ProxyRequests On
  <Directory proxy:*>
    Order deny,allow
    Deny from all
    Allow from IP_ADDRESS
  </Directory>
  ProxyVia On
</IfModule>
# End of Proxy Server Directives.
```

Note – If you configure the Apache software as a proxy server, the `CacheRoot` setting must point to a location on the cluster file system.

Note – If you are using the `clsetup` utility to configure Oracle Solaris Cluster HA for Apache, you do not need to update the `BindAddress`, `ServerRoot`, and `Port` directives. These directives are automatically updated when you run the `clsetup` utility.

4 Verify that the port number or numbers in the `httpd.conf` file match those of the `Port_list` standard resource property.

You can edit the `httpd.conf` configuration file to change its port number or numbers to match the standard Oracle Solaris Cluster resource property default (port 80). Alternatively, while you configure Oracle Solaris Cluster HA for Apache, you can set the `Port_list` standard property to match the setting in the `httpd.conf` file.

5 Update the paths in the Apache start/stop script file (`Bin_dir/apachectl`).

You must change the paths from the Apache defaults to match your Apache directory structure. For example, change the line in the `BIN_dir/apachectl` script beginning with `HTTPD=/usr/local/apache/bin/httpd` to the following.

```
HTTPD='/usr/local/apache/bin/httpd -f /global/foo/apache/conf/httpd.conf'
```

- 6 Perform the following tasks to verify your configuration changes.
 - a. Run `apachectl configtest` to check the Apache `httpd.conf` file for correct syntax.

Note – If you are using the `clsetup` utility to configure Oracle Solaris Cluster HA for Apache, skip this step. This step is automatically executed when you run the `clsetup` utility.

- b. Ensure that any logical hostnames or shared addresses that Apache uses are configured and online.
 - c. Issue `apachectl start` to start up your Apache server by hand.
If Apache does not start up correctly, correct the problem.
 - d. After Apache has started, stop it before moving to the next procedure.

Installing a Secure Apache Webserver

This section provides procedures for installing a secure Apache webserver. For procedures for installing a non-secure Apache webserver, see “[Installing a Non-Secure Apache Webserver](#)” on page 17.

▼ How to Install and Configure the Apache Software Using `mod_ssl`

This procedure installs a secure version of the Apache webserver. For procedures for installing a non-secure Apache webserver, see “[Installing a Non-Secure Apache Webserver](#)” on page 17.

- 1 On a cluster member, become superuser or assume a role that provides `solaris.cluster.admin` RBAC authorization.
- 2 Install the Apache software, including `mod_ssl`.
To install `mod_ssl`, see the Apache installation documentation or the installation instructions at <http://www.modssl.org>.
- 3 Update the `httpd.conf` configuration file.
 - Set the `ServerName` directive.
 - Set the `BindAddress` directive (optional).
 - Set the `ServerType`, `ServerRoot`, `DocumentRoot`, `ScriptAlias`, and `LockFile` directives.
 - Set the `Port` directive to the same number as the `Port_list` standard resource property. See [Step 4](#) for more information.
 - Add the following lines of code to the `httpd.conf` configuration file if you choose to configure the Apache software as a proxy server.

```
# Proxy Server Directives.  
<IfModule mod_proxy.c>  
  ProxyRequests On  
  <Directory proxy:*>  
    Order deny,allow  
    Deny from all  
    Allow from IP_ADDRESS  
  </Directory>  
  ProxyVia On  
</IfModule>  
# End of Proxy Server Directives.
```

Note – If you configure the Apache software as a proxy server, the `CacheRoot` setting must point to a location on the cluster file system.

4 Verify that the port number or numbers in the `httpd.conf` file match those of the `Port_list` standard resource property.

You can edit the `httpd.conf` configuration file to change its port number or numbers to match the standard Oracle Solaris Cluster resource property default (port 80). Alternatively, while you configure Oracle Solaris Cluster HA for Apache, you can set the `Port_list` standard property to match the setting in the `httpd.conf` file.

5 Install all certificates and keys.

6 In `Bin_dir` directory, create a file called `keypass`. Make sure that no one other than the owner has any permissions for this file.

```
# cd Bin_dir  
# touch keypass  
# chmod 700 keypass
```

7 If you are using an encrypted private key or keys, perform the following Step a and Step b.

a. In the `httpd.conf` file, look for `SSLPassPhraseDialog` directive and modify it as follows.

```
# SSLPassPhraseDialog exec: /Bin_dir/keypass
```

See the `mod_ssl` documentation for details about the `SSLPassPhraseDialog` directive.

b. Edit the `keypass` file so that it prints the pass phrase for the encrypted key corresponding to a host and a port.

This file will be called with `server:port algorithm` as arguments. Make sure that the file can print the pass phrase for each of your encrypted keys when called with the correct parameters.

Later, when you attempt to start the web server manually, it must not prompt you for a pass phrase. For example, for a secure web server listening on ports 8080 and 8888, with private keys for both encrypted using RSA, the `keypass` file could be the following.

```
# !/bin/ksh  
host='echo $1 | cut -d: -f1'
```

```
port='echo $1 | cut -d: -f2'
algorithm=$2

if [ "$host" = "phys-schost-1.example.com" -a "$algorithm" = "RSA" ]; then
  case "$port" in
    8080) echo passphrase-for-8080;;
    8888) echo passphrase-for-8888;;
  esac
fi
```

Note – The keypass file must not be readable, writable, or executable by anyone other than the owner.

- 8 **In the `httpd.conf` file, set the `SSLLogLevel` to warn to avoid logging a message every time the web server is probed by Oracle Solaris Cluster HA for Apache.**
`SSLLogLevel warn`
- 9 **Update the paths in the Apache start/stop script file (`Bin_dir/apachectl`).**
 You must change the paths from the Apache defaults to match your Apache directory structure.
- 10 **Perform the following tasks to verify your configuration changes.**
 - a. **Run `apachectl configtest` to check the Apache `httpd.conf` file for correct syntax.**
 - b. **Ensure that any logical hostnames or shared addresses that Apache uses are configured and online.**
 - c. **Issue `apachectl start` to start up your Apache server by hand.**
 Make sure that the web server does not ask you for a passphrase.
 If Apache does not start up correctly, correct the problem.
 - d. **After Apache has started, stop it before moving to the next procedure.**

▼ **How to Install and Configure the Apache Software Using `apache-ssl`**

This procedure installs a secure version of the Apache webserver. For procedures for installing a non-secure Apache webserver, see “Installing a Non-Secure Apache Webserver” on page 17.

- 1 **On a cluster member, become superuser or assume a role that provides `solaris.cluster.admin` RBAC authorization.**
- 2 **Install the Apache software, including `apache-ssl`, using the installation procedures found in the Apache installation documentation.**
 To install `apache-ssl`, see the Apache installation documentation or the installation instructions at <http://www.apache-ssl.org>.

3 Update the `httpd.conf` configuration file.

- Set the `ServerName` directive.
- Set the `BindAddress` directive (optional).
- Set the `ServerType`, `ServerRoot`, `DocumentRoot`, `ScriptAlias`, and `LockFile` directives.
- Set the `Port` directive to the same number as the `Port_list` standard resource property. See [Step 4](#) for more information.
- Add the following lines of code to the `httpd.conf` configuration file if you choose to configure the Apache software as a proxy server.

```
# Proxy Server Directives.  
<IfModule mod_proxy.c>  
  ProxyRequests On  
  <Directory proxy:*>  
    Order deny,allow  
    Deny from all  
    Allow from IP_ADDRESS  
  </Directory>  
  ProxyVia On  
</IfModule>  
# End of Proxy Server Directives.
```

Note – If you configure the Apache software as a proxy server, the `CacheRoot` setting must point to a location on the cluster file system.

4 Verify that the port number or numbers in the `httpd.conf` file match those of the `Port_list` standard resource property.

You can edit the `httpd.conf` configuration file to change its port number or numbers to match the standard Oracle Solaris Cluster resource property default (port 80). Alternatively, while you configure Oracle Solaris Cluster HA for Apache, you can set the `Port_list` standard property to match the setting in the `httpd.conf` file.

5 Install all certificates and keys.

6 Make sure that all your private keys are stored unencrypted.

Later, when you attempt to start the web server manually, it must not prompt you for a pass phrase.

7 Update the paths in the Apache start/stop script file (`Bin_dir/httpsdctl`).

You must change the paths from the Apache defaults to match your Apache directory structure.

8 Perform the following tasks to verify your configuration changes.

- a. Run `httpsdctl configtest` to check the Apache `httpd.conf` file for correct syntax.

- b. Ensure that any logical hostnames or shared addresses that Apache uses are configured and online.
- c. Issue `httpsdctl start` to start up your Apache server by hand.
If Apache does not start up correctly, correct the problem.
- d. After Apache has started, stop it before moving to the next procedure.

Next Steps If you did not install the HA for Apache during your initial Oracle Solaris Cluster installation, go to [“Installing the HA for Apache Packages” on page 25](#). Otherwise, go to [“Registering and Configuring Oracle Solaris Cluster HA for Apache” on page 27](#).

Installing the HA for Apache Packages

If you did not install the HA for Apache packages during your initial Oracle Solaris Cluster installation, perform this procedure to install the packages. To install the packages, use the `installer` program.

Note – You need to install the HA for Apache packages in the global cluster and not in the zone cluster.

▼ How to Install the HA for Apache Packages

Perform this procedure on each cluster node where you are installing the HA for Apache packages.

You can run the `installer` 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.

Note – Even if you plan to configure this data service to run in non-global zones, install the packages for this data service in the global zone. The packages are propagated to any existing non-global zones and to any non-global zones that are created after you install the packages.

Before You Begin Ensure that you have the Oracle Solaris Cluster installation media.

If you intend to run the `installer` program with a GUI, ensure that your `DISPLAY` environment variable is set.

- 1 On the cluster node where you are installing the data service packages, become superuser.

2 Load the Oracle Solaris Cluster installation media into the DVD-ROM drive.

If the Volume Management daemon `vold(1M)` is running and configured to manage DVD-ROM devices, the daemon automatically mounts the DVD-ROM on the `/cdrom` directory.

3 Change to the installation wizard directory of the DVD-ROM.

- **If you are installing the data service packages on the SPARC platform, type the following command:**

```
# cd /cdrom/cdrom0/Solaris_sparc
```

- **If you are installing the data service packages on the x86 platform, type the following command:**

```
# cd /cdrom/cdrom0/Solaris_x86
```

4 Start the installation wizard.

```
# ./installer
```

5 When you are prompted, accept the license agreement.

6 From the list of Oracle Solaris Cluster agents under Availability Services, select the data service for Apache.

7 If you require support for languages other than English, select the option to install multilingual packages.

English language support is always installed.

8 When prompted whether to configure the data service now or later, choose Configure Later.

Choose Configure Later to perform the configuration after the installation.

9 Follow the instructions on the screen to install the data service packages on the node.

The installation wizard displays the status of the installation. When the installation is complete, the wizard displays an installation summary and the installation logs.

10 (GUI only) If you do not want to register the product and receive product updates, deselect the Product Registration option.

The Product Registration option is not available with the CLI. If you are running the installation wizard with the CLI, omit this step.

11 Exit the installation wizard.

12 Unload the installation media from the DVD-ROM drive.

a. To ensure that the DVD-ROM is not being used, change to a directory that does *not* reside on the DVD-ROM.

b. Eject the DVD-ROM.

```
# eject cdrom
```

Next Steps See “[Registering and Configuring Oracle Solaris Cluster HA for Apache](#)” on page 27 to register Oracle Solaris Cluster HA for Apache and to configure the cluster for the data service.

Registering and Configuring Oracle Solaris Cluster HA for Apache

This section describes how to register and configure Oracle Solaris Cluster HA for Apache.

You can configure Apache as a failover service or as a scalable service, as follows.

- When you configure Apache as a failover service, you place the Apache application resources and the network resources in a single resource group.
- When you configure Apache as a scalable service, you create a scalable resource group for the Apache application resources and a failover resource group for the network resources.

The scalable resource group depends on the failover resource group. Additional steps are required to configure Apache as a scalable service. The leading text “For scalable services only” in the following procedure identifies these steps. If you are not configuring Apache as a scalable service, skip the steps marked “For scalable services only.”

Setting Oracle Solaris Cluster HA for Apache Extension Properties

The sections that follow contain instructions for registering and configuring Oracle Solaris Cluster HA for Apache resources. For information about the extension properties, see [Appendix A, “HA for Apache Extension Properties.”](#) The Tunable entry indicates when you can update a property.

See [Appendix A, “Standard Properties,”](#) in *Oracle Solaris Cluster Data Services Planning and Administration Guide* for details on all of the Oracle Solaris Cluster properties.

To set an extension property of a resource, include the following option in the `clresource(1CL)` command that creates or modifies the resource:

`-p property=value`

`-p property`

Identifies the extension property that you are setting.

value

Specifies the value to which you are setting the extension property.

You can also use the procedures in [Chapter 2, “Administering Data Service Resources,” in *Oracle Solaris Cluster Data Services Planning and Administration Guide*](#) to configure resources after the resources are created.

Monitoring Arbitrary URIs

Set the `Monitor Uri List` extension property if you want the web server fault monitor to probe an arbitrary list of applications (URIs) served by the web server. This extension property provides extended probing functionality and is useful if you are layering services in addition to your web server. The `Monitor Uri List` extension property is not supported with a secure HA for Apache instance. If you do not set the `Monitor Uri List` extension property, the fault monitor will perform the basic probing. See [“Tuning the Oracle Solaris Cluster HA for Apache Fault Monitor” on page 39](#) for details. The following examples show how to set the `Monitor Uri List` extension property when you add the HA for Apache instance to your configuration.

Example— Setting `Monitor Uri List` for Scalable HA for Apache Instance

(Add an insecure Apache instance with default load balancing.)

```
# clresource create -g resource-group-1 \
-t SUNW.apache -p Bin_dir=/opt/apache/bin -p Network_resources_used=schost-1, ... \
-p Monitor Uri_list=http://schost-1:8000/servlet/monitor \
-p Scalable=True \
-p Port_list=8000/tcp apache-insecure-1
```

Example— Setting `Monitor Uri List` for Failover HA for Apache Instance

(Add an insecure Apache application resource instance.)

```
# clresource create -g resource-group-1 \
-t SUNW.apache -p Bin_dir=/opt/apache/bin -p Network_resources_used=schost-1 \
-p Monitor Uri_list=http://schost-1:80/servlet/monitor \
-p Port_list=80/tcp apache-insecure-1
```

Tools for Registering and Configuring HA for Apache

Oracle Solaris Cluster provides the following tools for registering and configuring HA for Apache:

- **The `clsetup(1CL)` utility.** For more information, see “[How to Register and Configure the Oracle Solaris Cluster HA for Apache by Using `clsetup`](#)” on page 29.
- **Oracle Solaris Cluster Manager.** For more information, see the Oracle Solaris Cluster Manager online help.
- **Oracle Solaris Cluster maintenance commands.** For more information, see “[How to Register and Configure Oracle Solaris Cluster HA for Apache by Using Oracle Solaris Cluster Command Line Interface \(CLI\)](#)” on page 32.

The `clsetup` utility and Oracle Solaris Cluster Manager each provide a wizard for configuring HA for Apache. The wizards reduce the possibility for configuration errors that might result from command syntax errors or omissions. These wizards also ensure that all required resources are created and that all required dependencies between resources are set.

▼ How to Register and Configure the Oracle Solaris Cluster HA for Apache by Using `clsetup`

Perform this procedure during your initial setup of Oracle Solaris Cluster HA for Apache. Perform this procedure from one node only.

Note – The `clsetup` utility can be used to configure Apache versions 1.x and Apache 2.0.

Before You Begin Before you start the Oracle Solaris Cluster HA for Apache wizard, ensure that the following prerequisites are met:

- You have decided whether to run Oracle Solaris Cluster HA for Apache as a failover service or a scalable service.
- Prerequisites for configuring the required type of network resource are met.

The type of network resource that is required depends on your configuration of Oracle Solaris Cluster HA for Apache:

- A failover service requires a logical hostname resource.
- A scalable service requires a shared address resource.
- The Apache software is installed and configured.
- The Apache configuration files are available on the node where you will start the wizard.
- The Oracle Solaris Cluster HA for Apache packages are installed.

1 Become superuser on any cluster node.

2 Start the `clsetup` utility.

```
# clsetup
```

The `clsetup` main menu is displayed.

3 Type the number that corresponds to the option for data services and press Return.

The Data Services menu is displayed.

4 Type the number that corresponds to the option for configuring Oracle Solaris Cluster HA for Apache and press Return.

The `clsetup` utility displays information about Oracle Solaris Cluster HA for Apache.

5 Press Return to continue.

The `clsetup` utility displays a list of configuration modes for Oracle Solaris Cluster HA for Apache.

6 Type the number that corresponds to the configuration mode for Oracle Solaris Cluster HA for Apache and press Return.

The `clsetup` utility displays a list of available nodes.

7 Select the nodes where you require Oracle Solaris Cluster HA for Apache to run.

- **To accept the default selection of all listed nodes in an arbitrary order, type `a` and press Return.**

- **To select a subset of the listed nodes, type a comma-separated or space-separated list of the numbers that correspond to the nodes. Then press Return.**

Ensure that the nodes are listed in the order in which the nodes are to appear in the resource group's node list. The first node in the list is the primary node of this resource group.

- **To select all nodes in a particular order, type a comma-separated or space-separated ordered list of the numbers that correspond to the nodes. Then press Return.**

Ensure that the nodes are listed in the order in which the nodes are to appear in the resource group's node list. The first node in the list is the primary node of this resource group.

8 To confirm your selection of nodes, type `d` and press Return.

The `clsetup` utility displays a screen where you can specify the location of the Oracle Solaris Cluster HA for Apache configuration file.

9 Type the numbers that correspond to the location of the configuration file and press Return.

Note – The configuration file that you select here is used as a template. A new configuration file will be created using this template configuration file.

The `clsetup` utility displays a screen where you can specify the Oracle Solaris Cluster HA for Apache document root directory.

10 Type the numbers that correspond to the location of the document root directory and press Return.

The `clsetup` utility displays a screen where you can specify the Oracle Solaris Cluster HA for Apache mount point.

11 To confirm your selection of the mount point, type `d` and press Return.

The `clsetup` utility displays a screen where you can specify the Oracle Solaris Cluster HA for Apache network resource.

12 To confirm your selection of the network resource, type `d` and press Return.

The `clsetup` utility displays information about the Oracle Solaris Cluster HA for Apache configuration that the utility will create.

13 To confirm your selection of the configuration, type `d` and press Return.

The `clsetup` utility displays information about the Oracle Solaris Cluster objects that the utility will create.

Note – The document root specified by you will be copied to the one that is mentioned in the screen. The configuration file specified by you will be edited and the edited file will be stored in the location mentioned in the screen.

14 To confirm your selection of the Oracle Solaris Cluster objects, type `d` and press Return.

The `clsetup` utility displays information about the Oracle Solaris Cluster configuration that the utility will create.

15 To create the configuration, type `c` and Press Return.

The `clsetup` utility displays a progress message to indicate that the utility is running commands to create the configuration. When configuration is complete, the `clsetup` utility displays the commands that the utility ran to create the configuration.

Note – The `clsetup` utility will rollback the changes if it fails to complete the Apache configuration process.

16 Press Return to continue.

The `clsetup` utility returns you to the list of options for configuring Oracle Solaris Cluster HA for Apache.

17 (Optional) Type `q` and press Return repeatedly until you quit the `clsetup` utility.

If you prefer, you can leave the `clsetup` utility running while you perform other required tasks before using the utility again. If you choose to quit `clsetup`, the utility recognizes your Oracle Solaris Cluster HA for Apache resource group when you restart the utility.

18 Determine if the Oracle Solaris Cluster HA for Apache resource group and its resources are online.

Use the `clresourcegroup(1CL)` utility for this purpose. By default, the `clsetup` utility assigns the name `apache-server-rg` to the Oracle Solaris Cluster HA for Apache resource group.

```
# clresourcegroup status apache-server-rg
```

19 If the Oracle Solaris Cluster HA for Apache resource group and its resources are *not* online, bring them online.

```
# clresourcegroup online apache-server-rg
```

▼ How to Register and Configure Oracle Solaris Cluster HA for Apache by Using Oracle Solaris Cluster Command Line Interface (CLI)

Complete the registration and configuration on any cluster member.

Before You Begin

- Verify that all the network addresses that you use have been added to your name service database.

You should have performed this verification during your initial Oracle Solaris Cluster installation. See the planning chapter in the *Solaris Cluster Software Installation Guide for Solaris OS* for details.

Note – To avoid failures because of name service lookup, verify that all the network addresses are present in the `/etc/inet/hosts` file on all of the cluster nodes. Configure name service mapping in the `/etc/nsswitch.conf` file on the servers to first check the local files before accessing NIS, NIS+, or DNS.

1 On a cluster member, become superuser or assume a role that provides `solaris.cluster.admin` RBAC authorization.**2 Register the `SUNW.apache` resource type for the data service.**

```
# clresourcetype register SUNW.apache
```


3 Create a failover resource group to hold the network and application resources.

This resource group is required for both failover and scalable services. For failover services, the resource group contains both network and failover application resources. For scalable services, the resource group contains network resources only. A dependency is created between this group and the resource group that contains the application resources.

Optionally, you can select the set of nodes on which the data service can run with the `-n` option.

```
# clresourcegroup create [-n node-zone-list] resource-group
```

resource-group

Specifies the name of the failover resource group to add. This name can be your choice but must be unique for the resource groups within the cluster.

`[-n node-zone-list]`

Specifies a comma-separated, ordered list of zones that can master this resource group. The format of each entry in the list is *node*. In this format, *node* specifies the node name and *zone* specifies the name of a non-global Solaris zone. To specify the global zone, or to specify a node without non-global zones, specify only *node*.

This list is optional. If you omit this list, the global zone of each cluster node can master the resource group.

4 Add a network resource, such as logical hostname or shared address, to the failover resource group that you created in [Step 3](#).

```
# clressharedaddress create -g resource-group \  
-h hostname,... [-N netiflist] resource
```

`-h hostname,...`

Specifies a comma-separated list of network resources to add.

resource-group

Specifies the name of the failover resource group that you created in [Step 3](#).

resource

Specifies a resource name. If you do not supply your choice for a resource name, the name of the network resource defaults to the first name that is specified after the `-h` option.

`-N netiflist`

Specifies an optional, comma-separated list that identifies the IP Networking Multipathing groups that are on each node or zone. The format of each entry in the list is *netif@node*. The replaceable items in this format are as follows:

netif Specifies an IPMP group name, such as `sc_ipmp0`, or a public network interface card (NIC). If you specify a public NIC, Oracle Solaris Cluster attempts to create the required IPMP groups.

node Specifies the name or ID of a node and, optionally, the name of a non-global Solaris zone. To specify the global zone, or to specify a node without non-global zones, specify only *node*.

Note – If you require a fully qualified hostname, you must specify the fully qualified name with the `-h` option and you cannot use the fully qualified form in the resource name.

Note – Oracle Solaris Cluster does not currently support using the adapter name for `netif`.

5 For scalable services only – Create a scalable resource group to run on all of the desired cluster nodes.

If you run Oracle Solaris Cluster HA for Apache as a failover data service, proceed to [Step 7](#).

Create a resource group to hold a data service application resource. You must specify the maximum and desired number of primary nodes.

Note – If only a subset of nodes can be primaries for this resource group, you must use the `-n` option to specify the names of these potential primaries when you create the resource group.

You must also specify any dependency between this resource group and the failover resource group that you created in [Step 3](#). This dependency ensures that when failover occurs, if the two resource groups are being brought online on the same node, the Resource Group Manager (RGM) starts up the network resource before any data services that depend on the network resource.

```
# clresourcegroup create [-n node-zone-list] \
-p Maximum primaries=m -p Desired primaries=n \
-p RG_dependencies=resource-group resource-group
```

resource-group

Specifies the name of the scalable service resource group to add.

-p Maximum primaries=*m*

Specifies the maximum number of active primary nodes allowed for this resource group. If you do not assign a value to this property, the default is 1.

-p Desired primaries=*n*

Specifies the desired number of active primary nodes allowed for this resource group. If you do not assign a value to this property, the default is 1.

-p RG_dependencies= *resource-group*

Identifies the resource group that contains the shared address resource on which the resource group being created depends, that is, the name of the failover resource group that you created in [Step 3](#).

-n *node-zone-list*

Specifies a comma-separated, ordered list of zones that can master this resource group. The format of each entry in the list is *node*. In this format, *node* specifies the node name and *zone*

specifies the name of a non-global Solaris zone. To specify the global zone, or to specify a node without non-global zones, specify only *node*.

This list is optional. If you omit this list, the global zone of each cluster node can master the resource group.

6 For scalable services only – Create an application resource in the scalable resource group.

If you run Oracle Solaris Cluster HA for Apache as a failover data service, proceed to [Step 7](#).

```
# clresource create -g resource-group \
-t resource-type -p Bin_dir=bin-directory, ... \
-p Network_resources_used=network-resource, ... \
-p Port_list=port-number/protocol[, ...] \
-p Scalable=True resource
```

resource

Specifies your choice for the name of the resource to add.

-g *resource-group*

Specifies the name of the scalable resource group into which the resources are to be placed.

-t *resource-type*

Specifies the type of the resource to add.

-p *Network_resources_used= network-resource, ...*

Specifies a comma-separated list of network resource names that identify the shared addresses that the data service uses.

-p *Port_list=port-number/protocol, ...*

Specifies a comma-separated list of port numbers and protocol to be used, for example, 80/tcp, 81/tcp.

-p *Scalable=*

Specifies a required parameter for scalable services. This parameter must be set to True.

-p *Bin_dir=bin-directory*

Specifies the location where the Apache binaries—in particular, `apachectl`—are installed. Oracle Solaris Cluster HA for Apache requires this extension property.

The resource is created in the enabled state.

Note – Optionally, you can set additional extension properties that belong to the Apache data service to override their default values. See [Appendix A, “HA for Apache Extension Properties,”](#) for a list of extension properties.

7 For failover services only – Create an application resource in the failover resource group.

Perform this step only if you run Oracle Solaris Cluster HA for Apache as a failover data service. If you run Oracle Solaris Cluster HA for Apache as a scalable data service, you should have performed [Step 5](#) and [Step 6](#) and should now proceed to [Step 9](#).

```
# clresource create -g resource-group \  
-t resource-type -p Bin_dir=bin-directory \  
-p Network_resources_used=network-resource, ... \  
-p Port_list=port-number/protocol[, ...] resource
```

resource

Specifies your choice for the name of the resource to add.

-g *resource-group*

Specifies the name of the resource group into which the resources are to be placed, created in [Step 3](#).

-t *resource-type*

Specifies the type of the resource to add.

-p Network_resources_used= *network-resource*, ...

Specifies a comma-separated list of network resources that identify the shared addresses that the data service uses.

-p Port_list=*port-number/protocol*, ...

Specifies a comma-separated list of port numbers and protocol to be used, for example, 80/tcp, 81/tcp.

-p Scalable=

This property is required for scalable services only. Here the value is set to `False` or can be omitted.

-p Bin_dir=*bin-directory*

Specifies the location where the Apache binaries—in particular, `apachectl`—are installed. Oracle Solaris Cluster HA for Apache requires this extension property.

The resource is created in the enabled state.

8 Bring the failover resource group online.

```
# clresourcegroup online resource-group
```

resource-group

Specifies the name of the failover resource group.

9 For scalable services only – Bring the scalable resource group online.

```
# clresourcegroup online resource-group
```

resource-group

Specifies the name of the scalable resource group.

Example 1-1 Registering Scalable Oracle Solaris Cluster HA for Apache

For scalable services, you create the following resource groups.

- A failover resource group that contains the network resources
- A scalable resource group that contains the application resources

The following example shows how to register a scalable Apache service on a two-node cluster.

```

Cluster Information
Node names: phys-schost-1, phys-schost-2
Shared address: schost-1
Resource groups: resource-group-1 (for shared addresses),
                 resource-group-2 (for scalable Apache application
                 resources)
Resources: schost-1 (shared address), apache-1 (Apache application
           resource)

(Add a failover resource group to contain shared addresses.)
# clresourcegroup create resource-group-1

(Add the shared address resource to the failover resource group.)
# clressharedaddress create -g resource-group-1 -h schost-1

(Register the Apache resource type.)
# clresourcetype register SUNW.apache

(Add a scalable resource group.)
# clresourcegroup create -p Maximum primaries=2 \
-p Desired primaries=2 -p RG_dependencies=resource-group-1 resource-group-2

(Add Apache application resources to the scalable resource group.)
# clresource create -g resource-group-2 \
-t SUNW.apache -p Bin_dir=/opt/apache/bin -p Network_resources_used=schost-1 \
-p Scalable=True \
-p Port_list=80/tcp apache-1

(Bring the failover resource group online.)
# clresourcegroup online resource-group-1

(Bring the scalable resource group online on both nodes.)
# clresourcegroup online resource-group-2

```

Example 1-2 Registering Failover Oracle Solaris Cluster HA for Apache

The following example shows how to register a failover Apache service on a two-node cluster.

```

Cluster Information
Node names: phys-schost-1, phys-schost-2
Logical hostname: schost-1
Resource group: resource-group-1 (for all of the resources)
Resources: schost-1 (logical hostname),
           apache-1 (Apache application resource)

(Add a failover resource group to contain all of the resources.)

```

```
# clresourcegroup create resource-group-1

(Add the logical hostname resource to the failover resource group.)
# clreslogicalhostname create -g resource-group-1 -h schost-1

(Register the Apache resource type.)
# clresourcetype register SUNW.apache

(Add Apache application resources to the failover resource group.)
# clresource create -g resource-group-1 \
-t SUNW.apache -p Bin_dir=/opt/apache/bin -p Network_resources_used=schost-1 \
-p Port_list=80/tcp apache-1

(Bring the failover resource group online.)
# clresourcegroup online resource-group-1
```

How to Configure SUNW.HASStoragePlus Resource Type

The SUNW.HASStoragePlus resource type was introduced in Solaris Cluster 3.0 5/02. This new resource type performs the same functions as SUNW.HASStorage, and synchronizes actions between HA storage and the data service.

SUNW.HASStoragePlus also has an additional feature to make a local file system highly available.

See the SUNW.HASStoragePlus(5) man page and “[Relationship Between Resource Groups and Device Groups](#)” in *Oracle Solaris Cluster Data Services Planning and Administration Guide* for background information. See “[Synchronizing the Startups Between Resource Groups and Device Groups](#)” in *Oracle Solaris Cluster Data Services Planning and Administration Guide* for the procedure. (If you are using a Sun Cluster 3.0 version prior to 5/02, you must set up SUNW.HASStorage instead of SUNW.HASStoragePlus. See “[Synchronizing the Startups Between Resource Groups and Device Groups](#)” in *Oracle Solaris Cluster Data Services Planning and Administration Guide* for the procedure.)

How to Verify Data Service Installation and Configuration

After you configure Oracle Solaris Cluster HA for Apache, verify that you can open a web page with the network resources (logical hostnames or shared addresses) and port number from a web browser. Perform a switchover with the `clresourcegroup(1CL)` command to verify that the service continues to run on a secondary node and can be switched back to the original primary.

Tuning the Oracle Solaris Cluster HA for Apache Fault Monitor

The Oracle Solaris Cluster HA for Apache fault monitor is contained in a resource whose resource type is `SUNW.apache`.

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

Tuning the Oracle Solaris Cluster HA for Apache fault monitor involves the following tasks:

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

Information about the Oracle Solaris Cluster HA for Apache fault monitor that you need to perform these tasks is provided in the subsections that follow.

Tune the Oracle Solaris Cluster HA for Apache fault monitor when you register and configure Oracle Solaris Cluster HA for Apache. For more information, see [“Registering and Configuring Oracle Solaris Cluster HA for Apache”](#) on page 27.

For detailed information, see “Tuning Fault Monitors for Solaris Cluster Data Services” in *Solaris Cluster Data Services Planning and Administration Guide for Solaris OS*.

Operations by the Oracle Solaris Cluster HA for Apache Fault Monitor

The Oracle Solaris Cluster HA for Apache probe sends a request to the server to query the health of the Apache server.

Operations by the Fault Monitor Before a Probe

Before querying the Apache server, the probe checks to confirm that network resources are configured for this Apache resource. If no network resources are configured, an error message (No network resources found for resource) is logged, and the probe exits with failure.

Operations for a Nonsecure Web Server

For a *nonsecure* web server, the probe connects to the Apache server and performs an HTTP 1.0 HEAD check by sending the HTTP request and receiving a response. In turn, the probe connects to the Apache server on each IP address/port combination.

The result of this query can be either a failure or a success. If the probe successfully receives a reply from the Apache server, the probe returns to its infinite loop and continues the next cycle of probing and sleeping.

The query can fail for various reasons, such as heavy network traffic, heavy system load, and misconfiguration. Misconfiguration can occur if you did not configure the Apache server to listen on all of the IP address/port combinations that are being probed. The Apache server should service every port for every IP address that is specified for this resource.

The following probe failures are considered as complete failures.

- Failure to connect to the server, as the following error message flags, with %s indicating the hostname and %d the port number.
`Failed to connect to %s port %d %s`
- Running out of time (exceeding the resource property timeout `Probe_timeout`) after trying to connect to the server.
- Failure to successfully send the probe string to the server, as the following error message flags, with the first %s indicating the hostname, %d the port number, and the second %s indicating further details about the error.

`Failed to communicate with server %s port %d: %s`

When the monitor accumulates two partial failures within the resource property interval `Retry_interval`, it counts them as one complete failure.

The following probe failures are considered as partial failures:

- Running out of time (exceeding the resource property timeout `Probe_timeout`) while trying to read the reply from the server to the probe's query.
- Failing to read data from the server for other reasons, as the following error message flags, with the first %s indicating the hostname and %d the port number. The second %s indicates further details about the error.

`Failed to communicate with server %s port %d: %s`

Operations for a Monitored URI List

If you have configured URIs in the `Monitor_Uri_List` extension property, then the probe connects to the Oracle Solaris Cluster HA for Apache server and performs an HTTP 1.1 GET check by sending a HTTP request and receiving a response to each of the URIs in `Monitor_Uri_List`. If the HTTP server return code is 500 (Internal Server Error) or if the connect fails, the probe will take action.

Note – The `Monitor_Uri_List` extension property supports HTTP requests only. It does not support HTTPS requests.

The result of the HTTP requests is either failure or success. If all of the requests successfully receive a reply from the Oracle Solaris Cluster HA for Apache server, the probe returns and continues the next cycle of probing and sleeping.

Heavy network traffic, heavy system load, and misconfiguration can cause the HTTP GET probe to fail. Misconfiguration of the `Monitor_Uri_List` property can cause a failure if a URI in the `Monitor_Uri_List` includes an incorrect port or hostname. For example, if the web server instance is listening on logical host `schost-1` and the URI was specified as `http://schost-2/servlet/monitor`, the probe will try to contact `schost-2` to request `/servlet/monitor`.

Operations for a Secure Web Server

For a *secure* web server, the probe connects to each IP address and port combination. If this connection attempt succeeds, the probe disconnects and returns with a success status. No further checks are performed.

Actions in Response to Faults

Based on the history of failures, a failure can cause either a local restart or a failover of the data service. For detailed information, see “Tuning Fault Monitors for Solaris Cluster Data Services” in *Solaris Cluster Data Services Planning and Administration Guide for Solaris OS*.

Upgrading the SUNW.apache Resource Type

Upgrade the SUNW.apache resource type if the following conditions apply:

- You are upgrading from an earlier version of the HA for Apache data service.
- You need to use the new features of this data service.

For general instructions that explain how to upgrade a resource type, see “[Upgrading a Resource Type](#)” in *Oracle Solaris Cluster Data Services Planning and Administration Guide*. The information that you require to complete the upgrade of the SUNW.apache resource type is provided in the subsections that follow.

Information for Registering the New Resource Type Version

The relationship between a resource type version and the release of Oracle Solaris Cluster data services is shown in the following table. The release of Oracle Solaris Cluster data services indicates the release in which the version of the resource type was introduced.

Resource Type Version	Oracle Solaris Cluster Data Services Release
1	1.0
3.1	3.1 5/03
4	3.1 10/03
4.1	3.2

To determine the version of the resource type that is registered, use the `clresourcetype showcommand`.

The resource type registration (RTR) file for this resource type is `/opt/SUNWscapc/etc/SUNW.apache`.

Information for Migrating Existing Instances of the Resource Type

The information that you require to edit each instance of the SUNW.apache resource type is as follows:

- You can perform the migration at any time.
- If you need to use the new features of the Oracle Solaris Cluster HA for Apache data service, the required value of the `Type_version` property is 4.
- If you need to monitor deployed applications, set the `Monitor Uri List` extension property to a single URI or a list of URIs to specify the locations of the applications that are to be probed.

The following example shows a command for modifying an instance of the SUNW.apache resource type.

EXAMPLE 1-3 Migrating Instances of the SUNW.apache Resource Type

```
# clresource set -p Monitor Uri List=http://sghost-1/test.html \
-p Type_version=4 apache-rs
```

This command modifies the SUNW.apache resource named `apache-rs` as follows:

- The `Type_version` property of this resource is set to 4.
- The Fault Monitor probe will monitor the URI `http://sghost-1/test.html`.

HA for Apache Extension Properties

This section describes the extension properties for the resource type `SUNW.apache`. This resource type represents the Apache application in a Oracle Solaris Cluster configuration.

For details about system-defined properties, see the `r_properties(5)` man page and the `rg_properties(5)` man page.

The extension properties of the `SUNW.apache` resource type are as follows:

`Bin_dir`

The path to the Apache binaries, in particular, `apachectl`. Oracle Solaris Cluster HA for Apache requires this extension property.

Data type	String
Default	No default defined
Range	Not applicable
Tunable	At creation

`Monitor_retry_count`

The number of times that the process monitor facility (PMF) restarts the fault monitor during the time window that the `Monitor_retry_interval` property specifies. This property refers to restarts of the fault monitor itself rather than to the resource. The system-defined properties `Retry_interval` and `Retry_count` control restarting of the resource.

Data type	Integer
Default	4
Range	0 - 2,147,483,641
	-1 indicates an infinite number of retry attempts.
Tunable	At any time

Monitor_retry_interval

The time (in minutes) over which failures of the fault monitor are counted. If the number of times that the fault monitor fails exceeds the value that is specified in the extension property `Monitor_retry_count` within this period, the PMF does not restart the fault monitor.

Data type Integer

Default 2

Range 0 – 2,147,483,641

–1 indicates an infinite retry interval.

Tunable At any time

Monitor_Uri_List

A single URI or a list of URIs that can be used by the fault monitor to probe any deployed applications on the HA for Apache Web Server. Probe deployed applications by setting the property to one or more URIs that are serviced by applications deployed on the HA for Apache Web Server.

Introduced in release 3.1 10/03.

Data type String

Default Null

Range Not applicable

Tunable At any time

Probe_timeout

The timeout value (in seconds) that the fault monitor uses to probe an Apache instance.

Data type Integer

Default 90

Range 0 – 2,147,483,641

Tunable At any time

Index

A

Apache

See also Solaris Cluster HA for Apache

installation

installing non-secure webserver, 17

installing secure webserver, 21

installing software, 17

C

`clnode` command, 9

commands, node information, 8

configuring, Solaris Cluster HA for Apache, 32–38

E

extension properties, `SUNW.apache` resource type, 43–44

F

fault monitor, Oracle Solaris Cluster HA for Apache, 39

files, RTR, 42

G

global zone, 25

H

HA for Apache

installing, 25–27

resource type versions, 41

software packages, installing, 25–27

help, 8–9

`httpd.conf` file, configuring, 11

I

installing

Apache, 17

HA for Apache, 25–27

L

local zones, *See* non-global zones

M

messages file, 9

`Monitor Uri List`, 16, 42

N

non-global zones, 25

non-secure webserver, installing Apache webserver, 17

O

Oracle Solaris Cluster HA for Apache, fault monitor, 39

P

packages, 25–27

properties

See also extension properties

 Monitor_Uri_List, 42

 Type_version, 42

prtcnf -v command, 9

prtdiag -v command, 9

psrinfo -v command, 9

R

registering, Solaris Cluster HA for Apache, 32–38

resource type registration (RTR) file, 42

restrictions, zones, 25

RTR (resource type registration) file, 42

S

secure webserver, installing Apache webserver, 21

show-rev subcommand, 9

showrev -p command, 9

software packages, 25–27

Solaris Cluster HA for Apache

See also Apache

 configuration, planning, 11

 DocumentRoot directive, 11

 httpd.conf file, 11

 installation

 verifying, 38

 lock file, 11

 multiple instances, 11

 Port directive, 11

 registering and configuring, 32–38

 ServerName directive, 11

 ServerType directive, 11

 SUNW.HAStoragePlus resource type, 38

Solaris Cluster HA for Apache (*Continued*)

 task map, 16

Sun Cluster HA for Apache

 BindAddress directive, 11

 installation

 planning, 11

 ScriptAlias directive, 11

SUNW.apache resource type, extension

 properties, 43–44

SUNW.HAStoragePlus resource type, Solaris Cluster HA for Apache, 38

system properties, effect on fault monitors, 39

T

task map, Sun Cluster HA for Apache, 16

technical support, 8–9

Type_version property, 42

V

/var/adm/messages file, 9

verifying, Solaris Cluster HA for Apache

 installation, 38

versions, resource types, 41

Z

zones, 25