



**SIOS Protection Suite for Linux
NFS Recovery Kit
v9.1.2**

Administration Guide

June 2017

This document and the information herein is the property of SIOS Technology Corp. (previously known as SteelEye® Technology, Inc.) and all unauthorized use and reproduction is prohibited. SIOS Technology Corp. makes no warranties with respect to the contents of this document and reserves the right to revise this publication and make changes to the products described herein without prior notification. It is the policy of SIOS Technology Corp. to improve products as new technology, components and software become available. SIOS Technology Corp., therefore, reserves the right to change specifications without prior notice.

LifeKeeper, SteelEye and SteelEye DataKeeper are registered trademarks of SIOS Technology Corp.

Other brand and product names used herein are for identification purposes only and may be trademarks of their respective companies.

To maintain the quality of our publications, we welcome your comments on the accuracy, clarity, organization, and value of this document.

Address correspondence to:
ip@us.sios.com

Copyright © 2017
By SIOS Technology Corp.
San Mateo, CA U.S.A.
All rights reserved

Table of Contents

Chapter 1: Introduction	1
NFS Server Recovery Kit Documentation	1
SIOS Protection Suite Documentation	1
Reference Documents	1
Overview	1
LifeKeeper for Linux NFS Server Recovery Kit Overview	1
Requirements	2
Chapter 2: Configuration Considerations	4
Specific Configuration Considerations	4
Configuring NFS Server with LifeKeeper	5
NFS	5
Export Considerations	6
NFS File Lock Failover	7
RPC.MOUNTD Restart	7
NFS Resource Hierarchy	7
Set up for Automatic Startup of the Service	8
Set up for Automatic Startup for rpcbind/portmap	8
Handling of NFSv2/v3 and NFSv4	8
Examples	8
Configuration Requirements	9
Examples	9
Related	9
Active - Active - NFS v2-v3	9
Active/Active Configuration Example for NFS v2/v3	10
Configuration Notes:	10

Creating the First Resource Hierarchy on Server 1:	10
Extending the First Resource Hierarchy to Server 2:	11
Creating the Second Resource Hierarchy on Server 2:	11
Extending the Second Resource Hierarchy to Server 1:	11
Active - Standby - NFS v2-v3	11
Active/Standby Configuration Example for NFS v2/v3	12
Configuration Notes:	12
Creating a Resource Hierarchy to Server 1:	12
Extending a Resource Hierarchy to Server 2:	13
Active - Standby - NFS v4	13
Active/Standby Configuration Example for NFS v4	13
Creating a Resource Hierarchy to Server 1:	14
Extending a Resource Hierarchy to Server 2:	14
Chapter 3: Configuration Tasks	15
Creating an NFS Resource Hierarchy	15
Deleting a Resource Hierarchy	18
Extending Your Hierarchy	20
Testing Your Resource Hierarchy	23
Validating the Client Setup	24
Performing a Manual Switchover from the GUI	24
Recovery Operations	24
Unextending Your Hierarchy	24
Chapter 4: Troubleshooting	27
HA nfs-utils Installation and Configuration	27
Hierarchy Delete Messages and Errors	28
Hierarchy Restore, Remove and Recover Messages	28
Bringing an NFS Resource In-Service (Restore)	28
Taking an NFS Resource Out of Service (Remove)	30
Bringing an NFS Resource Back In Service (Recover)	30
NFS Extend Hierarchy Errors	30

NFS Hierarchy Creation Errors	31
--	-----------

Chapter 1: Introduction

NFS Server Recovery Kit Documentation

The SIOS Protection Suite for Linux NFS Server Recovery Kit provides fault resilience for Network File System (NFS) software in a SIOS Protection Suite environment. This enables a failure on the primary NFS server to be recovered on a designated backup server without significant lost time or human intervention.

[SIOS Protection Suite for Linux NFS Server Recovery Kit Overview](#)

SIOS Protection Suite Documentation

The following is a list of SIOS Protection Suite for Linux related information available from SIOS Technology Corp.:

- SPS for Linux Technical Documentation
- SPS for Linux Release Notes
- [SIOS Technology Corp. Documentation](#)

Reference Documents

The following is a list of reference documents associated with the SIOS Protection Suite NFS Server Recovery Kit:

- [NFS Online documentation](#)
- *Managing NFS and NIS, Hal Stern, O'Reilly & Associates, Inc. 1991*

Overview

LifeKeeper for Linux NFS Server Recovery Kit Overview

The NFS Server Recovery Kit provides a High Availability NFS service in hierarchical cooperation with the Filesystem Recovery Kit (provided as part of the steeleye-lk package) and the IP Recovery Kit (steeleye-lkIP).

The kit ensures that an IP resource and a file system resource containing the shared mount point are always in-service on the same server in the cluster. Clients who mount the file system using the LifeKeeper-protected IP resource can continue processing files on the volume virtually uninterrupted while the actual export service is switched between servers in the cluster (either manually or in response to a failure). Client recovery times will depend on the interaction between the client and the NFS server. For example, with NFSv3, the protocol timeouts for TCP are longer than that of UDP. In order to determine the best transport layer protocol to use

Requirements

with NFS, consider the recommendations of the OS vendor, the advantages and disadvantages of each transport protocol and your specific environment.

Note: TCP is strongly recommended with NFSv4 by most OS vendors and the NFS ARK has been validated with TCP and NFSv4.

Beginning with Version 7.4 of the NFS Server Recovery Kit, an NFS v4 **pseudo file system** export is now supported providing clients with seamless access to all exported objects on the server. Prior to Version 7.4, clients were forced to mount each shared server file system for access. With NFS Version 4, the server still specifies export controls for each server directory or file system to be exported for NFS access, and from these export controls, the server renders a **single directory tree** of all the exported data filling in gaps between the exported directories. This tree is known as a **pseudo file system**, and it starts at the NFS Version 4 server's pseudo root. This pseudo file system model allows an NFS v4 client, depending on its implementation, to perform a single mount of the server's pseudo root in order to access all the server's exported data.

All files on the file system become temporarily unavailable while a switchover or failover is in progress, but they become available again transparently when the resource transfer is complete. For a switchover, this can take between 5 and 30 seconds. For a failover, the recovery time depends on how long it takes to repair the file system. It is strongly recommended that you format the underlying disk volume with a Journaling File System (JFS) which is extremely robust to failure and can be repaired in a few seconds.

You may also choose to use a Linux file system (ext2) as the underlying file system, but in that case, failover times will be much longer.

Requirements

Before installing and configuring the LifeKeeper NFS Server Recovery Kit, be sure that your configuration meets the following requirements:

- **Servers.** The Recovery Kit requires two or more servers configured in accordance with the LifeKeeper requirements described in the SPS for Linux Technical Documentation and the SPS for Linux Release Notes. See the Release Notes for supported Linux distributions.
- **LifeKeeper software.** You must install the same version of LifeKeeper software and any patches on each server. Please refer to the SPS for Linux Technical Documentation and the SPS for Linux Release Notes for specific LifeKeeper requirements.
- **LifeKeeper IP Recovery Kit.** You must have the same version of the LifeKeeper IP Recovery Kit on each server.
- **IP network interface.** Each server requires at least one Ethernet TCP/IP-supported network interface. In order for IP switchover to work properly, user systems connected to the local network should conform to standard TCP/IP specifications.

Note: Even though each server requires only a single network interface, you should use multiple interfaces for a number of reasons: heterogeneous media requirements, throughput requirements, elimination of single points of failure, network segmentation and so forth.

- **TCP/IP software.** Each server also requires the TCP/IP software.
- **nfs-utils software.** Each server must have a high availability enabled version of the nfs-utils package

Requirements

installed and configured prior to configuring LifeKeeper and the LifeKeeper NFS Server Recovery Kit (on some OS distributions, `nfs-utils` is provided via another package). The same version must be installed on each server.

The LifeKeeper Installation Support setup script will configure `nfs-utils` for use in an HA environment. Additionally, when using the kit to protect **NFS v2/v3 exports**, the init scripts NFS lock (`/etc/init.d/nfslock`) and NFS server (`/etc/init.d/nfs` or `/etc/init.d/nfsserver` depending on the Linux distribution) should be configured to start automatically on system boot. When protecting **NFS v4** root exports, they should not be configured to automatically start at boot.

- Refer to the SPS for Linux Installation Guide for specific instructions on how to install or remove the LifeKeeper NFS Server Recovery Kit.

Chapter 2: Configuration Considerations

These following sections contain information to be considered before starting to configure and administer the NFS Server Recovery Kit as well as examples of typical LifeKeeper NFS configurations.

Please refer to SPS for Linux Technical Documentation for instructions on configuring your LifeKeeper Core resource hierarchies.

- [Configuring NFS Server with LifeKeeper](#)
- [Specific Configuration Considerations](#)
- [Configuration Examples](#)

Specific Configuration Considerations

The following should be considered before using the LifeKeeper NFS Server Recovery Kit:

- The NFS file system to be placed under LifeKeeper protection must be exported by the primary server (the server where the NFS resource is being created). This implies that NFS is running and the underlying file system is mounted.

Note: If the */home* directory is shared via NFS, then */home* is the underlying file system.

- **When you export a read/write file system, use the *sync* option.** This option requests that all file system writes be committed to disk before the write request completes. **If the *sync* option is not used with an NFS file system under LifeKeeper protection, data may be lost during a failover.**
- The underlying file system must be on a shared device and mounted with write permission.
- If the underlying file system is already protected by LifeKeeper, it must be in service on the primary server and have the highest priority. If the underlying file system is not under LifeKeeper protection, then the Recovery Kit will place it under protection.

- The **NFSv4** root and any bind mounts must be exported by the primary server (the server where the NFS resource is being created). This implies NFS is running and the underlying file systems (root and bind mounts) are mounted. The root and bind mounts must be on shared devices that LifeKeeper can protect. If the underlying file systems are already protected by LifeKeeper, they must be in service on the primary server and have the highest priority. If the file systems are not already protected, then the recovery kit will place them under protection.

Note: If the NFSv4 root is */export* with three subdirectories called *bin*, *opt* and *other*, local file systems */usr/local/bin* and */usr/local/opt* are bind mounted into the NFSv4 pseudo file system, then */export*, */usr/local/bin* and */usr/local/opt* must be mounted on a shared device with */export* mounted with write permission, */usr/local/bin* and */usr/local/opt* must be bind mounted into */export* (`mount --bind /usr/local/bin /export/bin` and `mount --bind /usr/local/opt /export/bin`) and

the `/export`, `export/bin` and `/export/opt` must all be exported. See the configuration example [Active/Standby for NFSv4](#) for further information.

- The NFS Server Recovery Kit requires an IP resource that must be created and in service on the primary server. The IP resource must also have its highest priority on the primary server.
- Before creation of the NFS resource, clients must be able to mount the NFS file system using the LifeKeeper-protected IP address.
- When you extend an NFS file system resource, the file system must mount at the same mount point on each server.
- When protecting NFSv4 root exports, `/var/lib/nfs` is moved to the NFSv4 root file system which must have write permissions set when mounted. To provide continued access, a symbolic link is created from `/var/lib/nfs` to the NFSv4 root. Because of this, Active/Active NFSv4 configurations are not supported nor are configurations with NFSv2/v3 and NFSv4. NFSv2/v3 exports should use the bind features of NFSv4 instead of attempting to make them standalone NFSv2/v3 exports.
- The Oracle Recovery Kit supports NFSv3 for shared database storage. NFSv4 is not supported by the Oracle Recovery Kit at this time due to NFSv4 file locking mechanisms.
- When using a system that adopts Linux kernel 3.12 or later as an NFS client, the file lock at the client side is lost when the communication is disconnected depending on specifications of the kernel. Therefore, takeover of the lock is not guaranteed during switchover or failover.
- In NFS v4, the file lock is lost when the client cannot communicate with the server for more than a certain period of time. This period of time is set as the variable `NFS_V4_LEASE_TIME` and the default value is 10 (seconds). When the communication between the client and the server is disconnected for a longer time than the period set in `NFS_V4_LEASE_TIME` during a switchover or failover, the file lock set by the client is forcibly disabled. In order to change this value, set an appropriate value of the environment variable `NFS_V4_LEASE_TIME` in `/etc/default/LifeKeeper`. The environment variable `NFS_V4_GRACE_TIME` should also be set as the same value as `NFS_V4_LEASE_TIME`.
If you increase the value of `NFS_V4_LEASE_TIME`, the above mentioned problem will less likely to occur but it will take time to reconnect to the client after the server switching.

Configuring NFS Server with LifeKeeper

This section contains information to consider before starting to configure and administer the NFS Server Recovery Kit as well as examples of typical LifeKeeper NFS configurations.

Please refer to SPS for Linux Technical Documentation for instructions on configuring your LifeKeeper Core resource hierarchies.

NFS

The following table describes the NFS files, commands and daemons that are important to the NFS Server Recovery Kit:

Export Considerations

NFS Component	Description
<code>exports(5)</code> (<code>/etc/exports</code>)	<p>Access control list for file systems exported to NFS clients. Each line of the file contains an export point, an optional list of clients that can mount the file system and an optional list of mount parameters.</p> <p>Note: When you create a LifeKeeper-protected NFS resource, the export information for the file system is removed from the <code>exports</code> file and maintained under LifeKeeper. If you delete the NFS resource, the export information is restored to the <code>exports</code> file.</p>
<code>/var/lib/nfs</code>	<p>Directory that contains NFS information on current exports, client mounts, locking status and more. In Version 7.4 and later, this directory is moved to the NFS export directory when protecting an NFS v4 pseudo file system. <code>/var/lib/nfs</code> is replaced with a symbolic link to the new location on both the primary and standby systems.</p>
<code>/var/lib/nfs/etab</code>	<p>File that contains the current table of exported file systems for NFS. This file is maintained by the <code>exportfs</code> command; the user does not edit the file directly.</p> <p>Note: When you bring an NFS resource into service on a backup server, the NFS file system is removed from the <code>etab</code> file on the primary server and inserted into the <code>etab</code> file on the backup server.</p>
<code>/var/lib/nfs/rpc_pipefs</code>	<p>Used for kernel to userspace communication for NFS. This directory is relocated to <code>/var/lib</code> during installation of LifeKeeper.</p>
<code>exportfs(8)</code> (<code>/usr/sbin/exportfs</code>)	<p>Command used to maintain the table of exported file systems in <code>/var/lib/nfs/etab</code>.</p>
<code>rpc.mountd(8)</code> (<code>/usr/sbin/rpc.mountd</code>)	<p>Daemon that authenticates a mount request and returns a filehandle if the client is permitted to mount the file system.</p>
<code>rpc.nfsd(8)</code> (<code>/usr/sbin/rpc.nfsd</code>)	<p>Daemon that handles client file system requests.</p>
<code>rpc.quotad(8)</code> (<code>/usr/sbin/rpc.rquotad</code>)	<p>The rpc server that returns quotas for a user of a local file system which is mounted remotely over NFS.</p>
<code>rpc.lockd(8)</code> (<code>/sbin/rpc.lockd</code>)	<p>Daemon that handles client file lock requests.</p>
<code>rpc.statd(8)</code> (<code>/usr/sbin/rpc.statd</code>)	<p>Daemon that monitors the status of and makes status notifications for NFS clients and servers. This daemon must be running in order for NFS file locking to work properly.</p>
<code>portmap/rpcbind</code>	<p>Daemon process that converts RPC program numbers into port numbers and must be running for NFS. A failure of this process will force a switchover to a standby node. On some systems, this function is provided by <code>portmap</code> while on others this function is provided by <code>rpcbind</code>.</p>
<code>rpc.idmapd</code>	<p>NFS v4 ID to name mapper daemon process for translating user and group IDs to names and names to user and group IDs. This process must be running for NFS v4 but is not required for NFS v2/v3.</p>

Export Considerations

LifeKeeper protection for a given exported file system depends on the export options being exactly of the form as described in the `exports(5)` man page. In particular, pay attention to the host restriction format. There are only four legal host restrictions: (single host, netgroup, wildcard host `*name*` and netmask).

In particular, a wildcard IP address (like `172.13.4.*`) is not legal and will lead to potential stale filehandles on switchover or failover. Check very carefully by executing `exportfs -v` and manually comparing the returned export description against the format described in the man page (unfortunately, `exportfs` doesn't check for you and will accept certain illegal export formats).

NFS File Lock Failover

The NFS Server Recovery Kit can fail over file locks taken by NFS clients when protecting NFS v2/v3 exports. With the enhanced locking features available in the NFS subsystem for a v4 export, this setting is not required as lock failover will occur automatically. In order to enable file lock failover for v2/v3 exports, the following entry in `/etc/default/LifeKeeper` must be modified:

FAILOVERNFSLOCKS=false

This should be changed to:

FAILOVERNFSLOCKS=true

Note: Currently, the recovery kit cannot protect file locks if there is more than one virtual IP address being used for NFS client connections. (This limitation stems from the `statd` daemon's inability to send status notifications out from more than one IP address. The `statd` daemon is responsible for sending notifications to all NFS clients of a given server, and therefore must be able to send out status notifications to clients on all IP addresses.) This limitation would include an Active/Active configuration. Additionally, in the 2.6 kernel on SUSE Linux Enterprise Server 10, the `rpc.statd` function runs internal to the kernel and not as a user process, thus LifeKeeper is unable to provide lock failover on this Linux distribution at this time.

RPC.MOUNTD Restart

Under certain conditions with multiple NFS resource hierarchies, `rpc.mountd` fails to properly advertise the list of exports available. As such, the NFS Recovery Kit on a restore will stop and restart `rpc.mount` to ensure the proper list of exports is available to all clients. This action of stopping and restarting `rpc.mount` is controlled via the `RESTARTMOUNTD` entry in `/etc/default/LifeKeeper`. By default, this entry is set to `true` to cause the stop and restart of `rpc.mount` on all NFS restores:

RESTARTMOUNT=true

To turn off this action set:

RESTARTMOUNT=false

NFS Resource Hierarchy

When you create a LifeKeeper protected NFS resource, LifeKeeper creates the following hierarchy:

- NFS file system resource (parent or root)
- HA-NFS resource

Set up for Automatic Startup of the Service

- File system resource (the underlying file system)
- For **NFS v4** hierarchies with bind mounts, a file system resource will be created for each bind mount.

Create the IP address resource before creating the NFS resource.

You have the option of creating the file system resource(s) before creating the NFS resource. If you do this, you can choose the name assigned to the file system resource(s). If not, the NFS Server Recovery Kit automatically creates the file system resource(s) when creating the NFS resource.

Set up for Automatic Startup of the Service

Set up `nfs-server.service` for automatic startup via the following commands when protecting NFSv2/NFSv3 exports in RHEL7, CentOS7, OEL7 and SLES12 environments (i.e. systemd environments).

```
# systemctl start nfs-server.service
# systemctl enable nfs-server.service
```

Note: Do not set up `nfs-server.service` for automatic startup when protecting NFSv4 exports.

Set up for Automatic Startup for rpcbind/portmap

Make sure that automatic startup for `rpcbind` or `portmap` is enabled.

- For systemd environment such as RHEL7, CentOS7, OEL7 and SLES12

```
# systemctl is-enabled rpcbind.socket
```

- For environment where systemd is not used

```
# chkconfig --list rpcbind
```

If the automatic startup of `rpcbind` or `portmap` is not active, execute following command to activate it:

- For systemd environment such as RHEL7, CentOS7, OEL7 and SLES12

```
# systemctl start rpcbind.socket
# systemctl enable rpcbind.socket
```

- For environment where systemd is not used

```
# service rpcbind start
# chkconfig rpcbind on
```

Handling of NFSv2/v3 and NFSv4

LifeKeeper protects the area as NFSv4 for which option “`fsid=0`” is specified. The area for which “`fsid=0`” is not specified is protected as NFSv2/3. Please note that the areas protected as NFSv2/3 and as NFSv4 cannot be mixed.

Examples

The examples in this section show how NFS instances can be configured on shared (or replicated) disks.

Each diagram shows the relationship between the type of configuration and the NFS parameters. Each configuration also adheres to the configuration rules and requirements described in this section that ensure compatibility between the NFS configuration and the LifeKeeper software.

This section first describes the configuration requirements and then provides the configuration examples **Active/Standby NFS v2/v3**, **Active/Active NFS v2/v3** and **Active/Standby NFS v4**.

Note: Active/Active for NFS v4 is not supported nor is any combination of NFS v2/v3 and NFS v4.

The examples in this section are only a sample of the configurations that can be established, but understanding these configurations and adhering to the configuration rules will help define and set up workable solutions for your computing environment.

Configuration Requirements

NFS Tag names are arbitrary names that describe protected file systems to LifeKeeper. The default tag name suggested by LifeKeeper is “*nfs-<export point>*.”

To understand the configuration examples, keep in mind that the underlying file system (or file systems if NFS v4 and bind mounts are used) must always be on shared or replicated disks. The file system(s) must be mountable from each of the paired systems.

Examples

- [Active/Standby Configuration for NFS v2/v3](#)
- [Active/Active Configuration for NFS v2/v3](#)
- [Active/Standby Configuration for NFS v4](#)

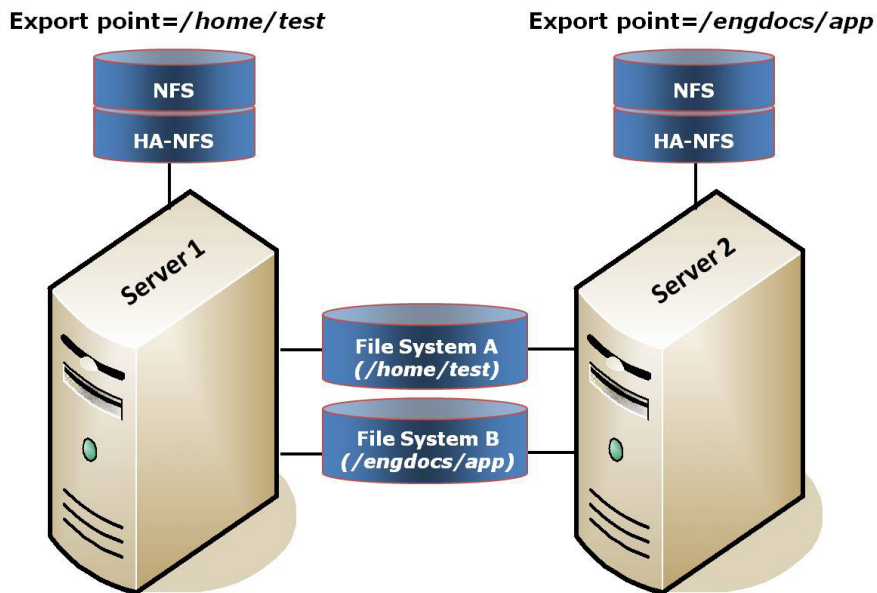
Related

- [Configuration Considerations](#)

Active - Active - NFS v2-v3

An example of **Active/Active for NFS v2/v3** consists of two or more systems actively running NFS and exporting file systems.

Active/Active Configuration Example for NFS v2/v3



Configuration Notes:

- The NFS software must be installed on both servers.
- Initially, *Server 1* exports */home/test* and *Server 2* exports */engdocs/app*. In a switchover situation, one system can export both file systems.
- File System A is the underlying file system for export point */home/test*. File System B is the underlying file system for export point */engdocs/app*.
- The underlying file systems are on different shared disks.

Creating the First Resource Hierarchy on *Server 1*:

Server:	<i>Server1</i>
Export Point:	<i>/home/test</i>
IP Tag:	<i>ip-172.17.100.202</i>
NFS Tag:	<i>nfs-/home/test</i>

Extending the First Resource Hierarchy to Server 2:

Template Server:	<i>Server1</i>
Tag to Extend:	<i>nfs-/home/test</i>
Target Server:	<i>Server2</i>
Target Priority:	10

Creating the Second Resource Hierarchy on Server 2:

Server:	<i>Server2</i>
Export Point:	<i>/engdocs/app</i>
IP Tag:	<i>ip-172.17.100.203</i>
NFS Tag:	<i>nfs-/engdocs/app</i>

Extending the Second Resource Hierarchy to Server 1:

Template Server:	<i>Server2</i>
Tag to Extend:	<i>nfs-/engdocs/app</i>
Target Server:	<i>Server1</i>
Target Priority:	10

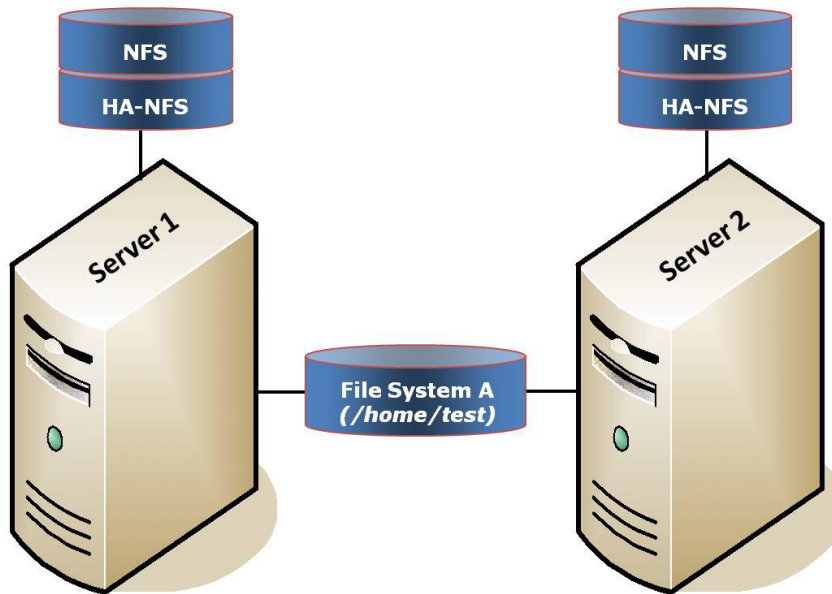
Active - Standby - NFS v2-v3

This section provides an example of **Active/Standby for NFS v2/v3**. In this configuration, *Server 1* is considered active because it is running NFS and exporting the file system. *Server 2* does other processing. If *Server 1* fails, *Server 2* gains access to the file system and uses the LifeKeeper secondary hierarchy to make it available to clients.

Note: In an active/standby configuration, *Server 2* might be running NFS but does not have any other NFS resources under LifeKeeper protection.

Active/Standby Configuration Example for NFS v2/v3

Export point=`/home/test`



Configuration Notes:

- The NFS software must be installed on both servers.
- The underlying file system (*File System A*) must be on a shared (or replicated) disk.
- The NFS export point is `/home/test`.
- The exported file system must have the same mount point on both the primary and backup servers.
- *Server 2* cannot access files and directories on the shared disk while *Server 1* is active.

Creating a Resource Hierarchy to Server 1:

Server:	<i>Server1</i>
Export Point:	<i>/home/test</i>
IP Tag:	<i>ip-172.17.100.202</i>
NFS Tag:	<i>nfs-/home/test</i>

Extending a Resource Hierarchy to Server 2:

Template Server:	<i>Server1</i>
Tag to Extend:	<i>nfs-/home/test</i>
Target Server:	<i>Server2</i>
Target Priority:	10

Active - Standby - NFS v4

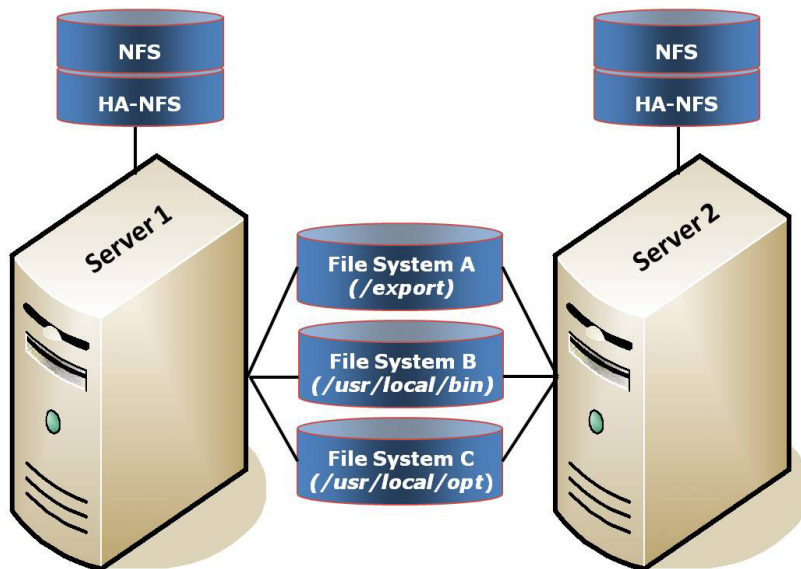
This section provides an example of **Active/Standby for NFS v4**. In this configuration, *Server 1* is considered active because it is running NFS and exporting the file systems. *Server 2* does other processing. If *Server 1* fails, *Server 2* gains access to the file systems and uses the LifeKeeper secondary hierarchy to make it available to clients.

Note: In an active/standby configuration, *Server 2* might be running NFS but does not have any other NFS resources under LifeKeeper protection.

Active/Standby Configuration Example for NFS v4

Export points=
/export, /export/bin and /export/opt

Bind mounts =
/usr/local/bin to /export/bin and /usr/local/opt to /export/opt



Configuration Notes:

Creating a Resource Hierarchy to Server 1:

- The NFS software must be installed on both servers.
- The underlying file systems (*File Systems A, B and C*) must be mounted and be on shared (or replicated) disks.
- Bind mounts must be completed prior to creation (bind */usr/local/bin* to */export/bin* and bind */usr/local/opt* to */export/opt*).
- The NFS export points are */export*, */export/bin* and */export/opt*.
- The exported file systems must have the same mount point on both the primary and backup servers.
- *Server 2* cannot access files and directories on the shared disk while *Server 1* is active.

Creating a Resource Hierarchy to Server 1:

Server:	<i>Server1</i>
Export Point:	<i>/export</i>
IP Tag:	<i>ip-172.17.100.202</i>
NFS Tag:	<i>nfs-/export</i>

Note: With the selection of the export points, bind information is picked up automatically; therefore, no input is required to create and extend the hierarchy.

Extending a Resource Hierarchy to Server 2:

Template Server:	<i>Server1</i>
Tag to Extend:	<i>nfs-/export</i>
Target Server:	<i>Server2</i>
Target Priority:	10

Chapter 3: Configuration Tasks

The following configuration tasks can be performed from the LifeKeeper GUI. These four tasks are described in this section as they are unique to an NFS Server resource instance and different for each Recovery Kit.

Note: Throughout this section, we explain how to configure your Recovery Kit by selecting certain tasks from the **Edit** menu of the LifeKeeper GUI. You can also select each configuration task from the toolbar. You may also right-click on a global resource in the **Resource Hierarchy Tree** (left-hand pane) of the status display window to display the same drop-down menu choices as the **Edit** menu. This, of course, is only an option when a hierarchy already exists.

You can also right-click on a resource instance in the **Resource Hierarchy Table** (right-hand pane) of the status display window to perform all the configuration tasks, except **Creating a Resource Hierarchy**, depending on the state of the server and the particular resource.

- [Create a Resource Hierarchy](#). Creates an application resource hierarchy in your LifeKeeper cluster.
- [Delete a Resource Hierarchy](#). Deletes a resource hierarchy from all servers in your LifeKeeper cluster.
- [Extend a Resource Hierarchy](#). Extends a resource hierarchy from the primary server to a backup server.
- [Unextend a Resource Hierarchy](#). Unextends (removes) a resource hierarchy from a single server in the LifeKeeper cluster.
- [Test Your Resource Hierarchy](#). Tests your NFS resource hierarchy by initiating a manual switchover.

The following tasks are described in the Administration section within the SPS for Linux Technical Documentation because they are common tasks with steps that are identical across all Recovery Kits.

- **Create a Resource Dependency**. Creates a parent/child dependency between an existing resource hierarchy and another resource instance and propagates the dependency changes to all applicable servers in the cluster.
- **Delete a Resource Dependency**. Deletes a resource dependency and propagates the dependency changes to all applicable servers in the cluster.
- **In Service**. Brings a resource hierarchy into service on a specific server.
- **Out of Service**. Takes a resource hierarchy out of service on a specific server.
- **View/Edit Properties**. View or edit the properties of a resource hierarchy on a specific server.

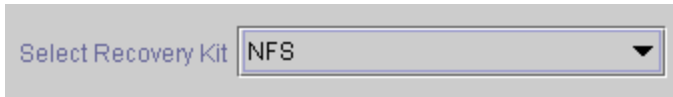
Creating an NFS Resource Hierarchy

To create a resource instance from the primary server, you should complete the following steps:

1. From the LifeKeeper GUI menu, select **Edit**, then **Server**. From the drop-down menu, select **Create Resource Hierarchy**.

To change a selection already entered or if an error message is encountered during any step in the creation of your NFS resource hierarchy, use the **Back** button to change your selection or make corrections (assuming the **Back** button is enabled).

A dialog box will appear with a drop-down menu listing all recognized Recovery Kits installed within the cluster. Select **NFS** from the drop-down menu.

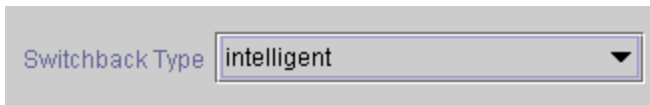


Select Recovery Kit

Click **Next** to proceed to the next dialog box.

Note: If you click the **Cancel** button at any time during the sequence of creating your hierarchy, LifeKeeper will cancel the entire creation process.

2. Select the **Switchback Type**. This dictates how the NFS instance will be switched back to the primary server when it comes back into service after a failover to the backup server. Choose either **Intelligent** or **Automatic**. **Intelligent switchback** requires administrative intervention to switch the instance back to the primary/original server. **Automatic switchback** means the switchback will occur as soon as the primary server comes back on line and is under LifeKeeper protection.



Switchback Type

The switchback type can be changed later, if desired, from the **General** tab of the **Resource Properties** dialog box.

Click **Next** to proceed to the next dialog box.

3. Select the **Server** where you want to create the NFS resource (typically this is referred to as the primary or template server). All the servers in your cluster are included in the drop-down menu.



Server

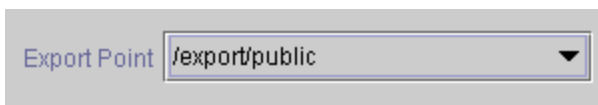
Click **Next** to proceed to the next dialog box.

4. The **Export Point** dialog displays a drop-down list of export points for NFS file systems that meet the following criteria:

Creating an NFS Resource Hierarchy

- The export point has been exported by NFS.
- The export point is on a shared drive.
- If the underlying file system is LifeKeeper-protected, it must be in service and have the highest priority on the server selected on the **Server** dialog.
- NFSv4 criteria:
 - For NFS v4 root export with bind mounts, bind mounts must be on a shared drive just like the export, and if the file system is LifeKeeper-protected, it must be in service and have the highest priority on the server selected on the Server dialog.
 - If an NFS v4 root export is already being protected, no choices will be provided (there should only be one v4 and a mixture of V2/v3 with v4 cannot be protected).
 - If an NFS v2/v3 is already being protected, no NFS v4 will be listed in the choices.
 - If nothing is protected, then the list could contain both v2/v3 and v4.

Select the NFS export point to be protected from the drop-down list.

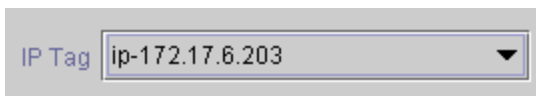


A screenshot of a software interface showing a dropdown menu labeled "Export Point". The selected option is "/export/public".

Click **Next** to proceed to the next dialog box.

5. The **IP Tag** dialog displays a drop-down list of tags corresponding to virtual IP addresses currently under LifeKeeper protection and in service on the server where the NFS resource is being created.

Select the **tag** for the virtual IP address used by clients to access the protected NFS file system.



A screenshot of a software interface showing a dropdown menu labeled "IP Tag". The selected option is "ip-172.17.6.203".

Note: At this point, LifeKeeper will check to ensure that there is a protected IP resource available. It will also validate that you have provided valid data to create your NFS resource hierarchy. If LifeKeeper detects a problem with either of these validations, an ERROR box will appear on the screen. If the directory paths are valid but there are errors with the NFS configuration itself, you may pause to correct these errors and continue with the hierarchy creation. You may even pause to create any LifeKeeper IP resources that are required.

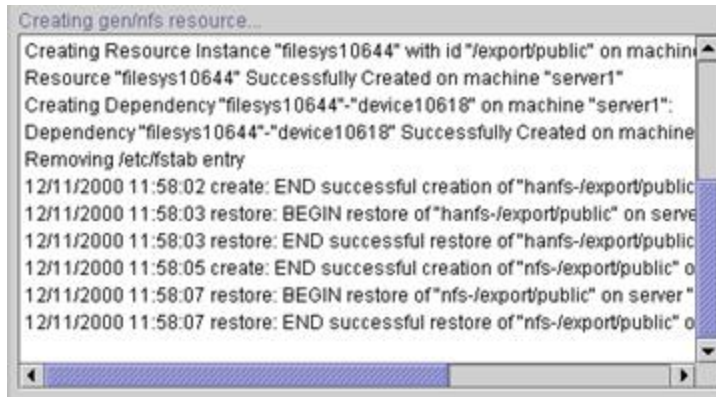
Note: If you are using other LifeKeeper Recovery Kits that have virtual IP address dependencies, you might want to create a different virtual IP address for the NFS resource. Otherwise, if the virtual IP resource fails over to a backup server, all of the resources that depend on that IP resource will fail over at the same time.

Click **Next** to proceed to the next dialog box.

6. Select or enter the **NFS Tag**. This is a tag name given to the NFS hierarchy. You can select the default or enter your own tag name.

A screenshot of a text input field labeled "NFS Tag". The field contains the text "nfs-/export/public".

When you click the **Create** button, the **Create Resource Wizard** will create your NFS resource.



When you click **Next**, LifeKeeper will launch the **Pre-Extend Wizard** that is discussed in [Extending Your Hierarchy](#).

Note: The NFS resource hierarchy should be created successfully at this point. However, error messages may be encountered indicating that the new NFS instance has failed to start correctly. Note that the new NFS hierarchy must be started (In Service) before it can be extended to another system. A failure to start may remove the hierarchy, but if not, you may pause at this point and correct the problem based on the error message displayed. If the errors are not correctable, you will only be given the choice to cancel which cancels the resource create.

Bring the new hierarchy In Service before proceeding with [extending your hierarchy](#).

Notes: Please disable automatic startup of `nfs-server.service` after creating NFS resources on RHEL 7.1 or later and SLES12 SP1 or later. Since it is necessary for `rpcbind.service` to be running at the startup of NFS resources, please configure `rpcbind.service` to start automatically.

Deleting a Resource Hierarchy

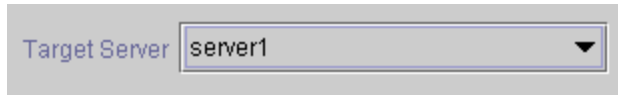
To delete a resource hierarchy from all the servers in your LifeKeeper environment, complete the following steps:

1. From the LifeKeeper GUI menu, select **Edit**, then **Resource**. From the drop-down menu, select **Delete Resource Hierarchy**.
2. Select the name of the **Target Server** where you will be deleting your NFS resource hierarchy.

Deleting a Resource Hierarchy

Note: This dialog box does not appear if you select the **Delete Resource** task by right-clicking from either of the following:

- The right pane on an individual resource instance
- The left pane on a global resource when the resource is on only one server



Click **Next** to proceed to the next dialog box.

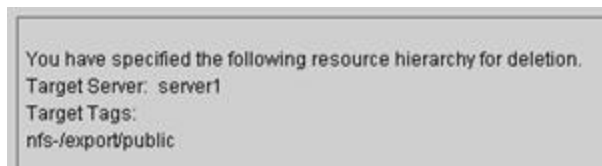
3. Select the **Hierarchy to Delete**. Identify the resource hierarchy you wish to delete and highlight it.

Note: If you selected the Delete Resource task by right-clicking from either the left pane on a global resource or the right pane on an individual resource instance, this dialog will not appear.



Click **Next** to proceed to the next dialog box.

4. An information box appears confirming your selection of the target server and the hierarchy you have selected to delete.



Click **Delete** to proceed to the next dialog box.

5. Another information box appears confirming that the NFS resource was deleted successfully.



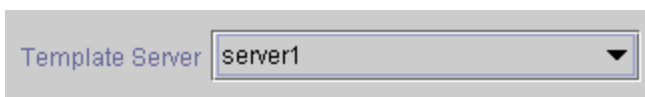
6. Click **Done** to exit out of the **Delete Resource Hierarchy** menu selection.

Extending Your Hierarchy

After you have created a hierarchy, you will want to extend that hierarchy to another server in the cluster. There are two possible scenarios to extend your resource instance from the template server to a target server. The first scenario is when you “Continue” from creating the resource into extending that resource to another server. The other scenario is when you enter the Extend Resource Hierarchy task from the edit menu as shown below. Both scenarios take you through the same dialog boxes (with a few exceptions, which are detailed below).

1. If you are entering the **Extend Wizard** from the **LifeKeeper GUI** menu, select **Edit**, then **Resource**. From the drop-down menu, select **Extend Resource Hierarchy**. This will launch the **Extend Resource Hierarchy** wizard.
2. The first dialog box to appear will ask you select the **Template Server** where your NFS resource hierarchy is currently in service. It is important to remember that the Template Server you select now and the **Tag to Extend** that you select in the next dialog box represent an in service resource hierarchy. An error message will appear if you select a resource tag that is not in service on the template server you selected. The drop-down box in this dialog provides the names of all the servers in your cluster.

Note: If you are entering the **Extend Resource Hierarchy** task immediately following the creation of a NFS resource hierarchy, this dialog box will not appear, since the wizard has already identified the template server in the create stage. This is also the case when you right click on either the NFS resource icon in the left hand pane or right-click on the NFS resource box in the right hand pane on the of the GUI window and choose **Extend Resource Hierarchy**.



Note: If you click the **Cancel** button at any time during the sequence of extending your hierarchy, LifeKeeper will cancel the extension process to that particular server. LifeKeeper will also unextend

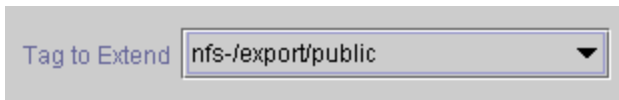
any dependent resources in the hierarchy (IP address or file system) that are currently extended past the cancellation point. However, if you have already extended the NFS resource hierarchy to another server, that instance will continue to be in effect until you specifically unextend it.

For example, let's say you have created your resource on Server 1 and extended that resource to Server 2. In the middle of extending the same resource to Server 3, you change your mind and click on the Cancel button inside one of the dialog boxes. This will cancel only your action to extend the resource to Server 3, not the extension you created to Server 2. If you want to remove Server 2 from this hierarchy, you must unextend the resource from Server 2.

Click **Next** to proceed to the next dialog box.

3. Select the **Tag to Extend**. This is the name of the NFS instance you wish to extend from the template server to the target server. The wizard will list in the drop down menu all the resources that you have created on the template server, which you selected in the previous dialog box.

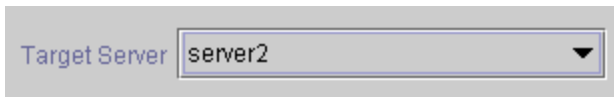
Note: Once again, if you are entering the **Extend Resource Hierarchy** task immediately following the creation of an NFS resource hierarchy, this dialog box will not appear, since the wizard has already identified the tag name of your NFS resource in the create stage. This is also the case when you right-click on either the NFS resource icon in the left hand pane or on the NFS resource box in the right hand pane of the GUI window and choose **Extend Resource Hierarchy**.



Tag to Extend

Click **Next** to proceed to the next dialog box.

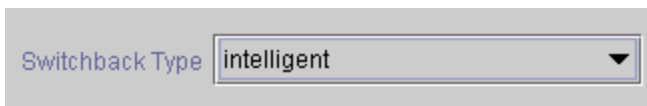
4. Select the **Target Server** where you are extending your NFS resource hierarchy. The drop-down box provides the names of the servers in your cluster that are not already in the selected hierarchy.



Target Server

Click Next to proceed to the next dialog box.

5. Select the **Switchback Type**. This dictates how the NFS instance will be switched back to the primary server when it comes back into service after a failover to the backup server. You can choose either intelligent or automatic. Intelligent switchback requires administrative intervention to switch the instance back to the primary/original server. Automatic switchback means the switchback will occur as soon as the primary server comes back on line and is under LifeKeeper protection.



Switchback Type

The switchback type can be changed later, if desired, from the **General** tab of the **Resource Properties** dialog box.

Click **Next** to proceed to the next dialog box.

6. Select or enter a **Template Priority**. This is the priority for the NFS hierarchy on the server where it is currently in service. Any unused priority value from 1 to 999 is valid, where a lower number means a higher priority (1=highest). The extend process will reject any priority for this hierarchy that is already in use by another system. The default value is recommended. Note: This selection will appear only for the initial extend of the hierarchy.

Click **Next**.

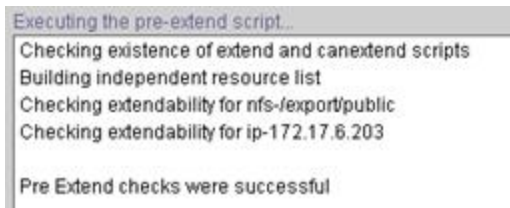
7. Select or enter the **Target Priority**. This is the priority for the new extended NFS hierarchy relative to equivalent hierarchies on other servers. Any unused priority value from 1 to 999 is valid, indicating a server's priority in the cascading failover sequence for the resource. A lower number means a higher priority (1=highest). Note that LifeKeeper assigns the number "1" to the server on which the hierarchy is created by default. The priorities need not be consecutive, but no two servers can have the same priority for a given resource.



A screenshot of a dialog box showing the 'Target Priority' field. The field is a text input box containing the number '10' and a dropdown arrow on the right side.

Click **Next**.

8. An information box will appear explaining that LifeKeeper has successfully checked your environment and that all the requirements for extending this NFS resource have been met. If there were some requirements that had not been met, LifeKeeper would not allow you to select Next, and the **Back** button would be enabled.



```
Executing the pre-extend script...
Checking existence of extend and canextend scripts
Building independent resource list
Checking extendability for nfs-fexport/public
Checking extendability for ip-172.17.6.203

Pre Extend checks were successful
```

If you click **Back**, you can make changes to your resource extension according to any error messages that may appear in the information box.

If you click **Cancel** now, you will need to come back and extend your NFS resource hierarchy to another server at some other time to put it under LifeKeeper protection.

When you click **Next**, LifeKeeper will launch you into the **Extend Resource Hierarchy** configuration task and the **NFS Tag** dialog box will display.

9. This screen provides information about the **Template Server**, **Tag to Extend**, **Target Server** and the default **NFS Tag**. The **NFS Tag** is a tag name given to the NFS hierarchy extension. You can select

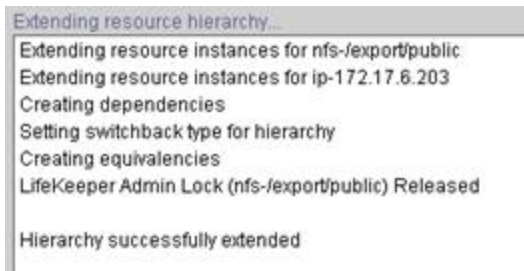
the default or enter your own tag name.



Click **Next** to proceed to the next dialog box.

10. An information box will appear verifying that the extension is being performed.

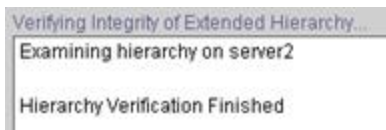
Note: If you have not already extended the IP resource to the target server, the NFS Server Recovery Kit extends it in the process of extending the NFS resource. Before displaying the extension verification information box, the Recovery Kit displays several additional dialog boxes related to the extension of the IP resource.



Click **Next Server** if you want to extend the same NFS resource instance to another server in your cluster. This will repeat the Extend Resource Hierarchy operation.

If you click **Finish**, LifeKeeper will verify that the extension of the NFS resource was completed successfully.

11. If you click **Finish**, the following screen appears.



12. Click **Done** to exit.

Note: Be sure to test the functionality of the new instance on both servers.

Testing Your Resource Hierarchy

Before testing your NFS resource hierarchy, you should validate your client setup as described below. You can then test your NFS resource hierarchy by initiating a manual switchover. This will simulate a failover of a resource instance from the primary server to the backup server.

Validating the Client Setup

In general, clients must mount the file system using the LifeKeeper-protected IP address you selected during the Create NFS Resource Hierarchy task. There is no client-side checking to ensure that you select the correct IP address, so you must carefully follow the validation steps below to ensure the client is using the correct IP number for the file system.

To validate the client setup, do the following:

1. Verify that no NFS instances are in service on the secondary system.
2. Mount the file system on the client using the correct LifeKeeper-protected IP address.
3. Perform a manual switchover to the secondary system and ensure that the NFS instance you just switched over is the only NFS instance currently in service on the secondary.
4. When the switchover has completed, ensure that the client can still access the file system.

Performing a Manual Switchover from the GUI

After you define the dependencies, LifeKeeper automatically controls the starting and stopping of the application whenever it detects faults, which initiate failover recovery. You can also manually initiate a switchover for administrative reasons, such as maintenance.

You can initiate a manual switchover from the LifeKeeper GUI by selecting Edit, Resource, and In Service from the drop-down menu. For example, an in service request executed on a backup server causes the application hierarchy to be placed in service on the backup server and taken out of service on the primary server.

If you execute the Out of Service request, the application is taken out of service without bringing it in service on the other server.

Note: To take an NFS resource out of service, you must take both the NFS resource and the associated HA-NFS resource out of service.

For activities within the application, all actions are those defined in the application's documentation. LifeKeeper does not regulate or control internal operations such as rollbacks and backing-up archives. Tape archiving and restoration are the responsibility of the application administrator.

Recovery Operations

When the primary server fails, the NFS Server Recovery Kit software performs the following tasks:

- Starts the NFS daemons if they are not running (the `rpc.lockd` and `rpc.statd` daemons will be monitored if file lock failover has been enabled).
- Exports the NFS file system.

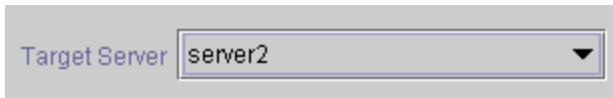
Unextending Your Hierarchy

Perform the following steps to unextend a resource hierarchy:

Unextending Your Hierarchy

1. From the **LifeKeeper GUI menu**, select **Edit** and **Resource**. From the drop-down menu, select **Unextend Resource Hierarchy**.
2. Select the **Target Server** where you want to unextend the NFS resource. It cannot be the server where the NFS resource is currently in service.

Note: If you selected the **Unextend** task by right-clicking from the right pane on an individual resource instance this dialog box will not appear.

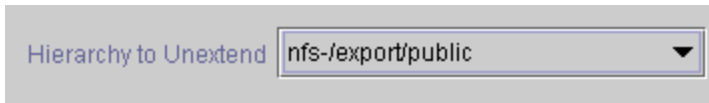


Target Server server2 ▼

Click **Next** to proceed to the next dialog box.

3. Select the **NFS Hierarchy to Unextend**.

Note: If you selected the **Unextend** task by right-clicking from either the left pane on a global resource or the right pane on an individual resource instance, this dialog will not appear.



Hierarchy to Unextend nfs-/export/public ▼

Click **Next** to proceed to the next dialog box.

4. An information box appears confirming the target server and the NFS resource hierarchy you have chosen to unextend.



You have specified the following resource hierarchy for unextend.
Target Server = server2
Target Tag = nfs-/export/public

Click **Unextend**.

5. Another information box appears confirming that the NFS resource was unextended successfully.

Unextending Your Hierarchy



6. Click **Done** to exit the **Unextend Resource Hierarchy** menu selection and return to the LifeKeeper GUI.

You will receive the warning **One or More Resources Unprotected** if the hierarchy is unextended down to one server.

Chapter 4: Troubleshooting

This section provides a list of messages that you may encounter while creating and extending a LifeKeeper NFS resource hierarchy or removing and restoring a resource. Where appropriate, it provides an additional explanation of the cause of an error and necessary action to resolve the error condition.

Messages from other SPS components are also possible. In these cases, please refer to the Message Catalog (located on our Technical Documentation site under “Search for an Error Code”) which provides a listing of all error codes, including operational, administrative and GUI, that may be encountered while using SIOS Protection Suite for Linux and, where appropriate, provides additional explanation of the cause of the error code and necessary action to resolve the issue. This full listing may be searched for any error code received, or you may go directly to one of the individual Message Catalogs for the appropriate SPS component.

Click the following topics for Troubleshooting help.

- [HA nfs-utils Installation and Configuration](#)
- [NFS Hierarchy Creation Errors](#)
- [NFS Extend Hierarchy Errors](#)
- [Hierarchy Restore, Remove and Recover Messages and Errors](#)
- [Hierarchy Delete Messages and Errors](#)

HA nfs-utils Installation and Configuration

Version 5.0.0 and greater of the LifeKeeper NFS Server Recovery Kit requires the installation and configuration of a high availability enabled `nfs-utils` package (on some OS distribution versions, `nfs-utils` is provided via another package). The Recovery Kit will attempt to verify the presence of this HA enabled `nfs-utils` package. If it fails to detect a correctly configured `nfs-utils` package, the LifeKeeper Installation Support setup script may need to be rerun or the server may need to be rebooted.

The LifeKeeper Installation Support setup script will configure `nfs-utils` for use in an HA environment and will also perform setup configuration for **NFS v4** exports. You must answer **YES** to this HA NFS question for any NFS version that will be used.

The configuration needed for **NFS v4** requires the movement of `rpc_pipefs` from `/var/lib/nfs` to `/var/lib`. To do this may require the unloading of kernel modules and the addition or modification of configuration and boot time scripts. A system reboot may be required if Installation Support is unable to unload and reload kernel modules after the change. If this should occur, the user will be notified of the need for a system reboot. Completing the `rpc_pipefs` setup including a system reboot is required for successful operation of LifeKeeper.

Hierarchy Delete Messages and Errors

Error Number	Error Message	Description
106015	Unable to restore the entry for export point "EXPORT POINT" in <i>/etc/exports</i> on server "SERVER"	Restore the entry manually.
106021	An entry for export point "EXPORT POINT" already exists in <i>/etc/exports</i> . The entry that was being used by the NFS Server Recovery Kit has been placed in the file "FILENAME"	Verify that <i>/etc/exports</i> has the correct export entry.
106049	Restore <i>statedir</i> from <i>/var/lib/.nfs.LK</i> to <i>/var/lib/nfs</i> failed on server SERVER.	Restoring the NFS state directory <i>/var/lib/nfs</i> failed. Try manually restoring the directory by moving <i>/var/lib/.nfs.LK</i> to <i>/var/lib/nfs</i> .

Hierarchy Restore, Remove and Recover Messages

Bringing an NFS Resource In-Service (Restore)

Error Number	Error Message	Description
106007	Cannot bring NFS or HANFS resource "TAG" in service on server "SERVER"	Review other error messages to determine the action to take. After correcting the problem, try bringing the resource in service manually.
106010	NFS is not running on server "SERVER". LifeKeeper will attempt to restart NFS.	This message is for information only. LifeKeeper will try to restart the NFS daemons automatically. If LifeKeeper encounters problems while restarting one of the daemons, you will receive a message that starting NFS failed.
106011	Starting NFS on server "SERVER" failed	LifeKeeper encountered a problem while restarting the NFS daemons. Try manually restarting NFS.
106012	The export point "EXPORT POINT" is not exported on server "SERVER". LifeKeeper will attempt to export the entry.	LifeKeeper has detected that the export point is no longer exported, and will try to export it.
106013	Unable to export "EXPORT POINT" on server "SERVER"	Try manually exporting the file system.
106014	Usage: USAGE STRING	Usage of command run with incorrect arguments. <i>Command line only.</i>

Error Number	Error Message	Description
106019	Executing command: "COMMAND"	This message is displayed when LifeKeeper restarts an NFS daemon or exports/unexports an export point. It provides additional information that can be useful if there is a problem.
106024	Unable to stop and restart rpc.mountd on "SERVER!"	During a hierarchy restore the rpc.mountd daemon process needed to be restarted and this process failed. Manually attempt to stop and restart the process to determine the error and the action to take.
106027	Open of "ABC" on server "SERVER" failed: "File not found"	The attempted open failed for the reason listed.
106028	Mount of /proc/fs/nfsd failed on server "SERVER!"	In 2.6 and later kernels /proc/fs/nfsd is used for client authentication and an attempt to mount it failed. Manually attempt to mount /proc/fs/nfsd to determine the failure.
106029	Unable to get exclusive lock on "/var/lib/nfs/rmtab" on server "SERVER"	Unable to place an exclusive lock on /var/lib/nfs/rmtab for update after 20 seconds indicating a problem with the file.
106030	Unable to restore client info for "123.45.678.90" on server "SERVER!": "ABC"	Attempts to failover client 123.45.678.90 locks to server SERVER1 failed for the listed reason. Correct the failure condition and attempt to restore the hierarchy again.
106037	Attempts to get exclusive lock on "/var/lib/nfs/rmtab" on server "SEVER" failed: "ABC"	Unable to place an exclusive lock on /var/lib/nfs/rmtab for updating. See the error message for the cause.
106039	Open of "FILE" on server "SERVER" failed: "ABC" or Attempt to get exclusive lock on "FILE" on server "SERVER" failed: "ABC"	An attempt to open or obtain an exclusive lock on a file has failed. See the error message for the cause.
106040	Multiple virtual IP addresses detected. In this release NFS lock failover only supports one virtual IP address.	Recreate the NFS resource hierarchies to use only one virtual IP address or set FAILOVERNFSLOCKS to false in the LifeKeeper defaults file.
106052	Unable to mount <i>rpc_pipefs</i> on "SERVER". Reason: "REASON".	<i>rpc_pipefs</i> was not mounted on SERVER and the mount attempt failed for REASON.
106053	<i>rpc_pipefs</i> successfully mounted on "SERVER"	<i>rpc_pipefs</i> was successfully mounted on SERVER.

Taking an NFS Resource Out of Service (Remove)

Error Number	Error Message	Description
106008	Unable to unexport the export point "EXPORT POINT" on server "SERVER"	Use the exportfs(8) command to unexport it.
106014	Usage: USAGE STRING	Usage of command run with incorrect arguments. <i>Command line only.</i>
106019	Executing command: "COMMAND"	This message is displayed when LifeKeeper restarts an NFS daemon or exports/unexports an export point. It provides additional information that can be useful if there is a problem.
106046	Unable to umount child bind directory "BIND MOUNT POINT" for export "EXPORT POINT" on "SERVER".	The attempt to umount the bind mount "BIND MOUNT POINT" for EXPORT POINT on SERVER failed.

Bringing an NFS Resource Back In Service (Recover)

The LifeKeeper core periodically checks the health of every NFS instance In Service on the local server by running an NFS “quickCheck” script. This script verifies the following:

- The file system is exported
- The NFS/HA-NFS daemons are running

If the instance is not fully functional, a “recover” script is invoked to attempt to restart the instance. This simply logs an error message, invokes “restore,” prints the final error or success message – depending on error or success of the “restore” script – and returns the same result as “restore.” If restore/recover fails, this instance is failed over to another server.

NFS Extend Hierarchy Errors

The error messages that might be displayed during NFS hierarchy extension are listed below, along with a suggested explanation for each. Note that these error messages appear when the GUI indicates it is “Executing the pre-extend script...” to validate the hierarchy prior to extending it to the new system.

During NFS Resource Hierarchy Creation on Target Server

Error Number	Error Message	Description
106016	"REQUIRED SOFTWARE" cannot be found or does not have the expected permissions on server "SERVER"	NFS must be installed on the primary server and all backup servers. Verify that the nfs-utils has been installed.
106017	The file system "FILE SYSTEM" on template server "SERVER" has a different mount point "MOUNT POINT" on server "SERVER"	The resources must be created with the same mount point on each server. Either unextend the file system hierarchy from the target server or recreate it with the same mount point on the template and target servers.
106018	Unable to copy the file "FILENAME" from server "SERVER" to server "SERVER"	Possible causes: <ul style="list-style-type: none"> • Communication path between the servers is down • File system is full of the target server • File system problems - steeleye-1kNFS package needs to be reinstalled
106020	The generated id "nfs-/export" conflicts with an existing resource id	The internally generated resource ID for the nfs or hanfs resource has produced a duplicate.
106022	The export point "EXPORT POINT" is in <code>/etc/exports</code> on the target server "SERVER"	Remove the export point from the <code>/etc/exports</code> file on the target server before trying to extend the resource.
106023	The export point "EXPORT POINT" is exported on the target server "SERVER"	Unexport the export point on the target server before trying to extend the resource.
106051	Unable to create active/active configurations with NFS v4 exports. Either "TEMPLATE SERVER" or "TARGET SERVER" currently protects an NFS v4 root export.	Unable to extend the NFS resource from the TEMPLATE SERVER to the TARGET SERVER as one or both of the servers already protects an NFS v4 export and active/active configurations with NFS v4 exports is not supported.

NFS Hierarchy Creation Errors

The error messages that might be displayed during the NFS hierarchy creation are listed below, along with a suggested explanation for each. The messages listed cover both the creation of the nfs and hanfs resources. Error messages displayed by the LifeKeeper core and by other recovery kits are not listed in this guide. Note that you may stop to correct any problem described here, and then continue with hierarchy creation from the point where you left off – including creating any new LifeKeeper resources you might need for your NFS configuration.

Note: In the following error messages, *Command line* only indicates that you can only receive the message if you are entering commands on the command line; you cannot receive it if you are using the LifeKeeper GUI. Additionally, at the end of hierarchy create a resource restore is initiated. See Hierarchy Restore, Remove and Recover Messages and Errors for an explanation of messages and errors that can occur during that process.

NFS Hierarchy Creation Errors

Error Number	Error Message	Description
106000	Export point not specified	You must specify the export point for the NFS file system when you create the resource hierarchy. <i>Command line only.</i>
106001	The path "EXPORT POINT" is not exported by NFS	The export point you specified is not currently exported by NFS. Use exportfs(8) to export the path and verify the path is in the <i>/var/lib/nfs/etabfile</i> . <i>Command line only.</i>
106002	create: The export point "EXPORTPOINT" on "server1" for client "CLIENT" either does not contain an FSID export option or the value is not unique. A unique FSID will be generated and "EXPORTPOINT" will be re-exported using the new FSID value.	All export point under LifeKeeper protection must use a unique fsid= export option for high availability NFS. The selected export did not meet this requirement so a unique value was generated followed by a re-export for the selected export point. Note: a client of "*" or "world" indicates the export point is available to all clients.
106003	Unable to create the export entry file for "EXPORT POINT" in LifeKeeper	Possible causes: <ul style="list-style-type: none"> • The file system is full on the target server. • File system problems - Steeleye-lkNFS package needs to be reinstalled.
106004	The export point "EXPORT POINT" is not on a shared file system on server "SERVER"	Make sure that the export point is for a shared file system. <i>Command line only.</i>
106005	Unable to create the HA-NFS hierarchy "TAG" with child resource "TAG" on server "SERVER"	Review the other error messages to determine the action to take.
106006	Unable to remove entry for export point "EXPORT POINT" from <i>/etc/exports</i> on server "SERVER"	Verify that the <i>/etc/exports</i> file exists and is readable.
106014	Usage: USAGE STRING	Usage of command run with incorrect arguments. <i>Command line only.</i>
106016	"REQUIRED SOFTWARE" cannot be found or does not have the expected permissions on server "SERVER"	NFS must be installed on the primary server and all backup servers. Verify that the nfs-utils has been installed
106019	Executing command: "COMMAND"	This message is displayed when LifeKeeper restarts an NFS daemon or exports/unexports an export point. It provides additional information that can be useful if there is a problem.
106020	The generated id "nfs-/export" conflicts with an existing resource id	The internally generated resource ID for the nfs or hanfs resource has produced a duplicate.

NFS Hierarchy Creation Errors

Error Number	Error Message	Description
106025	An unknown error has occurred while running "newtag" on server "Server1"	An unexpected error occurred while running the command newtag to generate a tag for the nfs for hanfs resource.
106026	Adding dependency between HA-NFS resource "hanfs-/export" and filesysresource "export" on server "server1" failed.	Dependency creation between the selected hanfs resource and the filesys resource has failed for unknown reasons. See output for more information.
106027	Open of "ABC" on server "SERVER" failed: "File not found"	The attempted open failed for the reason listed.
106029	Unable to get exclusive lock on "/var/lib/nfs/rmtab" on server "SERVER"	Unable to place an exclusive lock on /var/lib/nfs/rmtab for update after 20 seconds indicating a problem with the file.
106031	Re-export of "/export" to add FSID option failed on server "SERVER"	The export point "/export" did not contain a fsid argument and the re-export after one was generated failed. Manually add a fsid argument to the /etc/exports entry and re-export to determine the failure.
106032	Dependent IP resource tag name not specified	You must specify the resource tag for the protected IP address when you create the NFS resource hierarchy. Command line only.
106033	Selected IP resource "TAG" does not exist on server "SERVER"	You must create the IP resource on the specified server before you can create the NFS resource. Also, make sure that you typed the IP resource correctly when you entered the command. Command line only.
106034	Adding dependency between NFS resource "TAG" and IP resource "TAG" on server "SERVER" failed	Verify the IP resource is in-service on the server where the NFS resource is being created. Command line only.
106035	Creation of HA-NFS resource "TAG" on server "SERVER" failed	Review other error messages to determine the action to take.
106036	Adding dependency between IP resource "TAG" and HA-NFS resource "TAG" on server "SERVER" failed	Verify that the IP resource is in-service on the server where the HA-NFS resource is being created. Command line only.
106037	Attempts to get exclusive lock on "/var/lib/nfs/rmtab" on server "SEVER" failed: "ABC"	Unable to place an exclusive lock on /var/lib/nfs/rmtab for updating. See the error message for the cause.
106038	Unable to create directory "ABC" on server "SERVER": "ABC"	An attempt to create a directory on the exported file system has failed. See the error message for the cause.

NFS Hierarchy Creation Errors

Error Number	Error Message	Description
106039	Open of "FILE" on server "SERVER" failed: "ABC" or Attempt to get exclusive lock on "FILE" on server "SERVER" failed: "ABC"	An attempt to open or obtain an exclusive lock on a file has failed. See the error message for the cause.
106041	The selected IP resource "ip-123.45.678.90" is not ISP on server "SERVER"	The selected IP resource does exist on the server but is not currently in service. Bring the IP resource in service on the server and then re-attempt the creation. Command line only.
106045	Unable to remove bind mount entry "TAG" on "SERVER".	An unexpected error occurred while attempting to remove bind mount entries from <i>/etc/fstab</i> for TAG on SERVER.
106048	Multiple NFS v4 root exports found on "SERVER".	Multiple NFS v4 psuedo file systems found on SERVER where only one is supported.
106050	Unable to protect more than one NFS v4 export or a combination of NFS v4 and NFS v3 exports.	Attempting to protect a mix of NFS v2/v3 exports with NFS v4 which is not supported.