Oracle® Solaris Cluster Data Service for Apache Guide

ORACLE®

Oracle Solaris Cluster Data Service for Apache Guide

Part No: E57644

Copyright © 2012, 2015, Oracle and/or its affiliates. All rights reserved.

Access to Oracle Support

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

Contents

Tables

Examples

Using This Documentation

- **Overview** – Describes how to install and configure the Oracle Solaris Cluster HA for Apache data service.
- **Audience** – Experienced system administrators with extensive knowledge of Oracle software and hardware.
- **Required knowledge** – Knowledge of the Oracle Solaris operating system, of Oracle Solaris Cluster software, and expertise with the volume manager software that is used with Oracle Solaris Cluster software.

This document is not to be used as a planning or presales guide.

Product Documentation Library

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

Feedback

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

1

♦ ♦ ♦ **C H A P T E R   1**

Installing and Configuring HA for Apache

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

Install and configure this data service to run in the global zone or a zone cluster. For updated
information about supported configurations of this data service, see the Oracle Solaris Cluster
4 Compatibility Guide (`http://www.oracle.com/technetwork/server-storage/solaris-
cluster/overview/solariscluster4-compatibilityguide-1429037.pdf`).

This chapter contains the following sections:

- "Planning the Installation and Configuration" on page 13
- "Overview of the Installation and Configuration Process for HA for
  Apache" on page 19
- "Installing and Configuring Apache" on page 19
- "Installing the HA for Apache Package" on page 21
- "Registering and Configuring HA for Apache" on page 22
- "Tuning the HA for Apache Fault Monitor" on page 37
- "Upgrading the `SUNW.apache` Resource Type" on page 40

You can configure 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 4.3 Data
Services Planning and Administration Guide* and the *Oracle Solaris Cluster 4.3 Concepts
Guide* document for an overview of failover and scalable data services.

Planning the Installation and Configuration

Before you install 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. When installing the Apache package bundled with Oracle Solaris, the file is located in the `/etc/apache2/2.2` directory.

- **The `ServerName` directive that contains the hostname –** For 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 *Oracle Solaris Cluster 4.3 Concepts Guide* 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* explains 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 /local/apache/log/httpd.pid
```

Note - While using Apache 2.2 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 2.2, the `PidFile` directive can be modified to point to `/local/apache/log/httpd.pid`. Since this directory does not exist by default, you are required to create it manually or 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 `resource_dependencies` 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
phys-schost-1	80	tcp

phys-schost-1	443	tcp
phys-schost-1	8080	tcp
phys-schost-2	80	tcp
phys-schost-2	443	tcp
phys-schost-2	8080	tcp

However, if *phys-schost-1* serves ports `80` and `443` only and *phys-schost-2* serves ports `80` and `8080` only, you can configure the `Port_list` property for Apache as follows:

```
Port_list=phys-schost-1/80/tcp,phys-schost-1/443/tcp,
phys-schost-2/80/tcp,phys-schost-2/8080/tcp
```

Consider the following rules.

- You must specify hostnames or IP addresses (not network resource names) for *phys-schost-1* and *phys-schost-2*.

- If Apache serves *phys-schost-N/port* for every *phys-schost-N* in the `resource_dependencies` property, you can use a short form to replace the combination of, for example, *phys-schost-1/port1*, *phys-schost-2/port2*, and so on. See the following examples.

  Example One

  ```
  Port_list="80/tcp,phys-schost-1/443/tcp,phys-schost-2/8080/tcp"
  resource_dependencies=phys-schost-1,phys-schost-2
  ```

  This example probes the following IP-port combinations.

Host	Port	Protocol
phys-schost-1	80	tcp
phys-schost-1	443	tcp
phys-schost-2	80	tcp
phys-schost-2	8080	tcp

  Example Two

  ```
  Port_list="phys-schost-1/80/tcp,phys-schost-2/80/tcp"
  resource_dependencies=net-1,net-2
  #net-1 contains phys-schost-1.
  #net-2 contains phys-schost-2 and phys-schost-3.
  ```

This example probes the following IP-port combinations.

Host	Port	Protocol
phys-schost-1	80	tcp
phys-schost-2	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 `Resource_dependencies` 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.
```

You can add the following lines to the `httpd.conf` to configure Apache HTTPS. This is a simple configuration for illustration purposes. Refer to Apache documentation for all details.

```
...
LoadModule ssl_module modules/mod_ssl.so

Listen 443
<VirtualHost logical-host:443>
ServerName logical-host.example.com
SSLEngine on
```

```
SSLCertificateFile /global/apache/conf/ssl/server.crt
SSLCertificateKeyFile /global/apache/conf/ssl/server.key
</VirtualHost>
...
```

Note - If you run 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 HA for Apache, you must consider or provide information on the following points.

- Decide whether to run HA for Apache as a failover or scalable data service.
- Decide which fault monitoring resource properties to set. In most cases, the default values suffice. See the *Oracle Solaris Cluster 4.3 Data Services Planning and Administration Guide* 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 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 *Oracle Solaris Cluster 4.3 Concepts Guide* 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 4.3 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 HA for Apache Using Oracle Solaris Cluster Command Line Interface (CLI)" on page 28 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 HTTP instances of HA for Apache. See "Monitoring Arbitrary URIs" on page 23 for detailed information about using the `Monitor_Uri_List` extension property.

Overview of the Installation and Configuration Process for HA for Apache

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

TABLE 1   Task Map: Installing and Configuring HA for Apache

Task	Instructions
Install the Apache software	"Installing and Configuring Apache" on page 19
Install the HA for Apache package	"How to Install the HA for Apache Package" on page 21
Configure and start HA for Apache	"How to Register and Configure HA for Apache Using Oracle Solaris Cluster Command Line Interface (CLI)" on page 28
Tune the HA for Apache fault monitor	"Tuning the HA for Apache Fault Monitor" on page 37

Installing and Configuring Apache

The Apache web server can be installed and set up as either a HTTPS or a HTTP web server. Configuration of SSL or TLS is entirely done within the Apache configuration itself. When using HTTP apache, you must add the HTTP ports to the `Port_lists` property of the `SUNW. apache` resource when it is created. This section provides procedures for both types of installations. To install the Apache web server, see one of the following procedures.

- "How to Install the Apache Software From the Oracle Solaris Software Repository" on page 20
- "How to Install the Apache Software from the Apache Web Site" on page 20

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 Oracle sales representative for a complete list of Apache versions that are supported with the Oracle Solaris Cluster software.

Note - The HA for Apache software can be configured to run in a zone cluster.

Installing an Apache Web Server

This section provides procedures for installing a Apache web server.

▼ How to Install the Apache Software From the Oracle Solaris Software Repository

This procedure installs the Apache web server.

The Apache binaries are included in the `web/server/apache-22` package.

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

● **Run the `pkg info` command to determine if the Apache software has been installed.**

```
# pkg info web/server/apache-22
```

If the package has not been installed, install it as follows.

```
# pkg install web/server/apache-22
```

Use the `pkg info` command to confirm the installation.

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

This procedure installs the Apache web server.

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 an administrator that provides `solaris.cluster.admin` 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.

Installing the HA for Apache Package

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

▼ How to Install the HA for Apache Package

Perform this procedure on each cluster node where you want the HA for Apache software to run.

1. **On the cluster node where you are installing the data service package, assume the `root` role.**

2. **Ensure that the data service package is available from the configured publisher and that the `solaris` and `ha-cluster` publishers are valid.**

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

   For information about setting the `solaris` publisher, see "Adding, Modifying, or Removing Package Publishers" in *Adding and Updating Software in Oracle Solaris 11.3*.

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

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

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

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

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

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

Installation is successful if output shows that State is Installed.

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

   For instructions on updating your software, see Chapter 11, "Updating Your Software," in *Oracle Solaris Cluster 4.3 System Administration Guide* .

Registering and Configuring HA for Apache

This section describes how to register and configure 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."

This section contains the following information:

- "Setting HA for Apache Extension Properties" on page 22
- "Monitoring Arbitrary URIs" on page 23
- "Tools for Registering and Configuring HA for Apache" on page 24
- "How to Register and Configure HA for Apache by Using clsetup" on page 24
- "How to Register and Configure HA for Apache Using Oracle Solaris Cluster Command Line Interface (CLI)" on page 28
- "How to Configure SUNW.HAStoragePlus Resource Type" on page 37
- "How to Verify Data Service Installation and Configuration" on page 37

Setting HA for Apache Extension Properties

The sections that follow contain instructions for registering and configuring 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 the `rt_properties(5)`, `r_properties(5)`, and `rg_properties(5)` man pages for details on all of the Oracle Solaris Cluster extension properties.

To set an extension property of a resource, include the following option in the `clresource` 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 4.3 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 HTTP 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 HA for Apache Fault Monitor" on page 37 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 1   Setting `Monitor_Uri_List` for Scalable HA for Apache Instanc

(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 Resource_dependencies=schost-1, … \
-p Monitor_Uri_list=http://schost-1:8000/servlet/monitor \
-p Scalable=True \
-p Port_list=8000/tcp apache-insecure-1
```

EXAMPLE 2   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 Resource_dependencies=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` **utility.** For more information, see "How to Register and Configure HA for Apache by Using `clsetup`" on page 24 and the `clsetup(1CL)` man page.
- **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 HA for Apache Using Oracle Solaris Cluster Command Line Interface (CLI)" on page 28.

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

▼ How to Register and Configure HA for Apache by Using `clsetup`

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

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

- You have decided whether to run HA for Apache as a failover service or a scalable service.
- Any project you created to run Apache exists in the `projects` name service database for the user that runs that application. If no custom project is created for the application, the default project is used. For more information, see the `projects(1)` man page.
- Prerequisites for configuring the required type of network resource are met.

The type of network resource that is required depends on your configuration of 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 HA for Apache package is installed.

1. **Assume the `root` role 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 HA for Apache and press Return.**

   The `clsetup` utility displays information about HA for Apache.

5. **Press Return to continue.**

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

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

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

7. **Select the nodes where you require 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.

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 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 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 prompts you for the project to use.

11. **Type the option number to choose a listed project and press Return.**
    If you did not create a project where Apache will run, select the default project. The `clsetup` utility displays a screen where you can specify the HA for Apache mount point.

12. **To confirm your selection of the mount point, type d and press Return.**
    The `clsetup` utility displays a screen where you can specify the HA for Apache network resource or create a new one.

13. **To confirm your selection of the network resource, type d and press Return.**

    - **If you created a new network resource, the `clsetup` utility provides a screen where you can specify the PNM objects to use.**
      The `clsetup` utility provides a screen where you can specify the logical hostname that the resource is to make available.

Proceed to Step 14.

- **If you chose an existing network resource, the `clsetup` utility lists the names of the Oracle Solaris Cluster objects that the utility will create.**
  Skip to Step 16.

14. **Type the logical hostname that this resource is to make available and press Return.**

    - **If more than one PNM object is configured for the specified logical hostname, the `clsetup` utility provides a screen where you can specify the PNM objects to use.**
      Proceed to Step 15.

    - **If only one PNM object is configured for the specified logical hostname, the `clsetup` utility lists the names of the Oracle Solaris Cluster objects that the utility will create.**
      Skip to Step 16.

15. **Select from the list of available PNM objects one object for each cluster node.**
    The `clsetup` utility displays information about the HA for Apache configuration that the utility will create.

16. **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.

    ---

17. **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.

18. **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 roll back the changes if it fails to complete the HA for Apache configuration process.

---

19. **Press Return to continue.**

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

20. **(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 HA for Apache resource group when you restart the utility.

21. **Determine if the HA for Apache resource group and its resources are online.**

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

    `# clresourcegroup status apache-server-rg`

22. **If the HA for Apache resource group and its resources are *not* online, bring them online.**

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

▼ How to Register and Configure HA for Apache 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 *Oracle Solaris Cluster 4.3 Software Installation Guide* 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 on the servers to first check the local files before accessing NIS, NIS +, or DNS:

```
/usr/sbin/svccfg -s svc:/system/name-service/switch setprop config/host =
astring: \" cluster files nis\"
```

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

1. **Update the httpd.conf configuration file.**

    - Set the ServerName directive. (In Version 2.2 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 2.2. For Apache 22, 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 2.2 and beyond. For Apache versions prior to Apache 2.2, 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 2.2.

    - Set the Port directive to the same number as the Port_list standard resource property.
    - 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.

---

2. **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.

3. **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*'

4. **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 manually start your Apache server.**

      If Apache does not start up correctly, correct the problem.

   d. **After Apache has started, stop it before moving to the next procedure.**

5. **On a cluster member, become an administrator that provides `solaris.cluster. admin` authorization.**

6. **Register the `SUNW.apache` resource type for the data service.**

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

7. **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-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-list*]

> Specifies a comma-separated, ordered list of nodes that can master this resource group. In this format, *node* specifies the node name in a global zone.

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

8. **Add a network resource, such as logical hostname or shared address, to the failover resource group that you created in Step 7.**

```
# 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 7.

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 IPMP 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. To specify the global zone, 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`.

---

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

   If you run HA for Apache as a failover data service, proceed to .

   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 . 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 7.

-n *node- list*

> Specifies a comma-separated, ordered list of nodes that can master this resource group. The format of each entry in the list is *node*. In this format, *node* specifies the node name . To specify the global zone, specify only *node*.
>
> This list is optional. If you omit this list, the global zone of each cluster node can master the resource group.

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

If you run HA for Apache as a failover data service, proceed to Step 11.

```
# clresource create -g resource-group \
-t resource-type -p Bin_dir=bin-directory,... \
-p Resource_dependencies=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 Resource_dependencies= *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`. Specify HTTPS or HTTP ports being served by the Apache server.

-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. 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 HA for Apache data service to override their default values. See Appendix A, "HA for Apache Extension Properties" for a list of extension properties.

---

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

    Perform this step only if you run HA for Apache as a failover data service. If you run HA for Apache as a scalable data service, you should have performed Step 9 and Step 10 and should now proceed to Step 13.

    ```
    # clresource create -g resource-group \
    -t resource-type -p Bin_dir=bin-directory \
    -p Resource_dependencies=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 you created in Step 7 where the resources are to be placed.

    -t *resource-type*

    > Specifies the type of the resource to add.

    -p Resource_dependencies= *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`). Specify HTTPS or HTTP ports being served by the Apache server.

-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. HA for Apache requires this extension property.

The resource is created in the enabled state.

12. **Bring the failover resource group online.**

    # **clresourcegroup online -M** *resource-group*

    resource-group

    >   Specifies the name of the failover resource group.

13. **For scalable services only – Bring the scalable resource group online.**

    # **clresourcegroup online -M** *resource-group*

    resource-group

    >   Specifies the name of the scalable resource group.

Example 3   Registering Scalable 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.

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

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 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 Resource_dependencies=schost-1 \
-p Scalable=True \
-p Port_list=80/tcp apache-1
```

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

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

Example 4   Registering Failover HA for Apache

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

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

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 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 Resource_dependencies=schost-1 **
-p Port_list=80/tcp apache-1

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

How to Configure `SUNW.HAStoragePlus` Resource Type

See the `SUNW.HAStoragePlus(5)` man page and "Relationship Between Resource Groups and Device Groups" in *Oracle Solaris Cluster 4.3 Data Services Planning and Administration Guide* for background information. See "Synchronizing the Startups Between Resource Groups and Device Groups Using HAStoragePlus" in *Oracle Solaris Cluster 4.3 Data Services Planning and Administration Guide* for the procedure.

How to Verify Data Service Installation and Configuration

After you configure 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` command to verify that the service continues to run on a secondary node and can be switched back to the original primary.

Tuning the HA for Apache Fault Monitor

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

Standard 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 HA for Apache fault monitor *only* if you need to modify this preset behavior.

Tuning the 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 HA for Apache fault monitor that you need to perform these tasks is provided in the subsections that follow.

Tune the HA for Apache fault monitor when you register and configure HA for Apache. For more information, see "Registering and Configuring HA for Apache" on page 22.

For detailed information, see "Tuning Fault Monitors for Oracle Solaris Cluster Data Services" in *Oracle Solaris Cluster 4.3 Data Services Planning and Administration Guide* .

Operations by the HA for Apache Fault Monitor

The 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 Web Server

For a 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 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 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 HTTP Web Server

For a HTTP 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 Oracle Solaris Cluster Data Services" in *Oracle Solaris Cluster 4.3 Data Services Planning and Administration Guide* .

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 4.3 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

Resource Type Version	Oracle Solaris Cluster Data Services Release
3.1	3.1 5/03
4	3.1 10/03
4.1	3.2
4.2	4.0

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

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 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 5   Migrating Instances of the `SUNW.apache` Resource Type

```
# clresource set -p Monitor_Uri_List=http://schost-1/test.html \
-p Type_version=4.2 apache-rs
```

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

- The `Type_version` property of this resource is set to `4.2`.
- The Fault Monitor probe will monitor the URL `http://schost-1/test.html`.

A
♦♦♦ **A P P E N D I X   A**

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 an Oracle Solaris Cluster configuration.

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

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

`Bin_dir`

The path to the Apache binaries, in particular, `apachectl`. 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