IRIS FailSafe™ 2.0 NFS™ Administrator's Guide

Document Number 007-3949-001

CONTRIBUTORS

Written by Anita Manders
Illustrated by Dany Galgani
Edited by Rick Thompson
Production by Heather Hermstad
Engineering contributions by Gilberto Arnaiz, Chander Kant, Bill Sparks, and Paddy Sreenivasan
St. Peter's Basilica image courtesy of ENEL SpA and InfoByte SpA. Disk Thrower image courtesy of Xavier Berenguer, Animatica.

IRIS FailSafe™ 2.0 NFS™ Administrator's Guide
Document Number 007-3949-001

Contents

List of Figures

List of Tables

List of Examples

About This Guide

This guide provides information about configuring IRIS FailSafe 2.0 systems with the IRIS FailSafe 2.0 NFS option. This option enables NFS resources to be failed over from one node to another if a component fails. This guide is intended as a supplement to the information about configuring IRIS FailSafe included in the *IRIS FailSafe 2.0 Administrator's Guide*.

This guide was prepared in conjunction with the initial release of the IRIS FailSafe 2.0 NFS option and the 2.0 Release of the IRIS FailSafe software product. It describes IRIS FailSafe NFS software for NFS.

Audience

This guide is written for system administrators who are responsible for configuring and administering an IRIS FailSafe system with the optional IRIS FailSafe NFS software. These system administrators must be able to customize several shell scripts and must be familiar with NFS configuration and NFS startup and shutdown procedures.

Structure of This Document

This guide contains the following chapters:

- Chapter 1, "IRIS FailSafe NFS Overview," describes the IRIS FailSafe components that are included in the IRIS FailSafe NFS option. It also provides an overview on the steps required to add NFS as a highly available service to an IRIS FailSafe cluster.
- Chapter 2, "Configuring IRIS FailSafe for NFS," explains the procedures for adding NFS as a highly available service.

Related Documentation

For NFS installation information, see the *ONC3/NFS Administrator's Guide*.

Besides this guide, other documentation for the IRIS FailSafe system includes

- *IRIS FailSafe 2.0 Administrator's Guide*
- *IRIS FailSafe 2.0 Programmer's Guide*
- *IRIS FailSafe 2.0 INFORMIX Administrator's Guide*
- *IRIS FailSafe 2.0 Netscape Server Administrator's Guide*
- *IRIS FailSafe 2.0 Oracle Administrator's Guide*

The IRIS FailSafe reference pages are as follows:

- `failsafe`(7M)
- `ha_cilog`(1M)
- `ha_cmsd`(1M)
- `ha_exec2`(1M)
- `ha_filelock`(1M)
- `ha_fileunlock`(1M)
- `ha_fsadm`(1M)
- `ha_fsd`(1M)
- `ha_gcd`(1M)
- `ha_http_ping2`(1M) (IRIS FailSafe Netscape Web option)
- `ha_ifdadmin`(1M)
- `ha_ifd`(1M)
- `ha_ifmx2`(1M) (IRIS FailSafe INFORMIX option)
- `ha_macconfig2`(1M)
- `ha_mom`(1M)
- `ha_srmd`(1M)
- `ha_statd2`(1M)

Release notes are included with each IRIS FailSafe product. The names of the release notes are as follows:

Table i IRIS FailSafe Release Notes

Release Note	Product
`failsafe2`	IRIS FailSafe 2.0
`failsafe2_nfs`	IRIS FailSafe 2.0 NFS
`failsafe2_web`	IRIS FailSafe 2.0 Netscape

Conventions Used in This Guide

These type conventions and symbols are used in this guide:

Italics New terms, manual/book titles, variable command-line arguments, and variables to be supplied by the user in examples, code, and syntax statements

`Fixed-width type`
Command names and command line arguments, file names, directory names, resource types, and examples of command output and code

`Bold fixed-width type`
Commands and text that you are to type literally in response to shell and command prompts

`#` IRIX shell prompt for the super user (`root`)

Chapter 1

IRIS FailSafe NFS Overview

This chapter provides information about the components that are added to the base IRIS FailSafe 2.0 product by the IRIS FailSafe 2.0 NFS option. It assumes that you are familiar with the basic components of IRIS FailSafe described in Chapter 1 of the *IRIS FailSafe 2.0 Administrator's Guide*. This chapter also lists the steps you must take to add NFS to the highly available services that are failed over on an IRIS FailSafe cluster.

The major sections in this chapter are as follows:

- "IRIS FailSafe NFS File System Monitoring" on page 1
- "Overview of Configuring IRIS FailSafe for NFS" on page 3

IRIS FailSafe NFS File System Monitoring

The IRIS FailSafe NFS option adds file system support to the IRIS FailSafe base software. The file system monitoring scripts monitor the NFS file systems. If the monitor script detects a failure, it will report a local monitoring failure to FailSafe, and FailSafe will take the appropriate action. Figure 1-1 shows an example of the message paths in IRIS FailSafe.

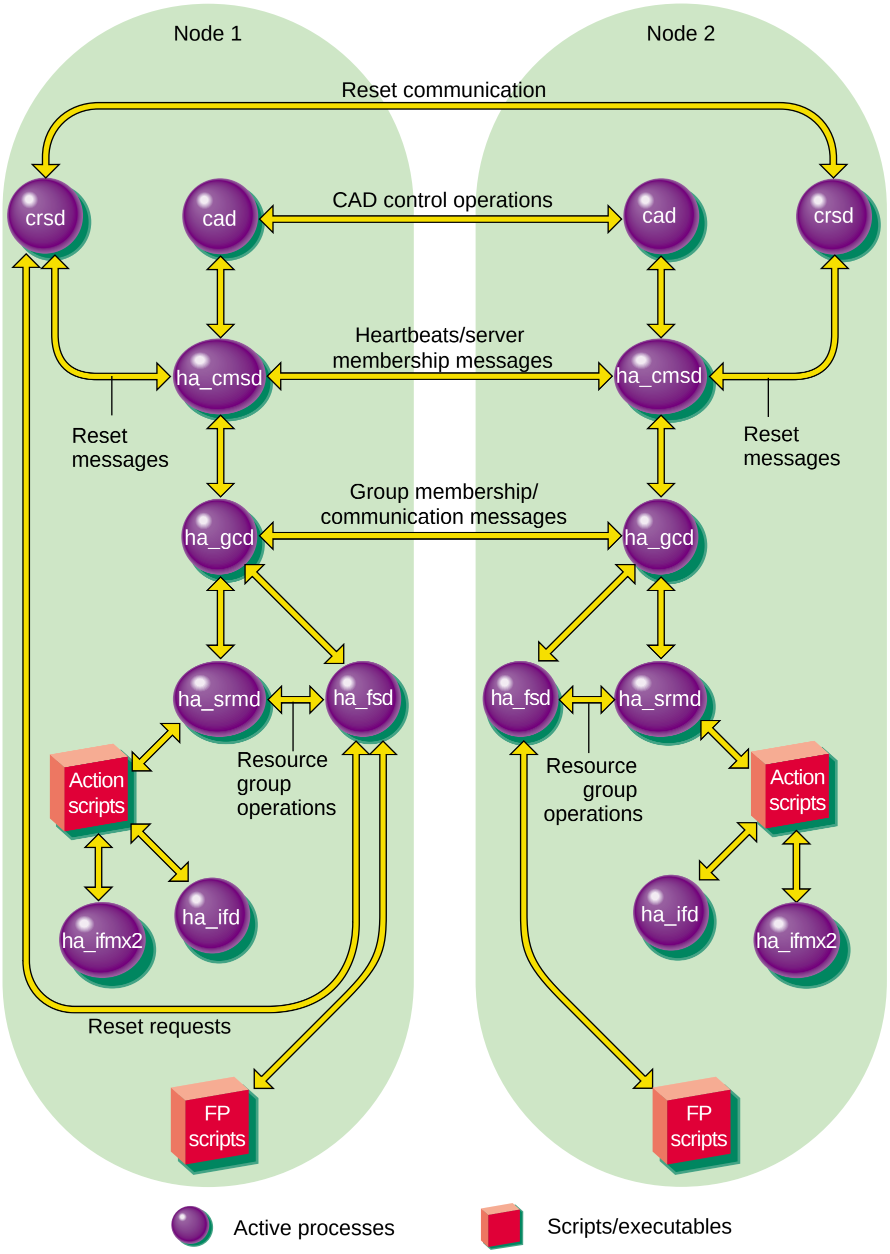

Figure 1-1 Message Paths in IRIS FailSafe

Overview of Configuring IRIS FailSafe for NFS

To configure an IRIS FailSafe cluster for failover of NFS, follow these steps:

1. Install, configure, and test the base IRIS FailSafe software as described in the *IRIS FailSafe 2.0 Administrator's Guide*.
2. Install additional software if required. See the section “Required Software” in Chapter 2.
3. Configure the NFS file system. See the section, “Configuring NFS File Systems” in Chapter 2.
4. Add the NFS resource type, resource attribute, and resource information to the configuration database (CDB) file. See the section “Adding NFS Resource Information to the Configuration Database (CDB)” in Chapter 2.
5. Test the NFS failover. See the section “Testing the NFS Resource” in Chapter 2.

Chapter 2

Configuring IRIS FailSafe for NFS

This chapter provides information about configuring the IRIS FailSafe NFS file system option for use on an IRIS FailSafe system. The major sections in this chapter are as follows:

- "Required Software" on page 5
- "Configuring NFS File Systems" on page 10
- "Adding NFS Resource Information to the Configuration Database (CDB)" on page 10
- "Testing the NFS Resource" on page 16

Required Software

The required software for NFS failover is as follows:

- NFS software

See the *ONC3/NFS Administrator's Guide* for more information about NFS.

- Base IRIS FailSafe software

  See the section "Installing the IRIS FailSafe Software" in Chapter 2 of the *IRIS FailSafe 2.0 Administrator's Guide* for a complete list of required base software.

- IRIS FailSafe NFS software
  - The *IRIS FailSafe 2.0 NFS Administrator's Guide* subsystem is `failsafe2_nfs.books.book_AG`.
  - The man page subsystem is `failsafe2_nfs.man.man`.
  - The release notes subsystem is `failsafe2_nfs.man.relnotes`.
  - The base software subsystem is `failsafe2_nfs.sw.base`.

Planning NFS File Systems

The IRIS FailSafe 2.0 NFS option enables IRIS FailSafe 2.0 to provide failover protection for the following:

- File systems that have been configured to be exported by the `exportfs` command
- NFS file-locking information, as supported by `rpc.lockd/rpc.statd`

In an IRIS FailSafe 2.0 cluster, one or more nodes can export NFS file systems. If a node that exports NFS file systems fails, another node provides backup service.

When the NFS server on one node fails, a surviving node must take over the NFS file-locking information from the failed node. IRIS FailSafe 2.0 does this by storing the NFS locking state for each exported file system. The information for each node is stored on that node in a single directory that is in one of the file systems on the shared disk. This directory is called the `statmon` directory and its name must be `statmon`.

Do not place NFS file systems on shared disks in the `/etc/exports` file. IRIS FailSafe 2.0 exports these file systems only after ensuring that another node does not have these file systems exported.

Note: The use of the `wsync` option is applicable only to NFS version 2, as NFS version 3 implements a safe asynchronous write. You can reconfigure NFS exports and XFS local mounts on the node to perform writes synchronously to disk (when received) by adding the parameter `wsync` to the `mode` parameter in all file system blocks and to the `export-info` parameter in all NFS blocks. Then, a reply to an NFS write will not be returned until the NFS write data is written to the server's disk. However, performance can be greatly affected by adding the `wsync` option in the file system blocks and in the NFS blocks. You must balance the risk of NFS data corruption during a node failure against the performance gain from using asynchronous NFS writes. Some applications perform their own error checking of data or perform writes in which data corruption does not occur.

FailSafe/NFS Servicing Diskless Clients

The following steps are required for FailSafe/NFS to act as a diskless NFS server to the Silicon Graphics diskless workstations. It assumes a detailed knowledge of IRIS FailSafe and the diskless workstation. (For more information on diskless workstations, see the *Diskless Workstation Administration Guide*.)

1. The IRIS FailSafe cluster should be set up and IRIS FailSafe/NFS should be fully configured and tested before starting the diskless installation. IRIS FailSafe should be failing over a single volume and filesystem; for example, `/diskless`. This will help define the variables that are required during the installation. After FailSafe is configured and running, do not perform any additional failovers until the diskless workstation is booting cleanly.
2. All nodes in the cluster must have all diskless boot patches.
3. During the diskless installation, use the floating FailSafe IP alias as the NFS server's IP address, called the FailSafe/NFS IP address.
4. Ensure that the client workstation's boot prom environment variables reference the FailSafe/NFS IP address and not specific node hostnames.
5. Verify that the FailSafe/NFS IP address has been used in the new `/etc/bootparams` entries added by the installation scripts. After completing the diskless installation on one node, the `/etc/bootparams` file must be copied to the peer node.
6. Verify that the /*diskless-filesystem*/`client`/*client-hostname*/`etc/fstab` file uses the FailSafe/NFS IP address instead of the node hostnames.
7. Duplicate the `/var/boot/`*client-hostname* directory tree on all nodes in the cluster.
8. Remove the NFS file systems that were added to the `/etc/exports` file on the diskless installation node. Transfer the information to the CDB FailSafe configuration file. Remove the original failover file system.

Example NFS File System Configuration with Two Nodes

An example configuration for a simple NFS file system configuration is as follows. If the node named `stocks` exports the NFS file system `/shared1` and the node named `bonds` exports the NFS file system `/shared2`, the two directories `/shared1/statmon` and `/shared2/statmon` should be created to hold the NFS file-locking information using the NFS client interface IP addresses of `1.2.3.4` and `1.2.3.5`, respectively. The directory `/shared1/statmon` stores NFS lock information that lets `bonds` recover the NFS locks for the `/shared1` file system if `stocks` fails. The directory `/shared2/statmon` stores NFS lock information that lets `stocks` recover NFS locks for `/shared2` if `bonds` fails.

Configuration Parameters for NFS File Systems

Table 2-1 lists the resource names and configuration parameters for the NFS file system in the configuration example described in the previous section, "Example NFS File System Configuration with Two Nodes."

Table 2-1 NFS Configuration Parameters

Resource Parameters	Value for stocks	Value for bonds	Comments
`resource-name`	`/shared1`	`/shared2`	Export mount point, which defines the export disk name used as input to the `exportfs`(1M) command.
resource attributes			
`export-info`	`rw,wsync`	`rw,wsync`	Export options, which lists the export options for the file system used in the `exportfs`(1M) command.
`filesystem`	`/shared1`	`/shared2`	Name of the file system that is used as input to the `mount`(1M) command.
resource dependencies			
`filesystem`	`/shared1`	`/shared2`	Name of the `filesystem` resource on which this `NFS` resource is dependent.
`statd`	`/shared1/statmon`	`/shared2/statmon`	Name of the `statd` resource on which this `NFS` resource is dependent.

To configure a complete NFS resource group, both the `NFS` resource and the `statd` resource must be defined. The `statd` resource defines how to make the NFS locks highly-available. See Table 2-2 for the list of the resource names and configuration parameters for the NFS `statd` resource.

Table 2-2 statd Configuration Parameters

Resource Parameters	Value for stocks	Value for bonds	Comments
`resource-name`	`/shared1/statd`	`/shared2/statd`	NFS lock directory, which stores information about the NFS client locks.
resource attributes			
`interface address`	`1.2.3.4`	`1.2.3.5`	IP_address of the highly-available interface that the NFS clients will use.
resource dependencies			
`filesystem`	`/shared1`	`/shared2`	Name of the `filesystem` resource on which this `statd` resource is dependent.
`IP_address`	`1.2.3.4`	`1.2.3.5`	Name of the `IP_address` resource on which this `statd` resource is dependent.

The procedure for configuring NFS file systems for IRIS FailSafe 2.0 is described in the following section, “Configuring NFS File Systems.”

Configuring NFS File Systems

You can use the following procedure to perform the NFS configuration for the file systems that will be failed over and for the status monitor `statd`. Note that no entries in `/etc/exports` are required for these file systems; IRIS FailSafe 2.0 software exports the NFS file systems.

1. Create or identify the file systems to be failed over. They must follow the guidelines in the section “Planning NFS File Systems.”
2. On each node that exports file systems, create the `statmon` directory on any one of the exported file systems on a shared disk. The name of the directory must be `statmon`.
3. On one node, open the file `/etc/config/statd.options` for editing (it may be new).
4. Put `-h` on the first line of the file:

   ```
   -h
   ```

5. Close the file.
6. Set the owner, group, and mode of the file, if you just created it:

   ```
   # chown root.sys /etc/config/statd.options
   # chmod 644 /etc/config/statd.options
   ```

7. Repeat steps 3 through 6 on the other nodes.

Adding NFS Resource Information to the Configuration Database (CDB)

This section describes procedures that show you how to create an NFS resource type, resource, resource group, and how to test the NFS resource. These procedures assume that a CDB that does not include NFS has already been created, installed, and tested as described in the *IRIS FailSafe 2.0 Administrator's Guide*.

Creating an NFS Resource Type

To create an NFS resource type, this subsection assumes that you are already familiar with the concepts of resource types. The NFS resource type defines the following resource attributes. See Table 2-1 for the definitions of these resource attributes.

- resource-name, which defines the name of the resource and is also the export disk name used as input to the exportfs(1M) command
- export-info, which lists the export options for the file system used in the exportfs(1M) command
- filesystem, which is the name of the file system that is used as input to the mount(1M) command

The NFS resource type is created at cluster creation time. If this automatic resource creation fails, the administrator must create the resource type before an NFS resource is created. The NFS resource type must be installed if you want to add an NFS resource to a cluster that was created before the NFS software was installed.

Example 2-1 NFS Installation Process

```
Inst> l *
 View: distribution
 Status: N=new, U=upgrade, S=same, D=downgrade
 Selection: i=install, r=remove, k=keep

 Subsystem Types [bdrpc]: b=reBoot needed, d=Default, r=Required,
 p=patch, c=Client only

i U failsafe2_nfs.books.book_AG [d] 0 FailSafe NFS Plugin 2.0
Administrator’s Guide
i U failsafe2_nfs.man.man [d] 0 FailSafe NFS Plugin 2.0 man pages
i U failsafe2_nfs.man.relnotes 0 FailSafe NFS Plugin 2.0 release notes
i U failsafe2_nfs.sw.base [d] 0 FailSafe NFS Plugin 2.0 base software

Disk space summary (Kbytes):            /  /nfs_cm1     /apps

Current free space                2272380   4422412   5656968
- Selections net change                 0         0         0
- Temporary inst overhead            1296+        0         0
= Minimum free during install     2271084   4422412   5656968

Final projected free space        2272380   4422412   5656968
```

```
Inst> g
Pre-installation check ..   8%
Checking space requirements ..  16%
Installing/removing files ..  16%
Upgrading selected failsafe2_nfs.books subsystems
Installing/removing files ..  65%
Upgrading selected failsafe2_nfs.man subsystems
Installing/removing files ..  72%
Upgrading selected failsafe2_nfs.sw subsystems
Installing/removing files ..  94%
Running exit-commands ..  99%
Checking dependencies .. 100% Done.
Installations and removals were successful.
You may continue with installations or quit now.

Inst> quit
```

Creating the NFS Resource Type

You can use one of the following methods to create the `NFS` resource type:

- Run cluster manager (`cmgr`) and manually create the resource type. For more information, see the *IRIS FailSafe 2.0 Administrator's Guide*.
- Run cluster manager (`cmgr`) and install the resource type, as follows:

```
cmgr> show resource_types installed

NFS
template
Netscape_web
statd
Oracle_DB
MAC_address
IP_address
INFORMIX_DB
filesystem
volume

cmgr> install resource_type NFS in cluster eagan

cmgr>
```

- Use the template scripts supplied with IRIS FailSafe 2.0 located in `/var/cluster/cmgr-template/cmgr-create-resource_type`.
- Execute `/var/cluster/ha/resource_type/NFS/create_resource_type` and include the path of the CDB argument and the cluster name.
- Use the **Load Resource Type** GUI task to load the resource type.

Creating an NFS Resource

After you have defined the resource type, the administrator must define the NFS resources based on the resource type. Each resource requires a unique resource name (for example, the `NFS` resource type is the NFS instance name). Then, the administrator must supply the resource parameters. To create the resource, either use the cluster manager (`cmgr`), the `cmgr-create-resource-NFS/statd` scripts, or the GUI.

Example 2-2 Creating an NFS/statd Resource using cmgr

```
cm2> /usr/cluster/bincluster_mgr
Welcome to IRIS FailSafe Cluster Manager Command-Line Interface

cmgr> create resource /disk5 of resource_type NFS in cluster eagan
Enter commands, when finished enter either "done" or "cancel"

Type specific attributes to create with set command:

Type Specific Attributes - 1: export-info
Type Specific Attributes - 2: filesystem

Resource type dependencies to add:

Resource Dependency Type - 1: statd
Resource Dependency Type - 2: filesystem

resource /disk5 ? set export-info to "rw,wsync,anon=root"
resource /disk5 ? set filesystem to /disk5
resource /disk5 ? done
Successfully created resource /disk5
```

```
cmgr> modify resource /disk5 of resource_type NFS in cluster eagan
Enter commands, when finished enter either "done" or "cancel"

Type specific attributes to modify with set command:

Type Specific Attribute - 1: export-info
Type Specific Attribute - 2: filesystem

No resource type dependencies to add

resource /disk5 ? add dependency /disk5 of type filesystem
resource /disk5 ? add dependency /disk5/statmon of type statd
resource /disk5 ? done
Successfully modified resource /disk5

cmgr> delete resource /disk5/statmon of resource_type statd in cluster
eagan

cmgr> create resource /disk5/statmon of resource_type statd in cluster
eagan
Enter commands, when finished enter either "done" or "cancel"

Type specific attributes to create with set command:

Type Specific Attributes - 1: InterfaceAddress

Resource type dependencies to add:

Resource Dependency Type - 1: IP_address
Resource Dependency Type - 2: filesystem

resource /disk5/statmon ? set InterfaceAddress to 128.162.101.22
resource /disk5/statmon ? done
Successfully created resource /disk5/statmon

cmgr> modify resource /disk5/statmon of resource_type statd in cluster
eagan
Enter commands, when finished enter either "done" or "cancel"

Type specific attributes to modify with set command:

Type Specific Attribute - 1: InterfaceAddress

No resource type dependencies to add
```

```
resource /disk5/statmon ? add dependency 128.162.101.22 of type
IP_address
resource /disk5/statmon ? add dependency /disk5 of type filesystem
resource /disk5/statmon ? done
Successfully modified resource /disk5/statmon

cmgr> show resource /disk5 of resource_type NFS in cluster eagan

Too many arguments for command: show
Failed to show resource

show command failed
cmgr> show resource /disk5 of resource_type NFS

export-info: rw,wsync,anon=root
filesystem: /disk5

Resource dependencies
statd /disk5/statmon
filesystem /disk5

cmgr>
```

Creating an NFS Resource Group

To create a resource group, you must first become familiar with the terms and concepts of FailSafe. A resource group can be created either by the GUI or the cluster manager (cmgr).

To define an effective resource group, you must include all of the resources that the NFS resource is dependent on, such as file systems, volumes, and IP addresses. The following example shows the creation of a typical resource group:

```
cmgr> create resource_group nfs in cluster eagan
Enter commands, when finished enter either "done" or "cancel"

resource_group nfs ? set failover_policy to ordered-in-order
resource_group nfs ? add resource nfs8 of resource_type NFS
resource_group nfs ? add resource 128.162.101.20 of resource_type
IP_address
resource_group nfs ? done
Successfully created resource group nfs
```

```
cmgr> show resource_group nfs in cluster eagan

Resource Group: nfs
        Cluster: eagan
        Failover Policy: ordered-in-order

Resources:
        nfs8 (type: NFS)
        128.162.101.20 (type: IP_Addresses)
```

Testing the NFS Resource

To ensure that the NFS resource has been correctly configured, you can test individual actions by executing the scripts. Each script, located at /var/cluster/ha/resource_types/NFS, requires two arguments, an input file and an output file. The contents of these files are the resource names. The scripts will display 0 if they are successfully executed or display a positive number that indicates the error type. For more information on error codes, see the *IRIS FailSafe 2.0 Programmer's Guide*.

In the following example, you can test the start script by starting the NFS resource with the resource name nfs8.

```
$ cd /var/cluster/ha/resource_types/NFS
$ echo "nfs8" > /tmp/ipfile
$ ./start /tmp/ipfile /tmp/opfile
```

This should start the nfs8 instance, named by the NFS resource nfs8.

To view the individual script actions, you must edit the script and add “set -x” to the action function.

In the following example, to see what the `start_nfs()` function does, edit the `start` script and add “`set -x`” in the `start` function.

```
#

# Start the resource on the local machine.

# Return HA_SUCCESS if the resource has been successfully started on
# the local machine and HA_CMD_FAILED otherwise. The resource name is #
the nfs instance id.

#

start_nfs()

{

set -x

 # for all nfs resource configured

...
```

You are now ready to run the script. In this example, `/dev/null` is the name of the output file.

```
cm2> echo "/disk1" > /tmp/ipfile

cm2> /var/cluster/ha/resource_types/NFS/start /tmp/ipfile /dev/null

+ HA_SCRIPTNAME=/var/cluster/ha/resource_types/NFS/start

+ set_global_variables

+ set_local_variables

+ ha_check_args /tmp/ipfile /dev/null

+ [ 0 -ne 0 ]

+ ha_read_infile

+ start_nfs

+ NFSFILEDIR=/var/cluster/ha/tmp/disk1

+ HA_CMD=/sbin/mkdir -p /var/cluster/ha/tmp/disk1

+ ha_execute_cmd creating nfs status file directory

+ get_nfs_info /disk1

+ [ 0 -ne 0 ]

+ ha_get_field export-info rw,wsync,anon=root

filesystem /disk1 export-point
```

```
+ [ 0 -ne 0 ]
+ ha_get_field export-info rw,wsync,anon=root
filesystem /disk1 export-info
+ [ 0 -ne 0 ]
+ export_opts=rw,wsync,anon=root
+ retstat=0
+ /usr/etc/exportfs
+ grep /disk1$
+ 1> /dev/null 2>& 1
+ retstat=1
+ [ 1 -eq 1 ]
+ /usr/etc/exportfs
+ grep /disk1
+ grep rw,wsync,anon=root$
+ 1> /dev/null 2>& 1
+ retstat=1
+ [ 1 -eq 1 ]
+ HA_CMD=/sbin/grep /disk1 /etc/mtab > /dev/null 2>&1
+ ha_execute_cmd check if the export-point exists
+ [ 0 -eq 0 ]
+ HA_CMD=/usr/etc/exportfs -i -o rw,wsync,anon=root /disk1
+ ha_execute_cmd export /disk1 directories to NFS clients
+ [ 0 -ne 0 ]
+ ha_write_status_for_resource /disk1 0
+ exit_script 0
```

Testing the start Script

You can use the following procedure to test the `start` script.

1. To test the `start` script, you must ensure that the file system has not already been exported. Use the following command to check if the file system has been exported. If it has already been exported, you must first unexport the file system by using the `exportfs`(1M) command, as in the following example:

   ```
   cm2> exportfs
   /dmf_home
   /move_fs
   /usr/cluster/bin
   /disk2
   disk1 -rw,wsync,anon=root
   cm2> exportfs -u /disk1
   ```

   As this example shows, `/disk1` must first be unexported before you can start the NFS resource `/disk1`.

2. To start the NFS resource, enter the following command:

   ```
   # echo "resource-name" > /tmp/ipfile
   # /var/cluster/ha/resource_types/NFS/start /tmp/ipfile /dev/null
   ```

3. You can check to see if the resource is online by looking for the exported file system list by executing the following command:

   ```
   # exportfs
   ```

Testing the stop Script

You can use the following procedure to test the `stop` script.

1. To test the `stop` script, you must ensure that the NFS file system is already exported. Use the following command to check if the file system has been exported. If it has not already been exported, you must first export the file system by using the `exportfs`(1M) command, as in the following example:

   ```
   cm2> exportfs
   /dmf_home
   /move_fs
   /usr/cluster/bin
   /disk2
   disk1 -rw,wsync,anon=root
   cm2>
   ```

   As this example shows, `/disk1` has already been exported.

2. To stop the `NFS` resource, enter the following command:

   ```
   # echo "resource-name" > /tmp/ipfile
   # /var/cluster/ha/resource_types/NFS/stop /tmp/ipfile /dev/null
   ```

3. Check to see if the resource is offline by looking for the exported file system list by using the following command:

   ```
   # exportfs
   ```

Testing the monitor Script

You can use the following procedure to test the `monitor` script.

1. To monitor the `NFS` resource, enter the following command:

   ```
   # echo "resource-name" > /tmp/ipfile
   # /var/cluster/ha/resource_types/NFS/monitor /tmp/ipfile /dev/null
   ```

2. As with all scripts, the output is logged to the script file located at `/var/cluster/ha/log/script_`*hostname*. To check that the local monitoring script has worked, use the `tail`(1) command and execute the script as in the following example:

```
cm2> /var/cluster/ha/resource_types/NFS/monitor /tmp/ipfile
/dev/null

cm2> echo $?

2

cm2> tail /var/cluster/ha/log/script_cm2

Thu Oct 8 13:17:00.952 <ha_cilog script 1369611:0>
/var/cluster/ha/resource_types/NFS/monitor[1365793]:
/var/cluster/ha/resource_types/NFS/monitor called with /tmp/ipfile
and /dev/null

Thu Oct 8 13:17:02.226 <ha_cilog script 1368658:0>
/var/cluster/ha/resource_types/NFS/monitor[1365793]: check to see if
/disk1 is mounted on /disk1

Thu Oct 8 13:17:02.344 <ha_cilog script 1368597:0>
/var/cluster/ha/resource_types/NFS/monitor[1365793]: /sbin/mount |
grep /disk1 | grep /disk1 >> /dev/null 2>&1 exited with status 0

Thu Oct 8 13:17:02.393 <ha_cilog script 1364700:0>
/var/cluster/ha/resource_types/NFS/monitor[1365793]: stat mount
point /disk1

Thu Oct 8 13:17:02.460 <ha_cilog script 1368798:0>
/var/cluster/ha/resource_types/NFS/monitor[1365793]: /sbin/stat
/disk1 exited with status 0

Thu Oct 8 13:17:02.514 <ha_cilog script 1371071:0>
/var/cluster/ha/resource_types/NFS/monitor[1365793]: check the
filesystem /disk1 is exported

Thu Oct 8 13:17:02.769 <ha_cilog script 1363634:0>
/var/cluster/ha/resource_types/NFS/monitor[1365793]:

Thu Oct 8 13:17:02.789 <ha_cilog script 1362780:0>
/var/cluster/ha/resource_types/NFS/monitor[1365793]: awk ‘{print
$1}’ /var/cluster/ha/tmp/exportfs.1365793 | grep /disk1 exited with
status 1

Thu Oct 8 13:17:02.828 <ha_cilog script 1371194:0>
/var/cluster/ha/resource_types/NFS/monitor[1365793]: failed to find
/disk1 in exported filesystem list:-

Thu Oct 8 13:17:02.864 <ha_cilog script 1373463:0>
/var/cluster/ha/resource_types/NFS/monitor[1365793]: /dmf_home
/move_fs /usr/cluster/bin /disk2
```

Testing the restart Script

You can use the following procedure to test the `restart` script.

1. To restart the `NFS` resource, enter the following command

   ```
   # echo "resource-name" > /tmp/ipfile
   # /var/cluster/ha/resource_types/NFS/restart /tmp/ipfile /dev/null
   ```

Testing the exclusive and probe Scripts

You can use the following procedure to test the `exclusive` and `probe` scripts.

1. To test the `exclusive` and `probe` scripts, enter one of the following sets of commands. For the `exclusive` script, enter the following command:

   ```
   # echo "resource-name" > /tmp/ipfile
   # /var/cluster/ha/resource_types/NFS/exclusive /tmp/ipfile /dev/null
   ```

   For the `probe` script, enter the following command:

   ```
   # echo "resource-name" > /tmp/ipfile
   # /var/cluster/ha/resource_types/NFS/probe /tmp/ipfile /dev/null
   ```

2. As with all scripts, the output is logged to the `/var/cluster/ha/log/script_`*hostname*. To check that the script has worked, use the `tail`(1) command and execute the script as in the following example:

   ```
   cm2> /var/cluster/ha/resource_types/NFS/probe /tmp/ipfile /dev/null
   cm2> tail /var/cluster/ha/log/script_cm2
   Thu Oct 8 13:23:31.076 <ha_cilog script 1368355:0>
   /var/cluster/ha/resource_types/NFS/probe[1362486]:
   /var/cluster/ha/resource_types/NFS/probe called with /tmp/ipfile and
   /dev/null

   Thu Oct 8 13:23:31.273 <ha_cilog script 1372343:0>
   /var/cluster/ha/resource_types/NFS/probe[1362486]: checking for nsfd
   processes

   Thu Oct 8 13:23:31.351 <ha_cilog script 1369024:0>
   /var/cluster/ha/resource_types/NFS/probe[1362486]: /sbin/killall -0
   nfsd exited with status 0
   ```

Testing Resource Group Failovers

You can test the failover policy by using either `cmgr` or the FailSafe GUI to move the resource group to another node in the cluster. To ensure that the resource group correctly failed over, use the `cmgr` or GUI to display the resource group states. The following example uses `cmgr` to test the failover policy:

```
cmgr> admin offline resource_group NFS in cluster eagan
Resource group is changing state -- please wait


cmgr> admin move resource_group NFS in cluster eagan to node cm2
Resource group is changing state -- please wait


cmgr> admin online resource_group NFS in cluster eagan
Resource group is changing state -- please wait
```