Sun Cluster 3.1 Data Service for Apache Guide

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 817–3313
October 2003, Revision A

030730@5943

Contents

Preface

Sun Cluster 3.1 Data Service for Apache Guide explains how to install and configure Sun™ Cluster HA for Apache on your Sun Cluster nodes.

This document is intended for system administrators with extensive knowledge of Sun 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 document assume knowledge of the Solaris™ operating environment and expertise with the volume manager software that is used with Sun Cluster.

UNIX Commands

This document contains information about commands that are specific to installing and configuring Sun 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 Solaris software environment
- Solaris operating environment man pages
- Other software documentation that you received with your system

Typographic Conventions

The following table describes the typographic changes used in this book.

TABLE P–1 Typographic Conventions

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

Shell Prompts in Command Examples

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

TABLE P–2 Shell Prompts

Shell	Prompt
C shell prompt	`machine_name%`
C shell superuser prompt	`machine_name#`
Bourne shell and Korn shell prompt	`$`
Bourne shell and Korn shell superuser prompt	`#`

Related Documentation

Information about related Sun Cluster topics is available in the documentation that is listed in the following table.

Topic	Title	Part Number
Data service administration	*Sun Cluster 3.1 Data Service Planning and Administration Guide*	817-3305
	Sun Cluster 3.1 10/03 Data Services Collection at `http://docs.sun.com/db/coll/573.11`	
Concepts	*Sun Cluster 3.1 10/03 Concepts Guide*	817-0519
Software installation	*Sun Cluster 3.1 10/03 Software Installation Guide*	817-0518
System administration	*Sun Cluster 3.1 10/03 System Administration Guide*	817-0516
Hardware administration	*Sun Cluster 3.1 Hardware Administration Manual*	817-0168
	Sun Cluster 3.x Hardware Administration Collection at `http://docs.sun.com/db/coll/1024.1`	
Data service development	*Sun Cluster 3.1 10/03 Data Services Developer's Guide*	817-0520
Error messages	*Sun Cluster 3.1 10/03 Error Messages Guide*	817-0521
Command and function reference	*Sun Cluster 3.1 10/03 Reference Manual*	817-0522
Release information	*Sun Cluster 3.1 Data Services 10/03 Release Notes*	817-3324
	Sun Cluster 3.1 10/03 Release Notes	817-0638
	Sun Cluster 3.x Release Notes Supplement	816-3381

Accessing Sun Documentation Online

The docs.sun.com^{SM} Web site enables you to access Sun technical documentation online. You can browse the docs.sun.com archive or search for a specific book title or subject. The URL is `http://docs.sun.com`.

Ordering Sun Documentation

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

Help

If you have problems installing or using Sun 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 and serial numbers of your systems
- The release number of the operating environment (for example, Solaris 8)
- The release number of Sun Cluster (for example, Sun Cluster 3.0)

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

Command	Function
`prtconf -v`	Displays the size of the system memory and reports information about peripheral devices
`psrinfo -v`	Displays information about processors
`showrev –p`	Reports which patches are installed
`prtdiag -v`	Displays system diagnostic information
`scinstall -pv`	Displays Sun Cluster release and package version information

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

Installing and Configuring Sun Cluster HA for Apache

This chapter describes the steps to install and configure Sun Cluster HA for Apache on your Sun Cluster servers.

This chapter contains the following procedures.

You can configure Sun Cluster HA for Apache as a failover or a scalable data service. See "Planning for Sun Cluster Data Services" in *Sun Cluster 3.1 Data Service Planning and Administration Guide* and the *Sun Cluster 3.1 Concepts Guide* document for an overview of failover and scalable data services.

Note – You can use SunPlex Manager to install and configure this data service. See the SunPlex Manager online help for details.

Planning the Installation and Configuration

Before you install Sun 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 Sun 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 *Sun Cluster 3.1 Concepts Guide* 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 `apachect1` script must exist.

  Note – Different Apache resources can share the same `httpd` binary, that is, the `apachect1` scripts for different resources can specify the path to the same `httpd` binary. However, you must modify each `apachect1` 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 Sun Cluster software controls the "back-end" server. If the server is an RDBMS that the Sun Cluster software supports, use one of the highly available RDBMS packages. Alternatively, you can use the APIs that the *Sun Cluster 3.1 Data Services Developer's Guide* documents to put the server under Sun 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
```

- **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
node1	`80`	`tcp`
node1	`443`	`tcp`
node1	`8080`	`tcp`
node2	`80`	`tcp`
node2	`443`	`tcp`
node2	`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 *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
node1	80	tcp
node1	443	tcp
node2	80	tcp
node2	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
node1	80	tcp
node2	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 – If you run Sun 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 Sun Cluster HA for Apache, you must consider or provide information on the following points.

- Decide whether to run Sun 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 (such as the `Thorough_probe_interval` or `Probe_timeout` properties) to set. In most cases, the default values suffice. See "Configuring Sun Cluster HA for Apache Extension Properties" on page 32 for information about these properties.
- Provide the name of the resource type for Sun 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 *Sun Cluster 3.1 Concepts Guide* 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 Sun Cluster Data Services" in *Sun Cluster 3.1 Data Service Planning and Administration Guide* for a discussion of the advantages and disadvantages of each location.
- Modify each copy of `apachect1` 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.

Note – If a scalable proxy is serving a scalable Web resource with the `LB_STICKY` policy, you must also set up an `LB_STICKY` policy for the proxy.

- Determine the entry for the `Port_list` property. The `Port_list` property can have multiple entries. See "How to Register and Configure Sun Cluster HA for Apache" on page 25 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 Sun Cluster HA for Apache agent probe to monitor any applications (URIs) deployed on the Sun Cluster HA for

Apache server. Use of the `Monitor_uri_list` extension property is not supported with secure instances of Sun Cluster HA for Apache. You must install Sun Cluster 3.1 10/03 HA for Sun Cluster HA for Apache to use this property. If you are upgrading Sun Cluster HA for Sun Cluster 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 *Sun Cluster 3.1 Data Service Planning and Administration Guide*. See "Configuring Sun Cluster HA for Apache Extension Properties" on page 32 for detailed information about optional extension property settings and example usage of `Monitor_uri_list`.

Installing and Configuring Sun Cluster HA for Apache

Table 1–1 lists the sections that describe the installation and configuration tasks.

TABLE 1–1 Task Map: Installing and Configuring Sun Cluster HA for Apache

Task	For Instructions, Go To
Install the Apache software	"Installing and Configuring Apache" on page 14
Install the Sun Cluster HA for Apache packages	"How to Install Sun Cluster HA for Apache Packages by Using the Web Start Program" on page 22 "How to Install Sun Cluster HA for Apache Packages by Using the `scinstall` Utility" on page 23
Configure and start Sun Cluster HA for Apache	"How to Register and Configure Sun Cluster HA for Apache" on page 25
Configure resource extension properties	"Configuring Sun Cluster HA for Apache Extension Properties" on page 32
View fault monitor information	"Sun Cluster HA for Apache Fault Monitor" on page 35

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 8 CD-ROM and Solaris 9 CD-ROM" on page 15
- "How to Install and Configure the Apache Software from the Apache Web Site" on page 16

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 18
- "How to Install and Configure the Apache Software Using apache-ssl" on page 20

Sun 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 Sun Cluster software.

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 18.

▼ How to Install and Configure the Apache Software from the Solaris 8 CD-ROM and Solaris 9 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 18.

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.

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

   If not, install as follows.

```
# pkgadd -d Solaris 8 Product directory SUNWapchr SUNWapchu SUNWapchd
...
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 Sun 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 of the Apache-related scripts have been renamed.**

Note – 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 -1 /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 -1 /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 18.

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

1. **Become superuser on a cluster member.**

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.
   - Make changes to run as a proxy server if you choose to run the Apache software as a proxy server. See the Apache documentation for more information. If you will run 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 Sun Cluster resource property default (port `80`). Alternatively, while you configure Sun 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.**

   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 15.

▼ 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 15.

1. **Become superuser on a cluster member.**

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.
   - Make changes to run as a proxy server if you choose to run the Apache software as a proxy server. See the Apache documentation for more information. If you will run 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 Sun Cluster resource property default (port `80`). Alternatively, while you configure Sun 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 encrypted private key(s), 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 webserver manually, it must not prompt you for a pass phrase. For example, for a secure webserver 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 [ "$host" = "button-1.eng.sun.com" -a "$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 webserver is probed by Sun 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 webserver 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 15.

1. **Become superuser on a cluster member.**

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.
- Make changes to run as a proxy server if you choose to run the Apache software as a proxy server. See the Apache documentation for more information. If you will run 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 Sun Cluster resource property default (port `80`). Alternatively, while you configure Sun 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 webserver manually, it must not prompt you for a pass phrase.

7. **Update the paths in the Apache start/stop script file (`Bin_dir/httpsdct1`).**

   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.**

Where to Go From Here

If the Apache data service packages have not been installed from the Sun Cluster Agents CD-ROM, go to “Installing Sun Cluster HA for Apache Packages” on page 22. Otherwise, go to “Registering and Configuring Sun Cluster HA for Apache” on page 24.

Installing Sun Cluster HA for Apache Packages

If you did not install the Sun Cluster HA for Apache packages during your initial Sun Cluster installation, perform this procedure to install the packages. Perform this procedure on each cluster node where you are installing the Sun Cluster HA for Apache packages. To complete this procedure, you need the Sun Cluster Agents CD-ROM.

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

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

- The Web Start program
- The `scinstall` utility

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

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

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

1. **On the cluster node where you are installing the Sun Cluster HA for Apache packages, become superuser.**
2. **(Optional) If you intend to run the Web Start program with a GUI, ensure that your `DISPLAY` environment variable is set.**
3. **Load the Sun Cluster Agents CD-ROM into the CD-ROM drive.**

   If the Volume Management daemon `vold`(1M) is running and configured to manage CD-ROM devices, it automatically mounts the CD-ROM on the `/cdrom/scdataservices_3_1_vb` directory.
4. **Change to the Sun Cluster HA for Apache component directory of the CD-ROM.**

The Web Start program for the Sun Cluster HA for Apache data service resides in this directory.

```
# cd /cdrom/scdataservices_3_1_vb/\
components/SunCluster_HA_Apache_3.1
```

5. **Start the Web Start program.**

```
# ./installer
```

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

7. **Follow instructions on the screen to install the Sun Cluster HA for Apache packages on the node.**

   After the installation is finished, the Web Start program provides an installation summary. This summary enables you to view logs that the Web Start program created during the installation. These logs are located in the `/var/sadm/install/logs` directory.

8. **Exit the Web Start program.**

9. **Unload the Sun Cluster Agents CD-ROM from the CD-ROM drive.**

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

   b. **Eject the CD-ROM.**

```
# eject cdrom
```

Where to Go From Here

See "How to Register and Configure Sun Cluster HA for Apache" on page 25 to register Sun Cluster HA for Apache and to configure the cluster for the data service.

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

You need the Sun Cluster Agents CD-ROM to complete this procedure. Perform this procedure on all of the cluster members that can master Sun Cluster HA for Apache.

1. **Load the Sun Cluster Agents CD-ROM into the CD-ROM drive.**

2. **Run the `scinstall` utility with no options.**

   This step starts the `scinstall` utility in interactive mode.

3. **Choose the menu option, Add Support for New Data Service to This Cluster Node.**

   The `scinstall` utility prompts you for additional information.

4. **Provide the path to the Sun Cluster Agents CD-ROM.**

   The utility refers to the CD as the "data services cd."

5. **Specify the data service to install.**

   The `scinstall` utility lists the data service that you selected and asks you to confirm your choice.

6. **Exit the `scinstall` utility.**

7. **Unload the CD from the drive.**

Where to Go From Here

See "How to Register and Configure Sun Cluster HA for Apache" on page 25 to register Sun Cluster HA for Apache and to configure the cluster for the data service.

Registering and Configuring Sun Cluster HA for Apache

This procedure describes how to use the `scrgadm`(1M) command to register and configure Sun 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."

▼ How to Register and Configure Sun Cluster HA for Apache

Note – Run this procedure on any cluster member.

1. **Become superuser on a cluster member.**
2. **Register the resource type for the data service.**

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

`-a`	Adds the data service resource type.
`-t SUNW.apache`	Specifies the predefined resource type name for your data service.

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 `-h` option.

   ```
   # scrgadm -a -g resource-group [-h nodelist]
   ```

`-a`	Adds a new configuration.
`-g` *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.
[`-h` *nodelist*]	An optional comma-separated list of physical node names or IDs that identify potential masters. The order specified here determines the order in which the nodes are considered as primary during failover.

   Note – Use `-h` to specify the order of the node list. If all of the nodes that are in the cluster are potential masters, you do not need to use the `-h` option.

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

You should have performed this verification during your initial Sun Cluster installation. See the planning chapter in the *Sun Cluster 3.1 Software Installation Guide* for details.

Note – To avoid failures because of name service lookup, verify that all of 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 prior to accessing NIS, NIS+, or DNS.

---

5. **Add a network resource (logical hostname or shared address) to the failover resource group that you created in Step 3.**

```
# scrgadm -a {-S | -L} -g resource-group \
-l hostname, ... [-j resource] \
[-X auxnodelist] [-n netiflist]
```

`-S | -L`	The `-S` option specifies shared address resources. The `-L` option specifies logical hostname resources.
`-l` *hostname*, …	Specifies a comma-separated list of network resources to add. You can use the `-j` option to specify a name for the resources. If you do not do so, the network resources have the name of the first entry on the list.
`-g` *resource-group*	Specifies the name of the failover resource group that you created in Step 3.
`-j` *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 `-l` option.
`-X` *auxnodelist*	Specifies a comma-separated list of physical node names or node IDs that identify cluster nodes that can host the shared address but never serve as primary in the case of failover. These nodes are mutually exclusive with the nodes identified in *nodelist* for the resource group, if specified.
`-n` *netiflist*	Specifies an optional, comma-separated list that identifies the IP Networking Multipathing groups that are on each node. Each element in *netiflist* must be in the form of `netif@node`. `netif` can be given as an IP Networking Multipathing group name, such as `sc_ipmp0`. The node can be identified by the node name or node ID, such as `sc_ipmp0@1` or `sc_ipmp@phys-schost-1`.

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

---

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

   If you run Sun Cluster HA for Apache as a failover data service, proceed to Step 8.

   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 `-h` 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.

```
# scrgadm -a -g resource-group \
-y Maximum_primaries=m -y Desired_primaries=n \
-y RG_dependencies=resource-group \
[-h nodelist]
```

`-g` *resource-group*	Specifies the name of the scalable service resource group to add.
`-y 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`.
`-y 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`.
`-y 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.
`-h` *nodelist*	An optional list of nodes that can be primaries for this resource group. You only need to specify this list if some nodes cannot act as primaries for this resource group.

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

   If you run Sun Cluster HA for Apache as a failover data service, proceed to Step 8.

   ```
   # scrgadm -a -j resource -g resource-group \
   -t resource-type -y Network_resources_used=network-resource, ... \
   -y Port_list=port-number/protocol[, ...] -y Scalable=True \
   -x Bin_dir=bin-directory, ...
   ```

   `-j` *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.

   `-y Network_resources_used=` *network-resource*, …
   Specifies a comma-separated list of network resource names that identify the shared addresses that the data service uses.

   `-y Port_list=`*port-number/protocol*, …
   Specifies a comma-separated list of port numbers and protocol to be used, for example, `80/tcp,81/tcp`.

   `-y Scalable=`
   Specifies a required parameter for scalable services. This parameter must be set to `True`.

   `-x Bin_dir=`*bin-directory*
   Specifies the location where the Apache binaries—in particular, `apachect1`—are installed. Sun Cluster HA for Apache requires this extension property.

Note – Optionally, you can set additional extension properties that belong to the Apache data service to override their default values. See Table 1–2 for a list of extension properties.

---

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

   Perform this step only if you run Sun Cluster HA for Apache as a failover data service. If you run Sun Cluster HA for Apache as a scalable data service, you should have performed Step 6 and Step 7 and should now proceed to Step 10.

   ```
   # scrgadm -a -j resource -g resource-group \
   -t resource-type -y Network_resources_used=network-resource, ... \
   -y Port_list=port-number/protocol[, ...] -y Scalable=False \
   -x Bin_dir=bin-directory
   ```

`-j` *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.

`-y Network_resources_used=` *network-resource*, …
Specifies a comma-separated list of network resources that identify the shared addresses that the data service uses.

`-y Port_list=`*port-number/protocol*, …
Specifies a comma-separated list of port numbers and protocol to be used, for example, `80/tcp,81/tcp`.

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

`-x Bin_dir=`*bin-directory*
Specifies the location where the Apache binaries—in particular, `apachect1`—are installed. Sun Cluster HA for Apache requires this extension property.

9. Bring the failover resource group online.

`# scswitch -Z -g` *resource-group*

`-Z`	Enables the shared address resource and fault monitoring, switches the resource group into a `MANAGED` state, and brings the resource group online.
`-g` *resource-group*	Specifies the name of the failover resource group.

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

`# scswitch -Z -g` *resource-group*

`-Z`	Enables the resource and monitor, moves the resource group to the `MANAGED` state, and brings the resource group online.
`-g` *resource-group*	Specifies the name of the scalable resource group.

Example – Registering Scalable Sun 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.)
```
# scrgadm -a -g resource-group-1
```

(Add the shared address resource to the failover resource group.)
```
# scrgadm -a -S -g resource-group-1 -l schost-1
```

(Register the Apache resource type.)
```
# scrgadm -a -t SUNW.apache
```

(Add a scalable resource group.)
```
# scrgadm -a -g resource-group-2 -y Maximum_primaries=2 \
-y Desired_primaries=2 -y RG_dependencies=resource-group-1
```

(Add Apache application resources to the scalable resource group.)
```
# scrgadm -a -j apache-1 -g resource-group-2 \
-t SUNW.apache -y Network_resources_used=schost-1 \
-y Scalable=True -y Port_list=80/tcp \
-x Bin_dir=/opt/apache/bin
```

(Bring the failover resource group online.)
```
# scswitch -Z -g resource-group-1
```

(Bring the scalable resource group online on both nodes.)
```
# scswitch -Z -g resource-group-2
```

Example – Registering Failover Sun 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.)

```
# scrgadm -a -g resource-group-1
```

(Add the logical hostname resource to the failover resource group.)

```
# scrgadm -a -L -g resource-group-1 -l schost-1
```

(Register the Apache resource type.)

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

(Add Apache application resources to the failover resource group.)

```
# scrgadm -a -j apache-1 -g resource-group-1 \
-t SUNW.apache -y Network_resources_used=schost-1 \
-y Scalable=False -y Port_list=80/tcp \
-x Bin_dir=/opt/apache/bin
```

(Bring the failover resource group online.)

```
# scswitch -Z -g resource-group-1
```

Where to Go From Here

Use the information in "How to Verify Data Service Installation and Configuration" on page 32 to verify the installation. See "Configuring Sun Cluster HA for Apache Extension Properties" on page 32 to set or modify resource extension properties.

▼ How to Configure SUNW.HAStoragePlus Resource Type

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

SUNW.HAStoragePlus also has an additional feature to make a local file system highly available. Sun Cluster HA for Apache is scalable, and therefore you should set up the SUNW.HAStoragePlus resource type.

See the SUNW.HAStoragePlus(5) man page and "Synchronizing the Startups Between Resource Groups and Disk Device Groups" in *Sun Cluster 3.1 Data Service Planning and Administration Guide* for background information. See "Synchronizing the Startups Between Resource Groups and Disk Device Groups" in *Sun Cluster 3.1 Data Service 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.HAStorage instead of SUNW.HAStoragePlus. See "Synchronizing the Startups Between Resource Groups and Disk Device Groups" in *Sun Cluster 3.1 Data Service Planning and Administration Guide* for the procedure.)

▼ How to Verify Data Service Installation and Configuration

After you configure Sun 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 `scswitch`(1M) command to verify that the service continues to run on a secondary node and can be switched back to the original primary.

Configuring Sun Cluster HA for Apache Extension Properties

The only required extension property when you create an Apache server resource is the `Bin_dir` property, whose value is the directory that contains the `apachect1` script.

Typically, you use the command-line `scrgadm -x` *parameter=value* to configure the extension properties when you create the Apache server resource. You can also follow the procedures described in "Administering Data Service Resources" in *Sun Cluster 3.1 Data Service Planning and Administration Guide* to configure the properties later.

See "Standard Properties" in *Sun Cluster 3.1 Data Service Planning and Administration Guide* for details on all of the Sun Cluster properties.

You can update some extension properties dynamically. You can update others, however, only when you create the Apache server resource. The following table describes extension properties that you can configure for the Apache server. The Tunable entries indicate when you can update the property.

TABLE 1–2 Sun Cluster HA for Apache Extension Properties

Name/Data Type	Description
`Bin_dir` (string)	The path to the Apache binaries—in particular, `apachect1`. Sun Cluster HA for Apache requires this extension property. **Default:** None **Range:** None **Tunable:** At creation

TABLE 1–2 Sun Cluster HA for Apache Extension Properties *(Continued)*

Name/Data Type	Description
`Monitor_retry_count` (integer)	Controls restarts of the fault monitor and indicates 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 resource restarts. **Default:** `4` **Range:**`0 – 2,147,483,641` –1 indicates an infinite number of retry attempts. **Tunable:** At creation
`Monitor_retry_interval` (integer)	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. **Default:** `2` **Range:**`0 – 2,147,483,641` `–1` indicates an infinite retry interval. **Tunable:** At creation
`Probe_timeout` (integer)	The timeout value (in seconds) that the fault monitor uses to probe an Apache instance. **Default:** `90` **Range:**`0 – 2,147,483,641` **Tunable:** At creation

TABLE 1–2 Sun Cluster HA for Apache Extension Properties *(Continued)*

Name/Data Type	Description
`Monitor_Uri_List`(string)	A single URI or a list of URIs which can be used by the fault monitor to probe any deployed applications on the Sun Cluster 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 Sun Cluster HA for Apache Web Server. Introduced in release: 3.1 10/03. **Default:** Null **Tunable:**Any time

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 Sun Cluster HA for Apache instance. If you do not set the `Monitor_uri_list` extension property, the fault monitor will perform the basic probing. See "Sun Cluster HA for Apache Fault Monitor" on page 35 for details. The following examples show how to set the `Monitor_uri_list` extension property when you add the Sun Cluster HA for Apache instance to your configuration.

Example— Setting `Monitor_uri_list` for Scalable Sun Cluster HA for Apache Instance

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

```
# scrgadm -a -j apache-insecure-1 -g resource-group-1 \
-t SUNW.apache -y Network_resources_used=schost-1, ... \
-y Scalable=True -y Port_list=8000/tcp -x Bin_dir=/opt/apache/bin \
-x Monitor_Uri_list=http://schost-1:8000/servlet/monitor
```

Example— Setting `Monitor_uri_list` for Failover Sun Cluster HA for ApacheInstance

(Add an insecure Apache application resource instance.)

```
# scrgadm -a -j apache-insecure-1 -g resource-group-1 \
-t SUNW.apache -y Network_resources_used=schost-1 \
```

```
-y Scalable=False -y Port_list=80/tcp \
-x Bin_dir=/opt/apache/bin \
-x Monitor_Uri_list=http://schost-1:80/servlet/monitor
```

Sun Cluster HA for Apache Fault Monitor

The Sun Cluster HA for Apache probe sends a request to the server to query the health of the Apache server. Before the probe actually queries 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.

The probe executes the following steps.

1. Uses the timeout value that the resource property `Probe_timeout` sets to limit the time spent trying to successfully probe the Apache server.
2. For a *non-secure* webserver, 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. If the reply to the query is not received within the `Probe_timeout` limit (previously specified in Step 1), the probe considers this scenario a failure on the part of the Apache data service and records the failure in its history. An Apache probe failure can be a complete failure or a partial failure.

   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 such 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
  ```

3. If you have configured URIs in the `Monitor_Uri_List` extension proterty, then the probe connects to the Sun 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.

   The result of the HTTP requests is either failure or success. If all of the requests successfully receive a reply from the Sun 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`.
4. For a *secure* webserver, 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.
5. Based on the history of failures, a failure can cause either a local restart or a failover of the data service. "Sun Cluster Data Service Fault Monitors" in *Sun Cluster 3.1 Data Service Planning and Administration Guide* further describes this action.

Index