Oracle® Solaris Cluster Data Service for Network File System (NFS) Guide

ORACLE®

Contents

Preface

Oracle Solaris Cluster Data Service for Network File System (NFS) Guide explains how to install and configure Oracle Solaris Cluster data services.

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

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

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

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

Using UNIX Commands

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

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

Typographic Conventions

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

TABLE P–1 Typographic Conventions

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

Shell Prompts in Command Examples

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

TABLE P–2 Shell Prompts

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

Related Documentation

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

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

Access to Oracle Support

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

Getting Help

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

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

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

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

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

1

Installing and Configuring HA for NFS

This chapter describes the steps to install and configure Oracle Solaris Cluster HA for Network File System (NFS) on your Oracle Solaris Cluster nodes.

Note – Install and configure this data service to run only in the global zone. At publication of this document, this data service is not supported in non-global zones. For updated information about supported configurations of this data service, contact your Oracle service representative.

This chapter contains the following sections.

- "Overview of the Installation and Configuration Process for HA for NFS" on page 10
- "Planning the HA for NFS Installation and Configuration" on page 10
- "Installing the HA for NFS Package" on page 12
- "Registering and Configuring HA for NFS" on page 12
- "Tuning the HA for NFS Fault Monitor" on page 30
- "Upgrading the SUNW.nfs Resource Type" on page 33

You must configure Oracle Solaris Cluster HA for NFS as a failover data service. See Chapter 1, "Planning for Oracle Solaris Cluster Data Services," in *Oracle Solaris Cluster Data Services Planning and Administration Guide* and the *Oracle Solaris Cluster Concepts Guide* document for general information about data services, resource groups, resources, and other related topics.

The NFS mount points that are placed under the control of the data service must be the same on all of the nodes that can master the disk device group that contains those file systems.

Oracle Solaris Cluster HA for NFS requires that all NFS client mounts be "hard" mounts.

No Oracle Solaris Cluster node may be an NFS client of a file system that is exported by Oracle Solaris Cluster HA for NFS and is being mastered on a node in the same cluster. Such cross-mounting of Oracle Solaris Cluster HA for NFS is prohibited. Use the cluster file system to share files among cluster nodes.

If Oracle Solaris Resource Manager is used to manage system resources allocated to NFS on a cluster, all Oracle Solaris Cluster HA for NFS resources which can fail over to a common cluster node must have the same Oracle Solaris Resource Manager project ID. This project ID is set with the `Resource_project_name` resource property.

Overview of the Installation and Configuration Process for HA for NFS

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

TABLE 1–1   Task Map: Installing and Configuring HA for NFS

Task	For Instructions
Install HA for NFS packages	"How to Install the HA for NFS Package" on page 12
Set up and configure HA for NFS	"Registering and Configuring HA for NFS" on page 12
Tune the HA for NFS fault monitor	"Tuning the HA for NFS Fault Monitor" on page 30
Upgrade the `SUNW.nfs` resource type	"Upgrading the `SUNW.nfs` Resource Type" on page 33

Planning the HA for NFS Installation and Configuration

This section contains the information that you need to plan the installation and configuration of your HA for NFS.

Service Management Facility Restrictions

The following Service Management Facility (SMF) services are related to NFS.

- `/network/nfs/cbd`
- `/network/nfs/mapid`
- `/network/nfs/server`
- `/network/nfs/rquota`
- `/network/nfs/client`
- `/network/nfs/status`
- `/network/nfs/nlockmgr`

The HA for NFS data service sets the property `application/auto_enable` to `FALSE` and the property `startd/duration` to `transient` for three of these services.

- `/network/nfs/server`
- `/network/nfs/status`

- `/network/nfs/nlockmgr`

These property settings have the following consequences for these services.

- When services that depend on these services are enabled, these services are not automatically enabled.
- In the event of any failure, SMF does not restart the daemons that are associated with these services.
- In the event of any failure, SMF does not restart these services.

NFSv3 Restrictions

If you are mounting file systems on the cluster nodes from external NFS servers, such as NAS filers, and you are using the NFSv3 protocol, you cannot run NFS client mounts and the HA for NFS data service on the same cluster node. If you do, certain HA for NFS data-service activities might cause the NFS daemons to stop and restart, interrupting NFS services. However, you can safely run the HA for NFS data service if you use the NFSv4 protocol to mount external NFS file systems on the cluster nodes.

Loopback File System Restrictions

Do *not* use the loopback file system (LOFS) if both conditions in the following list are met:

- HA for NFS is configured on a highly available local file system.
- The `automountd` daemon is running.

If both of these conditions are met, LOFS must be disabled to avoid switchover problems or other failures. If only one of these conditions is met, it is safe to enable LOFS.

If you require both LOFS and the `automountd` daemon to be enabled, exclude from the automounter map all files that are part of the highly available local file system that is exported by HA for NFS.

ZFS Restrictions

If you are using ZFS as the exported file system, you must set the `sharenfs` property to `off`.

To set the `sharenfs` property to `off`, run the following command.

```
$ zfs set sharenfs=off file_system/volume
```

To verify if the `sharenfs` property is set to `off`, run the following command.

```
$ zfs get sharenfs file_system/volume
```

Installing the HA for NFS Package

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

▼ How to Install the HA for NFS Package

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

1 **On the cluster node where you are installing the data service package, become superuser.**

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

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

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

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

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

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

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

Installation is successful if output shows that State is Installed.

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

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

Registering and Configuring HA for NFS

This section describes how to register and configure HA for NFS.

Note – Other options also enable you to register and configure the data service. See "Tools for Data Service Resource Administration" in *Oracle Solaris Cluster Data Services Planning and Administration Guide* for details about these options.

Before you register and configure HA for NFS, run the following command to verify that the HA for NFS package, `ha-cluster/data-service/nfs`, is installed on the cluster.

```
# pkg info -l ha-cluster/data-service/nfs
```

If the package has not been installed, see "Installing the HA for NFS Package" on page 12 for instructions on how to install the package.

Setting HA for NFS Extension Properties

The sections that follow contain instructions for registering and configuring resources. For information about the HA for NFS extension properties, see Appendix A, "HA for NFS Extension Properties." The Tunable entry indicates when you can update a property.

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 Data Services Planning and Administration Guide* to configure resources after the resources are created.

Tools for Registering and Configuring HA for NFS

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

- **The** clsetup(1CL) **utility.** For more information, see "How to Register and Configure HA for NFS (clsetup)" on page 13.
- **Oracle Solaris Cluster maintenance commands.** For more information, see "How to Register and Configure HA for NFS (Command Line Interface)" on page 17.

The clsetup utility provides a wizard for configuring HA for NFS. This wizard reduces the possibility for configuration errors that might result from command syntax errors or omissions. This 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 NFS (clsetup)

Perform this procedure during your initial set up of Oracle Solaris Cluster HA for NFS. Perform this procedure from one node only.

Note – The following instructions explain how to perform this operation by using the `clsetup` utility.

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

- Prerequisites for configuring a logical hostname resource are met.
- Prerequisites for configuring a highly available storage resource are met.
- The Oracle Solaris Cluster HA for NFS package is installed.

1 **Become superuser on any cluster node.**

2 **Ensure that the `hosts` and `rpc` entries in the `/etc/nsswitch.conf` file are correct.**

The example output shows a configuration that uses the NIS external naming service.

 a. Display the lookup entries for `hosts` and `rpc`.

  The following example shows the correct lookup entries.

```
# svccfg -s svc:/system/name-service/switch listprop config/host
hosts: cluster files [SUCCESS=return] nis
# svccfg -s svc:/system/name-service/switch listprop config/rpc
rpc: files nis
```

  - For hosts, files must follow cluster and precede any directory or name service. This modification enables HA for NFS to fail over correctly in the presence of public network failures.

  For rpc, files must precede any directory or name service. This configuration prevents timing-related errors for rpc lookups during periods of public network or name service unavailability.

 b. If a lookup entry is not correct, set the correct lookups.

  - For hosts:

```
# svccfg -s svc:/system/name-service/switch \
setprop config/host = astring: \"cluster files [SUCCESS=return] nis\"
```

  - For rpc:

```
# svccfg -s svc:/system/name-service/switch \
setprop config/rpc = astring: \"files nis\"
```

3 **Start the `clsetup` utility.**

```
# clsetup
```

The `clsetup` main menu is displayed.

4 Select the option for Data Services and press Return.

The Data Services menu is displayed.

5 Select the option for configuring Oracle Solaris Cluster HA for NFS and press Return.

The clsetup utility displays the list of prerequisites for performing this task.

6 Verify that the prerequisites are met, and press Return.

The clsetup utility displays a list of all cluster nodes that are online.

7 Select the nodes where you require Oracle Solaris Cluster HA for NFS 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 and 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 and press Return.**

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

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

The clsetup utility displays a list of existing logical hostname resources on the cluster.

9 You can choose any of the following options.

- **To use the existing logical hostname resource, select the option for the required logical hostname and press Return. Go to Step 12.**

- **To create a new logical hostname resource, type c and press Return. Go to Step 10**

10 Type the logical hostname and press Return.

The clsetup utility displays a list of existing logical hostname resources.

11 Select the option for the logical hostname resource to be created and press Return.

The clsetup creates the logical hostname resource.

12 To confirm your selection of logical hostname resource, type d and press Return.

The clsetup utility displays information about file system mount points.

13   Press Return to continue.

The clsetup utility displays the existing file system mount points.

14   Select the file system mount points for Oracle Solaris Cluster HA for NFS data files.

- **To select a subset of the listed file system mount points, type a comma-separated or space-separated list of the numbers that correspond to the file system node point and press Return.**

- **To select all file system mount points in a particular order, type a comma-separated or space-separated ordered list of the numbers that correspond to the file system mount points and press Return.**

15   To confirm your selection of file system mount points, type d and press Return.

The clsetup utility displays a screen where you can specify the path prefix for Oracle Solaris Cluster HA for NFS resource group.

16   Select the path prefix for Oracle Solaris Cluster HA for NFS resource group and press Return.

The clsetup utility displays a screen where you can change the share option for the file system mount point that the NFS server is sharing.

17   Select the share option and press Return.

The clsetup utility displays the share options for the selected mount points.

18   If you require a different name for any Oracle Solaris Cluster objects, change each name as follows.

a. Select the option for the name that you are changing and press Return.

The clsetup utility displays a screen where you can specify the new name.

b. At the New Value prompt, type the new name and press Return.

The clsetup utility returns you to the list of the names of the Oracle Solaris Cluster objects that the utility will create.

19   To confirm your selection of Oracle Solaris Cluster object names, type d and press Return.

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

20   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.

21 Press Return to continue.

The clsetup utility returns you to the Data Services Menu.

22 Type q and press Return.

The clsetup utility returns you to the Main Menu.

23 (Optional) Type q and press Return to quit the `clsetup` utility.

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

24 Determine whether the Oracle Solaris Cluster HA for NFS resource group and its resources are online.

Use the clresourcegroup command for this purpose. By default, the clsetup utility assigns the name nfs-mountpoint-admin-rg to the Oracle Solaris Cluster HA for NFS resource group.

```
# clresourcegroup status nfs-mountpoint-admin-rg
```

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

```
# clresourcegroup online -emM nfs-rg
```

▼ How to Register and Configure HA for NFS (Command Line Interface)

Before You Begin   Verify that all the cluster nodes are online.

```
# clnode status
```

1 On a cluster member, become superuser or assume a role that provides `solaris.cluster.admin` RBAC authorization.

2 Ensure that the `hosts` and `rpc` entries in the `/etc/nsswitch.conf` file are correct.

The example output shows a configuration that uses the NIS external naming service.

a. Display the lookup entries for `hosts` and `rpc`.

The following example shows the correct lookup entries.

```
# svccfg -s svc:/system/name-service/switch listprop config/host
hosts: cluster files [SUCCESS=return] nis
# svccfg -s svc:/system/name-service/switch listprop config/rpc
rpc: files nis
```

- For hosts, files must follow cluster and precede any directory or name service. This modification enables HA for NFS to fail over correctly in the presence of public network failures.

  For rpc, files must precede any directory or name service. This configuration prevents timing-related errors for rpc lookups during periods of public network or name service unavailability.

   b. **If a lookup entry is not correct, set the correct lookups.**

    - For hosts:

      ```
      # svccfg -s svc:/system/name-service/switch \
      setprop config/host = astring: \"cluster files [SUCCESS=return] nis\"
      ```

    - For rpc:

      ```
      # svccfg -s svc:/system/name-service/switch \
      setprop config/rpc = astring: \"files nis\"
      ```

3 Create the *Pathprefix* directory.

Create a *Pathprefix* directory on the HA file system (cluster file system or highly available local file system). HA for NFS resources will use this directory to maintain administrative information.

You can specify any directory for this purpose. However, you must manually create a *Pathprefix* directory for each resource group that you create.

```
# mkdir -p Pathprefix-directory
```

4 Create a failover resource group to contain the NFS resources.

```
# clresourcegroup create [-n nodelist] -p Pathprefix=Pathprefix-directory  resource-group
```

[-n *nodelist*]
   Specifies an optional, comma-separated list of physical node names or IDs that identify potential masters. The order here determines the order in which the Resource Group Manager (RGM) considers primary nodes during failover.

-p Pathprefix=*Pathprefix-directory*
   Specifies a directory that resources in this resource group will use to maintain administrative information. This is the directory that you created in Step 3.

resource-group
   Specifies the failover resource group.

5 Verify that you have added all your logical hostname resources to the name service database.

To avoid any failures because of name service lookups, verify that all IP addresses to hostname mappings that are used by Oracle Solaris Cluster HA for NFS are present in the server's and client's /etc/inet/hosts file.

6 (Optional) Customize the `nfsd` or `lockd` startup options.

Use the sharectl command to customize the nfsd and lockd options. For more information, see the nfsd(1M), lockd(1M), and sharectl(1M)man pages.

7 Add the desired logical hostname resources into the failover resource group.

You must set up a logical hostname resource with this step. The logical hostname that you use with Oracle Solaris Cluster HA for NFS *cannot* be a SharedAddress resource type.

`# clreslogicalhostname create -g` *resource-group* `-h` *logical-hostname, ...* [`-N` *netiflist*] *lhresource*

-g *resource-group*
    Specifies the resource group that is to hold the logical hostname resources.

-h *logical-hostname, …*
    Specifies the logical hostname resource to be added.

-N *netiflist*
    Specifies an optional, comma-separated list that identifies the IPMP groups that are on each node. Each element in *netiflist* must be in the form of netif@node. netif can be used as an IPMP 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 – 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.

---

8 From any cluster node, create the `SUNW.nfs` subdirectory.

Create a subdirectory called SUNW.nfs below the directory that the Pathprefix property identifies in Step 4.

`# mkdir` *Pathprefix-directory*`/SUNW.nfs`

9 Create a `dfstab.`*resource* file in the `SUNW.nfs` directory that you created in Step 8, and set up share options.

a. Create the *Pathprefix*`/SUNW.nfs/dfstab.`*resource* **file.**

This file contains a set of share commands with the shared path names. The shared paths should be subdirectories on a cluster file system.

Note – Choose a *resource* name suffix to identify the NFS resource that you plan to create in Step 11. A good resource name refers to the task that this resource is expected to perform. For example, a name such as user-nfs-home is a good candidate for an NFS resource that shares user home directories.

---

b. Set up the `share` options for each path that you have created to be shared.

The format of this file is exactly the same as the format that is used in the `/etc/dfs/dfstab` file.

```
# share -F nfs [-o specific_options] [-d "description"] pathname
```

-F nfs
   Identifies the file system type as `nfs`.

-o *specific_options*
   Grants read-write access to all the clients. See the share(1M) man page for a list of options. Set the `rw` option for Oracle Solaris Cluster.

-d *description*
   Describes the file system to add.

pathname
   Identifies the file system to share.

Note – If you want to share multiple paths, the above `share` command need to be repeated for each path that you are sharing.

When you set up your share options, consider the following points.

- When constructing `share` options, do not use the `root` option, and do not mix the `ro` and `rw` options.

- Do not grant access to the hostnames on the cluster interconnect.

   Grant read and write access to all the cluster nodes and logical hosts to enable the Oracle Solaris Cluster HA for NFS monitoring to do a thorough job. However, you can restrict write access to the file system or make the file system entirely read-only. If you do so, Oracle Solaris Cluster HA for NFS fault monitoring can still perform monitoring without having write access.

- If you specify a client list in the `share` command, include all the physical hostnames and logical hostnames that are associated with the cluster. Also include the hostnames for all the clients on all the public networks to which the cluster is connected.

- If you use net groups in the `share` command, rather than names of individual hosts, add all those cluster hostnames to the appropriate net group.

The share -o rw command grants write access to all the clients, including the hostnames that the Oracle Solaris Cluster software uses. This command enables Oracle Solaris Cluster HA for NFS fault monitoring to operate most efficiently. See the following man pages for details:

- share(1M)
- share_nfs(1M)

10   Register the NFS resource type.

clresourcetype register *resource-type*

resource-type
   Adds the specified resource type. For Oracle Solaris Cluster HA for NFS, the resource type is SUNW.nfs.

11   Create the NFS resource in the failover resource group.

clresource create -g *resource-group* **-t** *resource-type resource*

-g *resource-group*
   Specifies the name of a previously created resource group to which this resource is to be added.

-t *resource-type*
   Specifies the name of the resource type to which this resource belongs. This name must be the name of a registered resource type.

resource
   Specifies the name of the resource to add, which you defined in Step 9. This name can be your choice but must be unique within the cluster.

The resource is created in the enabled state.

12   Run the clresourcegroup command to manage the resource group.

clresourcegroup online -M *resource-group*

Example 1–1   Setting Up and Configuring HA for NFS

The following example shows how to set up and configure HA for NFS.

1. To create a logical host resource group and specify the path to the administrative files used by NFS (Pathprefix), the following command is run.

   # **clresourcegroup create -p Pathprefix=/global/nfs** *resource-group-1*

2. To add logical hostname resources into the logical host resource group, the following command is run.

   # **clreslogicalhostname create -g** *resource-group-1* **-h** *schost-1 lhresource*

3. To make the directory structure contain the Oracle Solaris Cluster HA for NFS configuration files, the following command is run.

```
# mkdir -p /global/nfs/SUNW.nfs
```

4. To create the dfstab.*resource* file under the nfs/SUNW.nfs directory and set share options, the following command is run.

```
# share -F nfs -o rw=engineering -d "home dirs" /global/nfs/SUNW.nfs
```

Note – You also need to add this entry to the dfstab.*resource* file.

---

5. To register the NFS resource type, the following command is run.

```
# clresourcetype register SUNW.nfs
```

6. To create the NFS resource in the resource group, the following command is run.

```
# clresource create -g resource-group-1 -t SUNW.nfs r-nfs
```

The resource is created in the enabled state.

7. To enable the resources and their monitors, manage the resource group, and switch the resource group into online state, the following command is run.

```
# clresourcegroup online -M resource-group-1
```

▼ How to Change Share Options on an NFS File System

If you use the rw, rw=, ro, or ro= options to the share -o command, NFS fault monitoring works best if you grant access to all the physical hosts or netgroups that are associated with all the Oracle Solaris Cluster servers.

If you use netgroups in the sharecommand, add all the Oracle Solaris Cluster hostnames to the appropriate netgroup. Ideally, grant both read access and write access to all the Oracle Solaris Cluster hostnames to enable the NFS fault probes to do a complete job.

Note – Before you change share options, read the share_nfs(1M) man page to understand which combinations of options are legal.

You can also modify the shared path and options dynamically without bringing offline the Oracle Solaris Cluster HA for NFS resource. See "How to Dynamically Update Shared Paths on an NFS File System" on page 24.

To modify the share options on an NFS file system while the Oracle Solaris Cluster HA for NFS resource is offline, perform the following steps.

1   On a cluster member, become superuser or assume a role that provides solaris.cluster.admin RBAC authorization.

2   Turn off fault monitoring on the NFS resource.

    # **clresource unmonitor** *resource*

3   Test the new share options.

a.   Before you edit the dfstab. *resource* **file with new share options, execute the new share command to verify the validity of your combination of options.**

    # **share -F nfs** [**-o** *specific_options*] [**-d "***description***"**] *pathname*

    -F nfs
       Identifies the file system type as NFS.

    -o *specific_options*
       Specifies an option. You might use rw, which grants read-write access to all the clients.

    -d *description*
       Describes the file system to add.

    pathname
       Identifies the file system to share.

b.   If the new share command fails, immediately execute another share command with the old options.

    When the new command executes successfully, proceed to Step 4.

4   Edit the dfstab. *resource* **file with the new share options.**

a.   To remove a path from the dfstab. *resource* **file, perform the following steps in order.**

i.   Run the unshare command.

    # **unshare -F nfs** [**-o** *specific_options*] *pathname*

    -F nfs
       Identifies the file system type as NFS.

    -o *specific_options*
       Specifies the options that are specific to NFS file systems.

    pathname
       Identifies the file system that is made unavailable.

ii.   From the dfstab. *resource* **file, delete the share command for the path that you want to remove.**

    # **vi dfstab.** *resource*

b.   To add a path or change an existing path in the dfstab. *resource* **file, verify that the mount point is valid, then edit the dfstab.** *resource* **file.**

5   Enable fault monitoring on the NFS resource.

```
# clresource monitor resource
```

▼ How to Dynamically Update Shared Paths on an NFS File System

You can dynamically modify the share command on an NFS file system without bringing offline the Oracle Solaris Cluster HA for NFS resource. The general procedure consists of modifying the dfstab.*resource* file for Oracle Solaris Cluster HA for NFS and then manually running the appropriate command, either the share command or the unshare command. The command is immediately effective, and Oracle Solaris Cluster HA for NFS handles making these paths highly available.

Ensure that the paths that are shared are always available to Oracle Solaris Cluster HA for NFS during failover so that local paths (on non-HA file systems) are not used.

If paths on a file system that is managed by HAStoragePlus are shared, the HAStoragePlus resource must be in the same resource group as the Oracle Solaris Cluster HA for NFS resource, and the dependency between them must be set correctly.

1   Use the `cluster status` command to find out the node on which the Oracle Solaris Cluster HA for NFS resource is online.

2   On this node, run the `/usr/sbin/share` command to see the list of paths currently shared and determine the changes to make to this list.

3   To add a new shared path, perform the following steps.

a.   Add the `share` command to the `dfstab.`*resource* file.

Oracle Solaris Cluster HA for NFS shares the new path the next time it checks the file. The frequency of these checks is controlled by the Thorough_Probe_Interval property (by default 120 seconds).

b.   Run the `share` command manually to make the newly added shared path effective immediately.

Running the command manually ensures that the shared paths are available to potential clients. Oracle Solaris Cluster HA for NFS detects that the newly added path is already shared and does not report an error.

4   To unshare a path, perform the following steps.

   a.   Run the `dfmounts` command to ensure that no clients are currently using the path.

   Although a path can be unshared even if clients are using it, these clients would receive a stale file error handle and would need special care (forced unmount, or even reboot) to recover.

   b.   Remove the `share` command from the `dfstab.`*resource* file.

   c.   Run the `unshare` command manually.

5   To modify options for an existing shared path, perform the following steps.

   a.   Modify the `dfstab.`*resource* file as needed.

   b.   Run the appropriate command (`share` or `unshare`) manually.

How to Tune HA for NFS Method Timeouts

The time that HA for NFS methods require to finish depends on the number of paths that the resources share through the `dfstab.`*resource* file. The default timeout for these methods is 300 seconds.

As a general guideline, allocate 10 seconds toward the method timeouts for each path that is shared. Default timeouts are designed to handle 30 shared paths.

- If the number of shared paths is less than 30, do not reduce the timeout.
- If the number of shared paths exceeds 30, multiply the number of paths by 10 to compute the recommended timeout. For example, if the `dfstab.`*resource* file contains 50 shared paths, the recommended timeout is 500 seconds.

Update the following method timeouts if the number of shared paths is greater than 30.

- `Prenet_start_timeout`
- `Postnet_stop_timeout`
- `Start_timeout`
- `Stop_timeout`
- `Validate_timeout`
- `Update_timeout`
- `Monitor_Start_timeout`
- `Monitor_Stop_timeout`
- `Monitor_Check_timeout`

To change method timeouts, use the `clresource set` command, as in the following example.

```
% clresource set -p Prenet_start_timeout=500 resource
```

Configuring SUNW.HAStoragePlus Resource Type

HA for NFS is a disk-intensive data service. Therefore, you should configure the
SUNW.HAStoragePlus resource type for use with this data service. For an overview of the
SUNW.HAStoragePlus resource type, see "Understanding HAStoragePlus" in *Oracle Solaris
Cluster Data Services Planning and Administration Guide*.

The procedure for configuring the SUNW.HAStoragePlus resource type depends on the type of
the file system that NFS is sharing. For more information, see the following sections:

- "How to Set Up the HAStoragePlus Resource Type for an NFS-Exported UNIX File System
  Using the Command Line Interface" on page 26
- "How to Set Up the HAStoragePlus Resource Type for an NFS-Exported ZFS" on page 28

▼ How to Set Up the HAStoragePlus Resource Type for an NFS-Exported UNIX File System Using the Command Line Interface

The HAStoragePlus resource type synchronizes the startups between resource groups and disk
device groups. The HAStoragePlus resource type has an additional feature to make a local file
system highly available. For background information about making a local file system highly
available, see "Enabling Highly Available Local File Systems" in *Oracle Solaris Cluster Data
Services Planning and Administration Guide*. To use both of these features, set up the
HAStoragePlus resource type.

Note – These instructions explain how to use the HAStoragePlus resource type with the UNIX
file system (UFS).

The following example uses a simple NFS service that exports home directory data from a
locally mounted directory /global/local-fs/nfs/export/ home. The example assumes the
following:

- The mount point /global/local-fs/nfs is used to mount a UFS local file system on an
  Oracle Solaris Cluster global device partition.
- The /etc/vfstab entry for the /global/local-fs/nfs file system should omit the global
  option and specify that the mount at boot flag is no.
- The path-prefix directory is on the root directory of the same file system that is to be
  mounted, for example, /global/local-fs/nfs. The path-prefix directory is the directory
  that HA-NFS uses to maintain administrative information and status information.

1   On a cluster node, become superuser or assume a role that provides `solaris.cluster.modify` RBAC authorization.

2   Determine whether the `HAStoragePlus` resource type and the `SUNW.nfs` resource type are registered.

The following command prints a list of registered resource types.

```
# clresourcetype show | egrep Type
```

3   If necessary, register the `HAStoragePlus` resource type and the `SUNW.nfs` resource type.

```
# clresourcetype register SUNW.HAStoragePlus
# clresourcetype register SUNW.nfs
```

4   Create the failover resource group `nfs-rg`.

```
# clresourcegroup create -p PathPrefix=/global/local-fs/nfs nfs-rg
```

5   Create a logical host resource of type `SUNW.LogicalHostname`.

```
# clreslogicalhostname create -g nfs-rg -h log-nfs nfs-lh-rs
```

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.

---

6   Create the resource `nfs-hastp-rs` of type `HAStoragePlus`.

```
# clresource create -g nfs-rg -t SUNW.HAStoragePlus \
-p FilesystemMountPoints=/global/local-fs/nfs \
-p AffinityOn=True nfs-hastp-rs
```

The resource is created in the enabled state.

Note – You can use the FilesystemMountPoints extension property to specify a list of one or more mount points for file systems. This list can consist of mount points for both local file systems and global file systems. The mount at boot flag is ignored by HAStoragePlus for global file systems.

---

7   Bring online the resource group `nfs-rg` on a cluster node.

The node or zone where the resource group is brought online becomes the primary node for the /global/local-fs/nfs file system's underlying global device partition. The file system /global/local-fs/nfs is then mounted on this node or zone.

```
# clresourcegroup online -M nfs-rg
```

8 Create the resource `nfs-rs` of type `SUNW.nfs` and specify its resource dependency on the resource `nfs-hastp-rs`.

The file `dfstab.nfs-rs` must be present in `/global/local-fs/nfs/SUNW.nfs`.

```
# clresource create -g nfs-rg -t SUNW.nfs \
-p Resource_dependencies_offline_restart=nfs-hastp-rs nfs-rs
```

The resource is created in the enabled state.

Note – Before you can set the dependency in the `nfs-rs` resource, the `nfs-hastp-rs` resource must be online.

---

9 Take offline the resource group `nfs-rg`.

```
# clresourcegroup offline nfs-rg
```

10 Bring online the `nfs-rg` group on a cluster node or zone.

```
# clresourcegroup online -M nfs-rg
```

Caution – Ensure that you switch only the resource group. Do *not* attempt to switch the device group. If you attempt to switch the device group, the states of the resource group and the device group become inconsistent, causing the resource group to fail over.

Whenever the service is migrated to a new node, the primary I/O path for `/global/local-fs/nfs` will always be online and colocated with the NFS servers. The file system `/global/local-fs/nfs` is locally mounted before the NFS server is started.

▼ How to Set Up the HAStoragePlus Resource Type for an NFS-Exported ZFS

The following procedure uses a simple NFS service.

See "Creating a Basic ZFS Storage Pool" in *Oracle Solaris Administration: ZFS File Systems* for information about how to create a ZFS pool. See "Creating a ZFS File System Hierarchy" in *Oracle Solaris Administration: ZFS File Systems* for information about how to create a ZFS file system in that ZFS pool.

1 On a cluster node, become superuser or assume a role that provides `solaris.cluster.modify` RBAC authorization.

2 Determine whether the `HAStoragePlus` resource type and the `SUNW.nfs` resource type are registered.

The following command prints a list of registered resource types.

```
# clresourcetype list
```

3   If necessary, register the `HAStoragePlus` resource type and the `SUNW.nfs` resource type.

```
# clresourcetype register SUNW.HAStoragePlus SUNW.nfs
```

4   Create the failover resource group.

```
# clresourcegroup create -p PathPrefix=path resource-group
```

5   Create a logical host resource of type `SUNW.LogicalHostname`.

```
# clreslogicalhostname create -g resource-group \
-h logical-hostname logicalhost-resource
```

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.

---

6   Create the ZFS file system resource of type `HAStoragePlus`.

```
# clresource create -g resource-group -t SUNW.HAStoragePlus \
-p Zpools=zpool HASP-resource
```

The resource is created in the enabled state.

Note – You can specify a list of one or more ZFS pools for the `Zpools` extension property.

---

7   Bring online the resource group on a cluster node in a managed state.

The node on which the resource group is brought online becomes the primary node for the ZFS file system. The ZFS pool *zpool* is imported on this node. The ZFS file system is consequently mounted locally on this node.

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

8   Create the resource of type `SUNW.nfs` and specify its resource dependency on the resource of type `SUNW.HAStoragePlus`.

The file dfstab.nfs-rs must be present in *zpool*/nfs/SUNW.nfs.

```
# clresource create -g resource-group -t SUNW.nfs \
-p resource_dependencies_offline_restart=HASP-resource NFS-resource
```

The resource is created in the enabled state.

Note – Before you can set the dependency in the *NFS-resource* resource, the *HASP-resource* resource must be online.

---

9   Bring online the *resource-group* group on a cluster node in a managed state.

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

Example 1–2   Setting Up the HAStoragePlus Resource Type for an NFS-Exported ZFS File System

The following example uses a simple NFS service. The example assumes the following:

- The nfs/export directory exists in the ZFS pool /nfszpool.
- The dfstab.*resource* file exists in the /nfszpool/nfs/SUNW.nfs directory.
- The path-prefix directory is on the root directory of the same file system that is to be mounted, for example, /nfszpool/nfs. The path-prefix directory is the directory that HA-NFS uses to maintain administrative information and status information.

```
phys-schost-1% su
Password:
# clresourcetype list
SUNW.LogicalHostname:2
SUNW.SharedAddress:2
# clresourcetype register SUNW.HAStoragePlus SUNW.nfs
# clresourcegroup create -p PathPrefix=/nfszpool/nfs nfs-rg
# clreslogicalhostname create -g nfs-rg -h log-nfs nfs-lh-rs
# clresource create -g nfs-rg -t SUNW.HAStoragePlus \
  -p Zpools=nfszpool nfs-hastp-rs
# clresourcegroup online -M nfs-rg
# clresource create -g nfs-rg -t SUNW.nfs \
  -p resource_dependencies_offline_restart=nfs-hastp-rs nfs-rs
```

Tuning the HA for NFS Fault Monitor

The HA for NFS fault monitor is contained in a resource whose resource type is SUNW.nfs.

For general information about the operation of fault monitors, see "Tuning Fault Monitors for Oracle Solaris Cluster Data Services" in *Oracle Solaris Cluster Data Services Planning and Administration Guide*.

Fault Monitor Startup

The NFS resource MONITOR_START method starts the NFS system fault monitor. This start method first checks if the NFS system fault monitor nfs_daemons_probeis already running under the process monitor daemon rpc.pmfd. If the NFS system fault monitor is not running, the start method starts the nfs_daemons_probe process under the control of the process monitor. The start method then starts the resource fault monitor nfs_probe, also under the control of the process monitor.

Fault Monitor Stop

The NFS resource MONITOR_STOP method stops the resource fault monitor. If no other NFS resource fault monitor is running on the local node, the stop method stops the NFS system fault monitor.

Operations of HA for NFS Fault Monitor During a Probe

This section describes the operations of the following fault monitoring processes:

- NFS system fault monitoring
- NFS resource fault monitoring
- Monitoring of file sharing

NFS System Fault Monitoring Process

The NFS system fault monitor probe monitors the NFS daemons `nfsd`, `mountd`, `statd`, and `lockd`, and the RPC portmapper service daemon `rpcbind`on the local node. The probe checks for the presence of the process and its response to a null `rpc` call. This monitor uses the following NFS extension properties:

- `Rpcbind_nullrpc_timeout`
- `Rpcbind_nullrpc_reboot`
- `Statd_nullrpc_timeout`
- `Lockd_nullrpc_timeout`
- `Mountd_nullrpc_timeout`
- `Mountd_nullrpc_restart`
- `Nfsd_nullrpc_timeout`
- `Nfsd_nullrpc_restart`

See "Setting HA for NFS Extension Properties" on page 13.

Each NFS system fault monitor probe cycle performs the following steps in a loop. The system property `Cheap_probe_interval` specifies the interval between probes.

1. The fault monitor probes `rpcbind`.

    If the process terminates unexpectedly, but a warm restart of the daemon is in progress, `rpcbind` continues to probe other daemons.

    If the process terminates unexpectedly, the fault monitor reboots the node.

    If a null `rpc` call to the daemon terminates unexpectedly, `Rpcbind_nullrpc_reboot=True`, and `Failover_mode=HARD`, the fault monitor reboots the node.

2. The fault monitor probes `statd` first, and then `lockd`.

    If `statd` or `lockd` terminate unexpectedly, the system fault monitor attempts to restart both daemons.

    If a null `rpc` call to these daemons terminates unexpectedly, the fault monitor logs a message to `syslog` but does not restart `statd` or `lockd`.

3. The fault monitor probes `mountd`.

    If `mountd` terminates unexpectedly, the fault monitor attempts to restart the daemon.

If the null `rpc` call to the daemon terminates unexpectedly and
`Mountd_nullrpc_restart=True`, the fault monitor attempts to restart `mountd` if the cluster
file system is available.

4. The fault monitor probes `nfsd`.

   If `nfsd` terminates unexpectedly, the fault monitor attempts to restart the daemon.

   If the null `rpc` call to the daemon terminates unexpectedly and
   `Nfsd_nullrpc_restart=TRUE`, the fault monitor attempts to restart `nfsd` if the cluster file
   system is available.

5. If any one of the above NFS daemons (except `rpcbind`) fails to restart during a probe cycle,
   the NFS system fault monitor will retry the restart in the next cycle. When all the NFS
   daemons are restarted and healthy, the resource status is set to `ONLINE`. The monitor tracks
   unexpected terminations of NFS daemons in the last `Retry_interval`. When the total
   number of unexpected daemon terminations has reached `Retry_count`, the system fault
   monitor issues a `scha_control` giveover. If the giveover call fails, the monitor attempts to
   restart the failed NFS daemon.

6. At the end of each probe cycle, if all daemons are healthy, the monitor clears the history of
   failures.

NFS Resource Fault Monitoring Process

NFS resource fault monitoring is specific to each NFS resource. The fault monitor of each
resource checks the status of each shared path to monitor the file systems that the resource
exports.

Before starting the NFS resource fault monitor probes, all the shared paths are read from the
`dfstab` file and stored in memory. In each probe cycle, the probe performs the following steps.

1. If `dfstab` has been changed since the last read, the probe refreshes the memory.

   If an error occurs while reading the `dfstab` file, the resource status is set to `FAULTED`, and the
   monitor skips the remainder of the checks in the current probe cycle.

2. The fault monitor probes all the shared paths in each iteration by performing `stat()` on the
   path.

   If any path is not functional, the resource status is set to `FAULTED`.

3. The probe checks for the presence of NFS daemons (`nfsd`, `mountd`, `lockd`, `statd`) and
   `rpcbind`.

4. If any of these daemons are down, the resource status is set to `FAULTED`.

5. If all shared paths are valid and NFS daemons are present, the resource status is reset to
   `ONLINE`.

Monitoring of File Sharing

The Oracle Solaris Cluster HA for NFS fault monitor probe monitors the success or failure of file sharing by monitoring the following files:

- `/etc/dfs/sharetab`

- `/etc/mnttab`

- *Pathprefix*/`SUNW.nfs/dfstab.`*resource*

  The *Pathprefix* part of the file path is the value of the `Pathprefix` extension property for the resource group, and *resource* is the resource name.

If the probe detects any modification to any of these files, it shares the paths in `dfstab.`*resource* again.

Upgrading the SUNW.nfs Resource Type

For general instructions that explain how to upgrade a resource type, see "Upgrading a Resource Type" in *Oracle Solaris Cluster Data Services Planning and Administration Guide*. The information that you require to complete the upgrade of the resource type is provided in the subsections that follow.

Information for Registering the New Resource Type Version

The release of Oracle Solaris Cluster data services indicates the release in which the version of the resource type was introduced.

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/SUNWscnfs/etc/SUNW.nfs`.

Information for Migrating Existing Instances of the Resource Type

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

- You must perform the migration when the resource group is in an unmanaged state.

- For Oracle Solaris Cluster 4.0, the required value of the `Type_version` property is `3.3`.

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

EXAMPLE 1–3 Migrating Instances of the SUNW.nfs Resource Type

```
# clresource set -p Type_version=3.3 nfs-rs
```

This command modifies the Type_version property of the nfs-rs resource to 3.3.

A
◆ ◆ ◆   A P P E N D I X   A

HA for NFS Extension Properties

This section describes the extension properties for the resource type SUNW.nfs. This resource type represents the Network File System (NFS) application in an Oracle Solaris Cluster configuration.

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

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

Lockd_nullrpc_timeout
   The time-out value (in seconds) to use when probing lockd

Data type	Integer
Default	120
Range	Minimum = 60
Tunable	At any time

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. Note that this property refers to restarts of the fault monitor itself, rather than the resource.

Data type	Integer
Default	4
Range	0 – 2,147,483,641
	A value of –1 indicates an infinite number of restart 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 is more than the value that is specified in the extension property `Monitor_retry_count` within this period, the PMF restarts the fault monitor.

Data type	Integer
Default	2
Range	0 – 2,147,483,641
	−1 indicates an infinite amount of time.
Tunable	At any time

`Mountd_nullrpc_restart`

A Boolean to indicate whether to restart `mountd` when a null `rpc` call fails.

Data type	Boolean
Default	True
Range	Not applicable
Tunable	At any time

`Mountd_nullrpc_timeout`

The time-out value (in seconds) to use when probing `mountd`.

Data type	Integer
Default	120
Range	Minimum = 60
Tunable	At any time

`Nfsd_nullrpc_restart`

A Boolean to indicate whether to restart `nfsd` when a null `rpc` call fails.

Data type	Boolean
Default	False
Range	Not applicable
Tunable	At any time

`Nfsd_nullrpc_timeout`

The time-out value (in seconds) to use when probing `nfsd`.

Data type	Integer
Default	120
Range	Minimum = 60

Tunable   At any time

Rpcbind_nullrpc_reboot
A Boolean to indicate whether to reboot the system when a null rpc call on rpcbind fails.

Data type   Boolean

Default   False

Range   Not applicable

Tunable   At any time

Rpcbind_nullrpc_timeout
The time-out value (in seconds) to use when probing rpcbind.

Data type   Integer

Default   120

Range   Minimum = 60

Tunable   At any time

Statd_nullrpc_timeout
The time-out value (in seconds) to use when probing statd.

Data type   Integer

Default   120

Range   Minimum = 60

Tunable   At any time

Index