



DataKeeper for Linux

v7.5

Technical Documentation

January 2012

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 © 2012
By SIOS Technology Corp.
San Mateo, CA U.S.A.
All rights reserved

Table of Contents

Chapter 1: Introduction	1
Introduction/Overview.....	1
Mirroring with SteelEye DataKeeper for Linux.....	1
DataKeeper Features.....	1
Synchronous vs. Asynchronous Mirroring.....	2
Synchronous Mirroring.....	2
Asynchronous Mirroring.....	2
How SteelEye DataKeeper Works.....	2
Synchronization (and Resynchronization).....	3
Standard Mirror Configuration.....	4
N+1 Configuration.....	4
Multiple Target Configuration.....	5
SteelEye DataKeeper Resource Hierarchy.....	6
Failover Scenarios.....	7
Scenario 1.....	7
Scenario 2.....	8
Scenario 3.....	8
Scenario 4.....	8
Chapter 2: Installation and Configuration	11
Installing and Configuring SteelEye DataKeeper for Linux.....	11
Before Configuring Your DataKeeper Resources.....	11
Hardware and Software Requirements.....	12
Hardware Requirements.....	12
Software Requirements.....	12
General Configuration.....	13

Network and LifeKeeper Configuration	13
Changing the Data Replication Path.....	13
Determine Network Bandwidth Requirements.....	14
Measuring Rate of Change on a Linux System (Physical or Virtual).....	14
Determine Network Bandwidth Requirements.....	14
Measuring Basic Rate of Change.....	15
Measuring Detailed Rate of Change.....	15
Analyze Collected Detailed Rate of Change Data	16
Graph Detailed Rate of Change Data	21
WAN Configuration	25
Confirm Failover and Block Resource Failover Settings.....	25
Confirm Failover On.....	25
Block Resource Failover On.....	26
Setting the Flags on Each Server.....	27
SteelEye DataKeeper for Linux Resource Types.....	27
Replicate New File System.....	27
Replicate Existing File System.....	28
DataKeeper Resource	28
Resource Configuration Tasks.....	28
Overview.....	29
Creating a DataKeeper Resource Hierarchy.....	29
Extending Your Hierarchy.....	30
Extending a DataKeeper Resource.....	32
Unextending Your Hierarchy.....	34
Deleting a Resource Hierarchy.....	34
Taking a DataKeeper Resource Out of Service.....	35
Bringing a DataKeeper Resource In Service.....	35
Testing Your Resource Hierarchy.....	36
Performing a Manual Switchover from the LifeKeeper GUI.....	36
Chapter 3: Administration.....	37

Administering SteelEye DataKeeper for Linux.....	37
Viewing Mirror Status.....	37
GUI Mirror Administration.....	38
Pause and Resume.....	39
Pause Mirror.....	39
Resume Mirror.....	40
Force Mirror Online.....	40
Set Compression Level.....	40
Set Rewind Log Location.....	40
Set Rewind Log Max Size.....	41
Create and View Rewind Bookmarks.....	41
Rewind and Recover Data.....	41
Command Line Mirror Administration.....	44
Mirror Actions.....	44
Examples:.....	44
Mirror Settings.....	44
Examples:.....	45
Bitmap Administration.....	45
Monitoring Mirror Status via Command Line.....	46
Example:.....	46
Server Failure.....	47
Resynchronization.....	47
Avoiding Full Resynchronizations.....	48
Method 1.....	48
Procedure.....	48
Method 2.....	49
Procedure.....	49
Chapter 4: Multi-Site Cluster.....	51
SteelEye Protection Suite for Linux Multi-Site Cluster.....	51
Overview.....	51

Migrating to a Multi-Site Cluster Environment	51
SteelEye Protection Suite for Linux Multi-Site Cluster	52
Multi-Site Cluster Configuration Considerations	53
Multi-Site Cluster Restrictions	54
Creating a SteelEye Protection Suite for Linux Multi-Site Cluster Resource Hierarchy	54
Replicate New File System	55
Replicate Existing File System	57
DataKeeper Resource	59
Extending Your Hierarchy	61
Extending a DataKeeper Resource	63
Extending a Hierarchy to a Disaster Recovery System	63
Configuring the Restore and Recovery Setting for Your IP Resource	66
Multi-Site Cluster Troubleshooting	66
Migrating to a Multi-Site Cluster Environment	66
Requirements	67
Before You Start	68
Performing the Migration	68
Successful Migration	77
Chapter 4: Troubleshooting	79
Chapter 4: Glossary	82
SteelEye DataKeeper for Linux Glossary of Terms	82

Chapter 1: Introduction

Introduction/Overview

SteelEye DataKeeper for Linux provides an integrated data mirroring capability for LifeKeeper environments. This feature enables LifeKeeper resources to operate in shared and non-shared storage environments.

[Mirroring with SteelEye DataKeeper for Linux](#)

[How SteelEye DataKeeper Works](#)

Mirroring with SteelEye DataKeeper for Linux

SteelEye DataKeeper for Linux offers an alternative for customers who want to build a high availability cluster (using SteelEye LifeKeeper) without shared storage or who simply want to replicate business-critical data in real-time between servers.

SteelEye DataKeeper uses either synchronous or asynchronous volume-level mirroring to replicate data from the primary server (mirror source) to one or more backup servers (mirror targets).

DataKeeper Features

SteelEye DataKeeper includes the following features:

- Allows data to be reliably, efficiently and consistently mirrored to remote locations over any TCP/IP-based Local Area Network (LAN) or Wide Area Network (WAN).
- Supports synchronous or asynchronous mirroring.
- Transparent to the applications involved because replication is done at the block level below the file system.
- Supports multiple simultaneous mirror targets including cascading failover to those targets when used with LifeKeeper.
- Supports point-in-time data rewind to allow recovery of lost or corrupted data.
- Built-in network compression allows higher maximum throughput on Wide Area Networks.
- Supports all major file systems (see the LifeKeeper Release Notes Product Description for more information regarding journaling file system support).

Synchronous vs. Asynchronous Mirroring

- Provides failover protection for mirrored data.
- Integrates into the LifeKeeper Graphical User Interface.
- Fully supports other LifeKeeper Application Recovery Kits.
- Automatically resynchronizes data between the primary server and backup servers upon system recovery.
- Monitors the health of the underlying system components and performs a local recovery in the event of failure.
- Supports Stonith devices for I/O fencing. For details, refer to the STONITH topic.

Synchronous vs. Asynchronous Mirroring

Understanding the differences between synchronous and asynchronous mirroring will help you choose the appropriate mirroring method for your application environment.

Synchronous Mirroring

SteelEye DataKeeper provides real-time mirroring by employing a synchronous mirroring technique in which data is written simultaneously on the primary and backup servers. For each write operation, DataKeeper forwards the write to the target device(s) and awaits remote confirmation before signaling I/O completion. The advantage of synchronous mirroring is a high level of data protection because it ensures that all copies of the data are always identical. However, the performance may suffer due to the wait for remote confirmation, particularly in a WAN environment.

Asynchronous Mirroring

With asynchronous mirroring, each write is made to the source device and then a copy is queued to be transmitted to the target device(s). This means that at any given time, there may be numerous committed write transactions that are waiting to be sent from the source to the target device. The advantage of asynchronous mirroring is better performance because writes are acknowledged when they reach the primary disk, but it can be less reliable because if the primary system fails, any writes that are in the asynchronous write queue will not be transmitted to the target. To mitigate this issue, SteelEye DataKeeper makes an entry to an intent log file for every write made to the primary device.

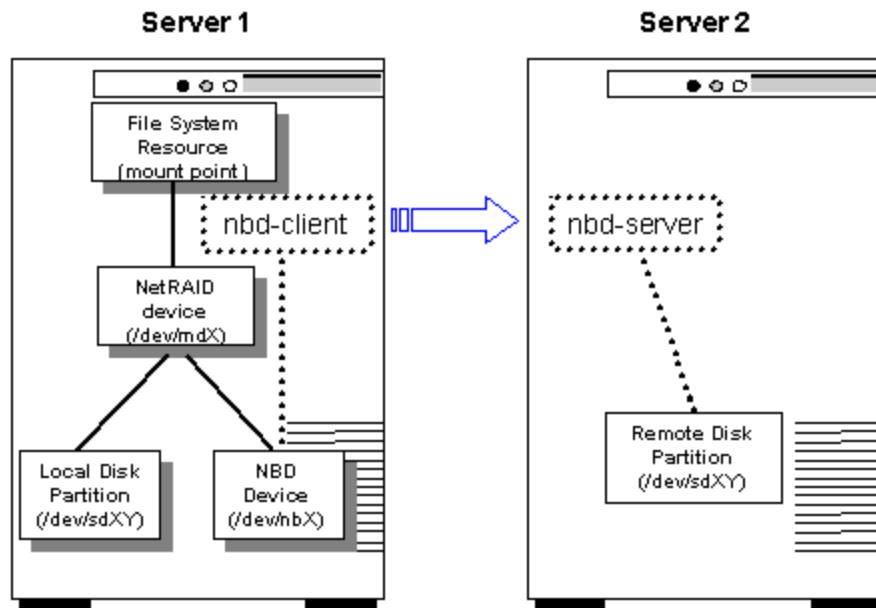
The intent log is a bitmap file indicating which data blocks are out of sync between the primary and target mirrors. In the event of a server failure, the intent log can be used to avoid a full resynchronization (or resync) of the data.

Note: The intent log can be used in both asynchronous and synchronous mirroring modes, but the intent log with asynchronous mirroring is supported only with a 2.6.16 or higher Linux kernel.

How SteelEye DataKeeper Works

SteelEye DataKeeper creates and protects NetRAID devices. A NetRAID device is a RAID1 device

that consists of a local disk or partition and a Network Block Device (NBD) as shown in the diagram below.



A LifeKeeper supported file system can be mounted on a NetRAID device like any other storage device. In this case, the file system is called a replicated file system. LifeKeeper protects both the NetRAID device and the replicated file system.

The NetRAID device is created by building the DataKeeper resource hierarchy. Extending the NetRAID device to another server will create the NBD device and make the network connection between the two servers. SteelEye DataKeeper starts replicating data as soon as the NBD connection is made.

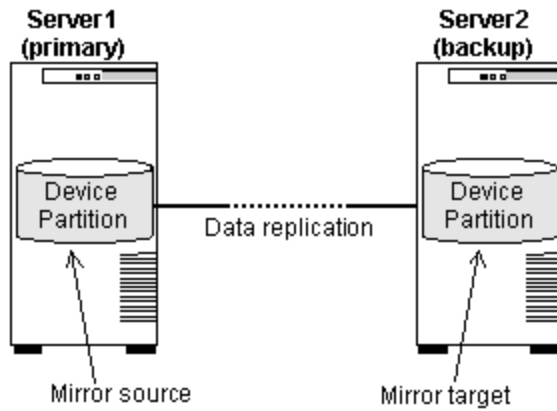
The nbd-client process executes on the primary server and connects to the nbd-server process running on the backup server.

Synchronization (and Resynchronization)

After the DataKeeper resource hierarchy is created and before it is extended, it is in a degraded mode; that is, data will be written to the local disk or partition only. Once the hierarchy is extended to the backup (target) system, SteelEye DataKeeper synchronizes the data between the two systems and all subsequent writes are replicated to the target. If at any time the data gets "out-of-sync" (i.e., a system or network failure occurs) SteelEye DataKeeper will automatically resynchronize the data on the source and target systems. If the mirror was configured to use an intent log (bitmap file), SteelEye DataKeeper uses it to determine what data is out-of-sync so that a full resynchronization is not required. If the mirror was not configured to use a bitmap file, then a full resync is performed after any interruption of data replication.

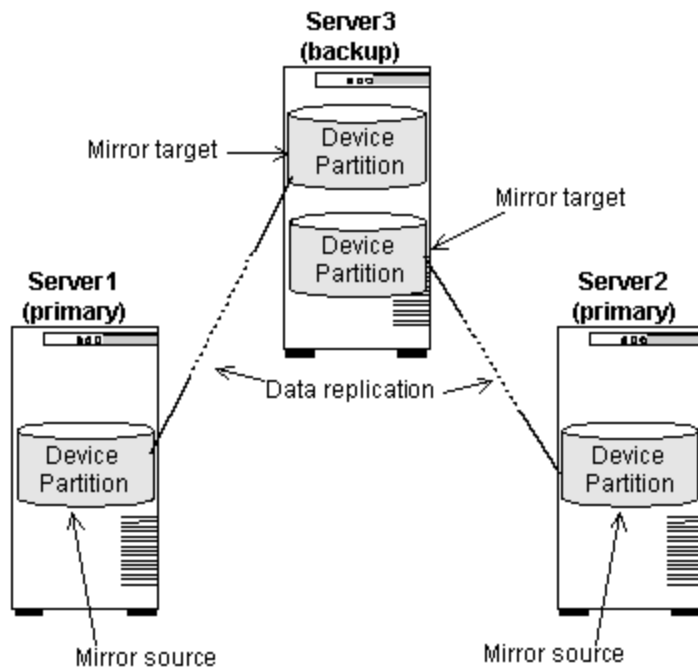
Standard Mirror Configuration

The most common mirror configuration involves two servers with a mirror established between local disks or partitions on each server, as shown below. Server1 is considered the primary server containing the mirror source. Server2 is the backup server containing the mirror target.



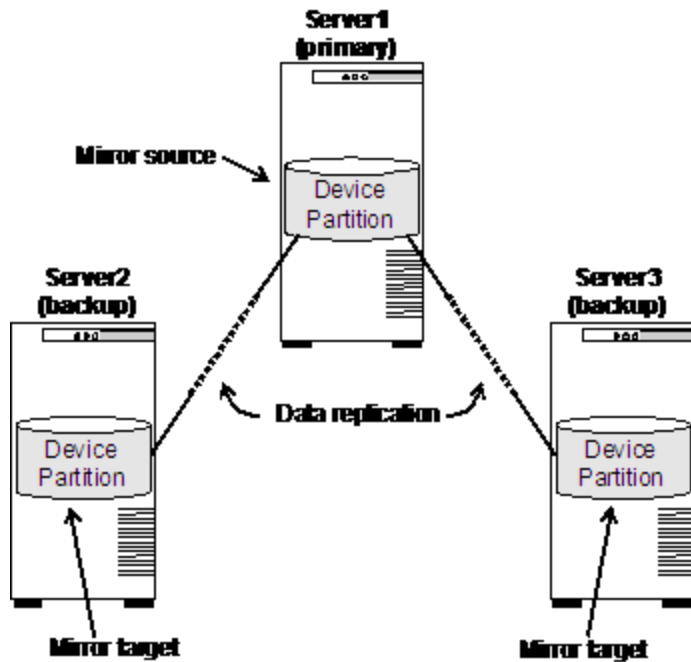
N+1 Configuration

A commonly used variation of the standard mirror configuration above is a cluster in which two or more servers replicate data to a common backup server. In this case, each mirror source must replicate to a separate disk or partition on the backup server, as shown below.



Multiple Target Configuration

When used with an appropriate Linux distribution and kernel version 2.6.7 or higher, SteelEye DataKeeper can also replicate data from a single disk or partition on the primary server to multiple backup systems, as shown below.



A given source disk or partition can be replicated to a maximum of 7 mirror targets, and each mirror target must be on a separate system (i.e. a source disk or partition cannot be mirrored to more than one disk or partition on the same target system).

This type of configuration allows the use of LifeKeeper’s cascading failover feature, providing multiple backup systems for a protected application and its associated data.

SteelEye DataKeeper Resource Hierarchy

The following example shows a typical DataKeeper resource hierarchy as it appears in the LifeKeeper GUI:

Hierarchies		adam eve sophocles		
Active Protected				
ext3-sdr		10 StandBy	1 Active	20 StandBy
datarep-ext3-sdr		10 Paused	1 Source	20 Target

The resource *datarep-ext3-sdr* is the NetRAID resource, and the parent resource *ext3-sdr* is the file system resource. Note that subsequent references to the DataKeeper resource in this documentation refer to both resources together. Because the file system resource is dependent on the NetRAID resource, performing an action on the NetRAID resource will also affect the file system resource above it.

Failover Scenarios

The following four examples show what happens during a failover using SteelEye DataKeeper. In these examples, the LifeKeeper for Linux cluster consists of two servers, Server 1 (primary server) and Server 2 (backup server).

Scenario 1

Server 1 has successfully completed its replication to Server 2 after which Server 1 becomes inoperable.



Result: Failover occurs. Server 2 now takes on the role of primary server and operates in a degraded mode (with no backup) until Server 1 is again operational. SteelEye DataKeeper will then initiate a resynchronization from Server 2 to Server 1. This will be a full resynchronization on kernel 2.6.18 and lower. On kernels 2.6.19 and later or with RedHat Enterprise Linux 5.4 kernels 2.6.18-164 or later (or a supported derivative of RedHat 5.4 or later), the resynchronization will be partial, meaning only the changed blocks recorded in the bitmap files on the source and target will need to be synchronized.

Note: SteelEye DataKeeper sets the following flag on the server that is currently acting as the mirror source:

```
$LKROOT/subsys/scsi/resources/netraid/$TAG_last_owner
```

When Server 1 fails over to Server 2, this flag is set on Server 2. Thus, when Server 1 comes back up; SteelEye DataKeeper removes the last owner flag from Server 1. It then begins resynchronizing the data from Server 2 to Server 1.

Scenario 2

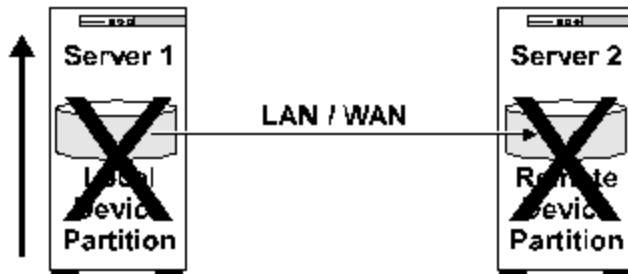
Scenario 2

Considering scenario 1, Server 2 (still the primary server) becomes inoperable during the resynchronization with Server 1 (now the backup server).

Result: Because the resynchronization process did not complete successfully, there is potential for data corruption. As a result, LifeKeeper will not attempt to fail over the DataKeeper resource to Server 1. Only when Server 2 becomes operable will LifeKeeper attempt to bring the DataKeeper resource in-service (ISP) on Server 2.

Scenario 3

Both Server 1 (primary) and Server 2 (target) become inoperable. Server 1 (primary) comes back up first.



Result: Server 1 will not bring the DataKeeper resource in-service. The reason is that if a source server goes down, and then it cannot communicate with the target after it comes back online, it sets the following flag:

```
$LKROOT/subsys/scsi/resources/netraid/$TAG_data_corrupt
```

This is a safeguard to avoid resynchronizing data in the wrong direction. In this case you will need to force the mirror online on Server1, which will delete the *data_corrupt* flag and bring the resource into service on Server 1. See [Force Mirror Online](#).

Note: The user must be certain that Server 1 was the last primary before removing the *\$TAG_data_corrupt* file. Otherwise data corruption might occur. You can verify this by checking for the presence of the *last_owner* flag.

Scenario 4

Both Server 1 (primary) and Server 2 (target) become inoperable. Server 2 (target) comes back up first.



Result: LifeKeeper will not bring the DataKeeper resource ISP on Server 2. When Server 1 comes back up, LifeKeeper will automatically bring the DataKeeper resource ISP on Server 1.

Chapter 2: Installation and Configuration

Installing and Configuring SteelEye DataKeeper for Linux

[Hardware/Software Requirements](#)

Before Configuring Your DataKeeper Resources

The following topics contain information for consideration before beginning to create and administer your DataKeeper resources. They also describe the three types of DataKeeper resources. Please refer to the Configuration section of the LifeKeeper for Linux Technical Documentation for instructions on configuring LifeKeeper Core resource hierarchies.

[General Configuration](#)

[Network and LifeKeeper Configuration](#)

[Changing the Data Replication Path](#)

[Determine Network Bandwidth Requirements](#)

[Measuring Rate of Change on a Linux System \(Physical or Virtual\)](#)

[WAN Configuration](#)

[Confirm Failover and Block Resource Failover Settings](#)

[SteelEye DataKeeper for Linux Resource Types](#)

[Resource Configuration Tasks](#)

[Creating a DataKeeper Resource Hierarchy](#)

[Extending Your Hierarchy](#)

[Unextending Your Hierarchy](#)

[Deleting a Resource Hierarchy](#)

[Taking a DataKeeper Resource Out of Service](#)

[Bringing a DataKeeper Resource In Service](#)

[Testing Your Resource Hierarchy](#)

Hardware and Software Requirements

Your LifeKeeper configuration should meet the following requirements **prior** to the installation of SteelEye DataKeeper.

Hardware Requirements

- **Servers** - Two or more LifeKeeper for Linux supported servers.
- **IP Network Interface Cards** - Each server requires at least one network interface card. Remember, however, that a LifeKeeper cluster requires two communication paths; two separate LAN-based communication paths using dual independent sub-nets are recommended, and at least one of these should be configured as a private network. However using a combination of TCP and TTY is also supported.

Note: Due to the nature of software mirroring, network traffic between servers can be heavy. Therefore, it is recommended that you implement a separate private network for your SteelEye DataKeeper devices which may require additional network interface cards on each server.

- **Disks or Partitions** - Disks or partitions on the primary and backup servers that will act as the source and target disks or partitions. The target disks or partitions must be at least as large as the source disk or partition.

Note: With the release of SteelEye Data Replication 7.1.1, it became possible to replicate an entire disk, one that has not been partitioned (i.e. `/dev/sdd`). Previous versions of SteelEye Data Replication required that a disk be partitioned (even if it was a single large partition; i.e. `/dev/sdd1`) before it could be replicated. SteelEye Data Replication 7.1.1 removed that restriction.

Software Requirements

- **Operating System** – SteelEye DataKeeper can be used with any major Linux distribution based on the 2.6 Linux kernel. See the LifeKeeper for Linux Release Notes for a list of supported distributions. Asynchronous mirroring and intent logs are supported only on distributions that use a 2.6.16 or later Linux kernel. Multiple target support (i.e., support for more than 1 mirror target) requires a 2.6.7 or later Linux kernel.
- **LifeKeeper Installation Support** - In most cases, you will need to install the following package (see the “Product Requirements” section in the LifeKeeper for Linux Release Notes for specific SteelEye DataKeeper requirements):

HADR-generic-2.6

This package must be installed on each server in your LifeKeeper cluster **prior** to the installation of SteelEye DataKeeper. The HADR package is located on the LifeKeeper Installation Support CD, and the appropriate package is automatically installed by the Installation Support **setup** script.

- **LifeKeeper Software** - You must install the same version of the LifeKeeper Core on each of your servers. You must also install the same version of each recovery kit that you plan to use on each server. See the LifeKeeper for Linux Release Notes for specific LifeKeeper requirements.
- **SteelEye DataKeeper software** - Each server in your LifeKeeper cluster requires SteelEye DataKeeper software. Please see the Installation section in the LifeKeeper for Linux Technical Documentation for specific instructions on the installation and removal of SteelEye DataKeeper.

General Configuration

- The size of the target disks or partitions (on the backup servers) must be equal to or greater than the size of the source disk or partition (on the primary server).
- Once the DataKeeper resource is created and extended, the synchronization process will delete existing data on the target disks or partitions and replace it with data from the source partition.

Network and LifeKeeper Configuration

- The network path that is chosen for data replication between each pair of servers must also already be configured as a LifeKeeper communication path between those servers. To change the network path, see [Changing the Data Replication Path](#).
- When configuring DataKeeper resources, avoid using an interface/address already in use by a LifeKeeper IP resource that has local recovery enabled. For example, if a LifeKeeper IP resource is configured on interface *eth1* having local recovery enabled with interface *eth2*, DataKeeper resources should avoid using either *eth1* or *eth2*. Enabling local recovery will disable the interface during switchover to the backup interface which can cause SteelEye DataKeeper failure.
- This release of SteelEye DataKeeper does not support Automatic Switchback for DataKeeper resources. Additionally, the Automatic Switchback restriction is applicable for any other LifeKeeper resource sitting on top of a DataKeeper resource.

Changing the Data Replication Path

Starting with LK 7.1, mirror endpoints can be modified using `lk_chg_value`. For example, to change a mirror endpoint from IP address 192.168.0.1 to 192.168.1.1:

1. `lkstop (lk_chg_value cannot be run while LifeKeeper is running)`
2. `lk_chg_value -o 192.168.0.1 -n 192.168.1.1`
3. `lkstart`

Execute these commands on all servers involved in the mirror(s) that are using this IP address.

Note: This command will also modify communication paths that are using the address in question.

Determine Network Bandwidth Requirements

Prior to installing SteelEye DataKeeper, you should determine the network bandwidth requirements for replicating your current configuration whether you are employing virtual machines or using physical Linux servers. If you are employing virtual machines (VMs), use the method [Measuring Rate of Change on a Linux System \(Physical or Virtual\)](#) to measure the rate of change for the virtual machines that you plan to replicate. This value indicates the amount of network bandwidth that will be required to replicate the virtual machines.

After determining the network bandwidth requirements, ensure that your network is configured to perform optimally. If your network bandwidth requirements are above your current available network capacity, you may need to consider one or more of the following options:

- Enable compression in SteelEye DataKeeper (or in the network hardware, if possible)
- Increase your network capacity
- Reduce the amount of data being replicated
- Create a local, non-replicated storage repository for temporary data and swap files
- Manually schedule replication to take place daily at off-peak hours

Measuring Rate of Change on a Linux System (Physical or Virtual)

DataKeeper for Linux can replicate data across any available network. In Multi-Site or Wide Area Network (WAN) configurations, special consideration must be given to the question, "Is there sufficient bandwidth to successfully replicate the partition and keep the mirror in the mirroring state as the source partition is updated throughout the day?"

Keeping the mirror in the mirroring state is critical because a switchover of the partition is not allowed unless the mirror is in the mirroring state.

Determine Network Bandwidth Requirements

Prior to installing SteelEye DataKeeper, you should determine the network bandwidth requirements for replicating your data. Use the method below to measure the rate of change for the data that you plan to replicate. This value indicates the amount of network bandwidth that will be required to replicate that data.

After determining the network bandwidth requirements, ensure that your network is configured to perform optimally. If your network bandwidth requirements are above your current available network capacity, you must consider one or more of the following options:

- Enable compression in DataKeeper, or in the network hardware, if possible
- Create a local, non-replicated storage repository for temporary data and swap files that don't

really need to be replicated

- Reduce the amount of data being replicated
- Increase your network capacity

SteelEye DataKeeper handles short bursts of write activity by adding that data to its async queue. However, make sure that over any extended period of time, the disk write activity for all replicated volumes combined remains, on average, below the amount of change that DataKeeper and your network can transmit.

If the network capacity is not sufficient to keep up with the rate of change that occurs on your disks, and the async queue fills up, the mirror will revert to synchronous behavior, which can negatively affect performance of the source server.

Measuring Basic Rate of Change

Use the following command to determine file(s) or partition(s) to be mirrored. For example `/dev/sda3`, and then measure the amount of data written in a day:

```
MB_START=`awk '/sda3 / { print $10 / 2 / 1024 }'
/proc/diskstats`
```

... wait for a day ...

```
MB_END=`awk '/sda3 / { print $10 / 2 / 1024 }'
/proc/diskstats`
```

The daily rate of change, in MB, is then `MB_END - MB_START`.

SteelEye DataKeeper can mirror daily, approximately:

T1 (1.5Mbps) - 14,000 MB/day (14 GB)

T3 (45Mbps) - 410,000 MB/day (410 GB)

Gigabit (1Gbps) - 5,000,000 MB/day (5 TB)

Measuring Detailed Rate of Change

The best way to collect Rate of Change data is to log disk write activity for some period of time (one day, for instance) to determine what the peak disk write periods are.

To track disk write activity, create a cron job which will log the timestamp of the system followed by a dump of `/proc/diskstats`. For example, to collect disk stats every two minutes, add the following link to `/etc/crontab`:

```
*/2 * * * * root ( date ; cat /proc/diskstats ) >> /path_
to/filename.txt
```

Analyze Collected Detailed Rate of Change Data

... wait for a day, week, etc ... then disable the cron job and save the resulting data file in a safe location.

Analyze Collected Detailed Rate of Change Data

The `roc-calc-diskstats` utility analyzes data collected in the previous step. This utility takes a `/proc/diskstats` output file that contains output, logged over time, and calculates the rate of change of the disks in the dataset.

roc-calc-diskstats

```
#!/usr/bin/perl
# Copyright (c) 2011, SIOS Technology, Corp.
# Author: Paul Clements
use strict;
sub msg {
printf STDERR @_ ;
}
sub dbg {
return if (! $ENV{'ROC_DEBUG'});
msg @_ ;
}
$0 =~ s@^\.*/@@; # basename
sub usage {
msg "Usage: $0 <interval> <start-time> <iostat-data-file> [dev-list]\n";
msg "\n";
msg "This utility takes a /proc/diskstats output file that contains\n";
msg "output, logged over time, and calculates the rate of change of\n";
msg "the disks in the dataset\n";
msg "OUTPUT_CSV=1 set in env. dumps the full stats to a CSV file on\n";
msg "STDERR\n";
msg "\n";
msg "Example: $0 1hour \"jun 23 12pm\" steeleye-iostat.txt sdg,sdh\n";
msg "\n";
msg "interval - interval between samples\n";
msg "start time - the time when the sampling starts\n";
msg "iostat-data-file - collect this with a cron job like:\n";
msg "\t0 * * * * (date ; cat /proc/diskstats) >> /root/diskstats.txt\n";
msg "dev-list - list of disks you want ROC for (leave blank for all)\n";
exit 1;
}
```

```

usage if (@ARGV < 3);
my $interval = TimeHuman($ARGV[0]);
my $starttime = epoch($ARGV[1]);
my $file = $ARGV[2];
my $blksize = 512; # /proc/diskstats is in sectors
my %devs = map { $_ => 1 } split /,/, $ARGV[3];
my %stat;
my $firsttime;
my $lasttime;
# datestamp divides output
my %days = ( 'Sun' => 1, 'Mon' => 1, 'Tue' => 1, 'Wed' => 1,
  'Thu' => 1, 'Fri' => 1, 'Sat' => 1);
my %fields = ( 'major' => 0,
  'minor' => 1,
  'dev' => 2,
  'reads' => 3,
  'reads_merged' => 4,
  'sectors_read' => 5,
  'ms_time_reading' => 6,
  'writes' => 7,
  'writes_merged' => 8,
  'sectors_written' => 9,
  'ms_time_writing' => 10,
  'ios_pending' => 11,
  'ms_time_total' => 12,
  'weighted_ms_time_total' => 13 );
my $devfield = $fields{'dev'};
my $scalffield = $ENV{'ROC_CALC_FIELD'} || $fields{'sectors_written'};
dbg "using field $scalffield\n";
open(FD, "$file") or die "Cannot open $file: $!\n";
foreach (<FD>) {
  chomp;
  @_ = split;
  if (exists($days{$_[0]}) { # skip datestamp divider
    if ($firsttime eq '') {
      $firsttime = join ' ', @_[0..5];
    }
    $lasttime = join ' ', @_[0..5];
  }
  next;
}

```

Analyze Collected Detailed Rate of Change Data

```
}
next if ($_[0] !~ /[0-9]/); # ignore
if (!%devs || exists $devs{$_[$devfield]}) {
push @{$stat{$_[$devfield]}}, $_[$calcfld];
}
}
@{$stat{'total'}} = totals(\%stat);
printf "Sample start time: %s\n", scalar(localtime($starttime));
printf "Sample end time: %s\n", scalar(localtime($starttime +
((@{$stat{'total'}} - 1) * $interval)));
printf "Sample interval: %ss #Samples: %s Sample length: %ss\n",
$interval, (@{$stat{'total'}} - 1), (@{$stat{'total'}} - 1) * $interval;
print "(Raw times from file: $firsttime, $lasttime)\n";
print "Rate of change for devices " . (join ', ', sort keys %stat) .
"\n";

foreach (sort keys %stat) {
my @vals = @{$stat{$_}};
my ($max, $maxindex, $roc) = roc($_, $blksize, $interval, @vals);
printf "$_ peak:%sB/s (%sb/s) (@ %s) average:%sB/s (%sb/s)\n",
HumanSize($max), HumanSize($max * 8), scalar localtime($starttime +
($maxindex * $interval)), HumanSize($roc), HumanSize($roc * 8);
}

# functions
sub roc {
my $dev = shift;
my $blksize = shift;
my $interval = shift;
my ($max, $maxindex, $i, $first, $last, $total);
my $prev = -1;
my $first = $_[0];
if ($ENV{'OUTPUT_CSV'}) { print STDERR "$dev," }
foreach (@_) {
if ($prev != -1) {
if ($_ < $prev) {
dbg "wrap detected at $i ($_ < $prev)\n";
$prev = 0;
}
my $this = ($_ - $prev) * $blksize / $interval;
if ($this > $max) {
$max = $this;

```

```

$maxindex = $i;
}
if ($ENV{'OUTPUT_CSV'}) { print STDERR "$this," }
}
$prev = $_; # store current val for next time around
$last = $_;
$i++;
}
if ($ENV{'OUTPUT_CSV'}) { print STDERR "\n" }
return ($max, $maxindex, ($last - $first) * $blksize / ($interval * ($i
- 1)));
}
sub totals { # params: stat_hash
my $stat = shift;
my @totalvals;
foreach (keys %$stat) {
next if (!defined($stat{$_}));
my @vals = @{$stat{$_}};
my $i;
foreach (@vals) {
$totalvals[$i++] += $_;
}
}
return @totalvals;
}
# converts to KB, MB, etc. and outputs size in readable form
sub HumanSize { # params: bytes/bits
my $bytes = shift;
my @suffixes = ( '', 'K', 'M', 'G', 'T', 'P' );
my $i = 0;
while ($bytes / 1024.0 >= 1) {
$bytes /= 1024.0;
$i++;
}
return sprintf("%.1f %s", $bytes, $suffixes[$i]);
}
# convert human-readable time interval to number of seconds
sub TimeHuman { # params: human_time
my $time = shift;

```

Analyze Collected Detailed Rate of Change Data

```
my %suffixes = ('s' => 1, 'm' => 60, 'h' => 60 * 60, 'd' => 60 * 60 *
24);
$time =~ /^([0-9]*)(.*?)$/;
$time = $1;
my $suffix = (split //, $2)[0]; # first letter from suffix
if (exists $suffixes{$suffix}) {
$time *= $suffixes{$suffix};
}
return $time;
}
sub epoch { # params: date
my $date = shift;
my $seconds = `date +%s' --date "$date" 2>&1`;
if ($? != 0) {
die "Failed to recognize time stamp: $date\n";
}
return $seconds;
}
```

Usage:

```
# ./roc-calc-diskstats <interval> <start_time> <diskstats-data-
file> [dev-list]
```

Usage Example (Summary only):

```
# ./roc-calc-diskstats 2m "Jul 22 16:04:01" /root/diskstats.txt
sdb1,sdb2,sdc1 > results.txt
```

The above example dumps a summary (with per disk peak I/O information) to *results.txt*

Usage Example (Summary + Graph Data):

```
# export OUTPUT_CSV=1
# ./roc-calc-diskstats 2m "Jul 22 16:04:01" /root/diskstats.txt
sdb1,sdb2,sdc1 2> results.csv > results.txt
```

The above example dumps graph data to *results.csv* and the summary (with per disk peak I/O information) to *results.txt*

Example Results (from results.txt)

Sample start time: Tue Jul 12 23:44:01 2011

Sample end time: Wed Jul 13 23:58:01 2011

Sample interval: 120s #Samples: 727 Sample length: 87240s

(Raw times from file: Tue Jul 12 23:44:01 EST 2011, Wed Jul 13 23:58:01 EST 2011)

Rate of change for devices dm-31, dm-32, dm-33, dm-4, dm-5, total

dm-31 peak:0.0 B/s (0.0 b/s) (@ Tue Jul 12 23:44:01 2011)
average:0.0 B/s (0.0 b/s)

dm-32 peak:398.7 KB/s (3.1 Mb/s) (@ Wed Jul 13 19:28:01 2011)
average:19.5 KB/s (156.2 Kb/s)

dm-33 peak:814.9 KB/s (6.4 Mb/s) (@ Wed Jul 13 23:58:01 2011)
average:11.6 KB/s (92.9 Kb/s)

dm-4 peak:185.6 KB/s (1.4 Mb/s) (@ Wed Jul 13 15:18:01 2011)
average:25.7 KB/s (205.3 Kb/s)

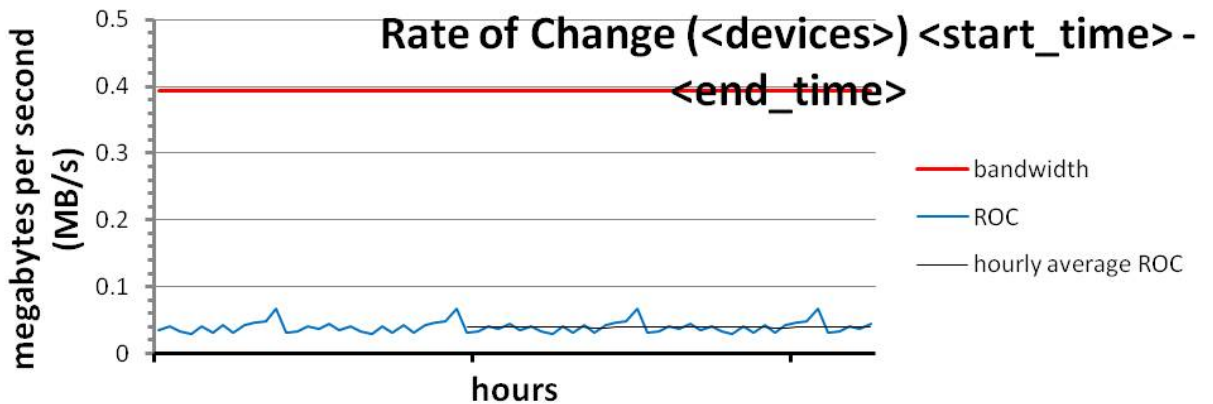
dm-5 peak:2.7 MB/s (21.8 Mb/s) (@ Wed Jul 13 10:18:01 2011)
average:293.0 KB/s (2.3 Mb/s)

total peak:2.8 MB/s (22.5 Mb/s) (@ Wed Jul 13 10:18:01 2011)
average:349.8 KB/s (2.7 Mb/s)

Graph Detailed Rate of Change Data

To help understand your specific bandwidth needs over time, SIOS has created a template spreadsheet called diskstats-template.xlsx. This spreadsheet contains sample data which can be overwritten with the data collected by roc-calc-diskstats.

diskstats-template



1. Open results.csv, and select **all rows**, including the total column.

Graph Detailed Rate of Change Data

	A	B	C	D	E	F	G	H	I	J	K	L
1	dm-31	0	0	0	0	0	0	0	0	0	0	0
2	dm-32	3549.867	6826.667	3549.867	273.0667	7099.733	3549.867	6826.667	3686.4	341.3333	7099.733	3549.8
3	dm-33	3857.067	4505.6	3310.933	1911.467	4846.933	2935.467	4471.467	3310.933	1911.467	4710.4	2935.4
4	dm-4	2218.667	2389.333	2150.4	1570.133	2525.867	2116.267	2389.333	2013.867	4300.8	2833.067	1809.0
5	dm-5	25326.93	26683.73	23449.6	25164.8	26419.2	23048.53	28612.27	21367.47	35140.27	32145.07	40797.1
6	total	34952.53	40405.33	32460.8	28919.47	40891.73	31650.13	42299.73	30378.67	41693.87	46788.27	49092.1

- Open **diskstats-template.xlsx**, select the **diskstats.csv** worksheet.



- In cell 1-A, right-click and select **Insert Copied Cells**.
- Adjust the **bandwidth** value in the cell towards the bottom left of the worksheet to reflect an amount of bandwidth you have allocated for replication.

Units: Megabits/second (Mb/sec)

Note: The cells to the right will automatically be converted to bytes/sec to match the raw data collected.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	dm-31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	dm-32	3549.86667	6826.66667	3549.867	273.0667	7099.733	3549.867	6826.667	3686.4	341.3333	7099.733	3549.867	6826.667	3276.8	273.0667	6826.667	3549.867	6826.667
3	dm-33	3857.06667	4505.6	3310.933	1911.467	4846.933	2935.467	4471.467	3310.933	1911.467	4710.4	2935.467	4710.4	2935.467	2798.933	4676.267	3857.067	4710.4
4	dm-4	2218.66667	2389.33333	2150.4	1570.133	2525.867	2116.267	2389.333	2013.867	4300.8	2833.067	1809.067	27955.2	1570.133	2286.933	2525.867	2116.267	2628.267
5	dm-5	25326.9333	26683.73333	23449.6	25164.8	26419.2	23048.53	28612.27	21367.47	35140.27	32145.07	40797.87	28492.8	23338.67	28561.07	27067.73	27784.53	29849.6
6	total	34952.5333	40405.33333	32460.8	28919.47	40891.73	31650.13	42299.73	30378.67	41693.87	46788.27	49092.27	67985.07	31121.07	33920	41096.53	37307.73	44014.93
7																		
8	bandwidth (Mb/s)																	
9		10	1310720	1310720	1310720	1310720	1310720	1310720	1310720	1310720	1310720	1310720	1310720	1310720	1310720	1310720	1310720	1310720

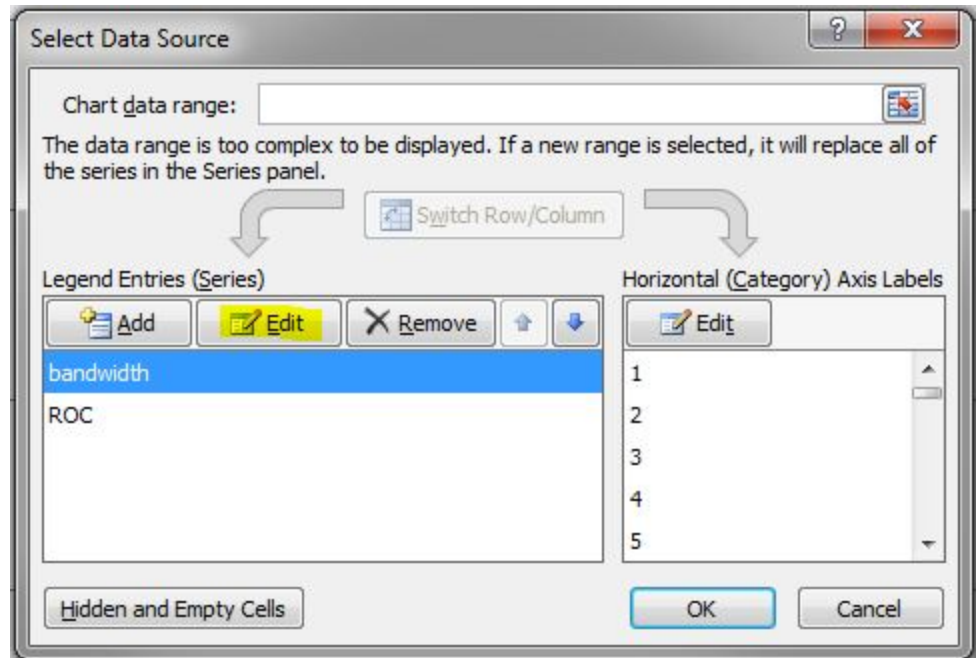
- Make a note of the following row/column numbers:
 - Total (row 6 in screenshot below)
 - Bandwidth (row 9 in screenshot below)
 - Last datapoint (column R in screenshot below)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	dm-31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	dm-32	3549.86667	6826.66667	3549.867	273.0667	7099.733	3549.867	6826.667	3686.4	341.3333	7099.733	3549.867	6826.667	3276.8	273.0667	6826.667	3549.867	6826.667
3	dm-33	3857.06667	4505.6	3310.933	1911.467	4846.933	2935.467	4471.467	3310.933	1911.467	4710.4	2935.467	4710.4	2935.467	2798.933	4676.267	3857.067	4710.4
4	dm-4	2218.66667	2389.33333	2150.4	1570.133	2525.867	2116.267	2389.333	2013.867	4300.8	2833.067	1809.067	27955.2	1570.133	2286.933	2525.867	2116.267	2628.267
5	dm-5	25326.9333	26683.73333	23449.6	25164.8	26419.2	23048.53	28612.27	21367.47	35140.27	32145.07	40797.87	28492.8	23338.67	28561.07	27067.73	27784.53	29849.6
6	total	34952.5333	40405.33333	32460.8	28919.47	40891.73	31650.13	42299.73	30378.67	41693.87	46788.27	49092.27	67985.07	31121.07	33920	41096.53	37307.73	44014.93
7																		
8	bandwidth (Mb/s)																	
9		10	1310720	1310720	1310720	1310720	1310720	1310720	1310720	1310720	1310720	1310720	1310720	1310720	1310720	1310720	1310720	1310720

- Select the **bandwidth vs ROC** worksheet.



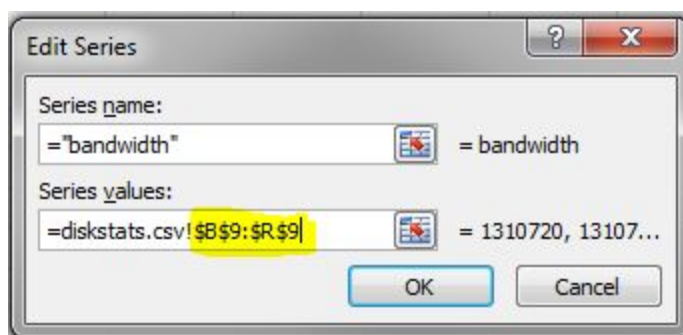
7. Right-click on the graph and select **Select Data...**
 - a. Adjust **Bandwidth Series**
 - i. From the **Series** list on the left, select **bandwidth**
 - ii. Click **Edit**



- iii. Adjust the **Series Values:** field with the following syntax:

`"=diskstats.csv!B<row>:$<final_column>$<row>"`

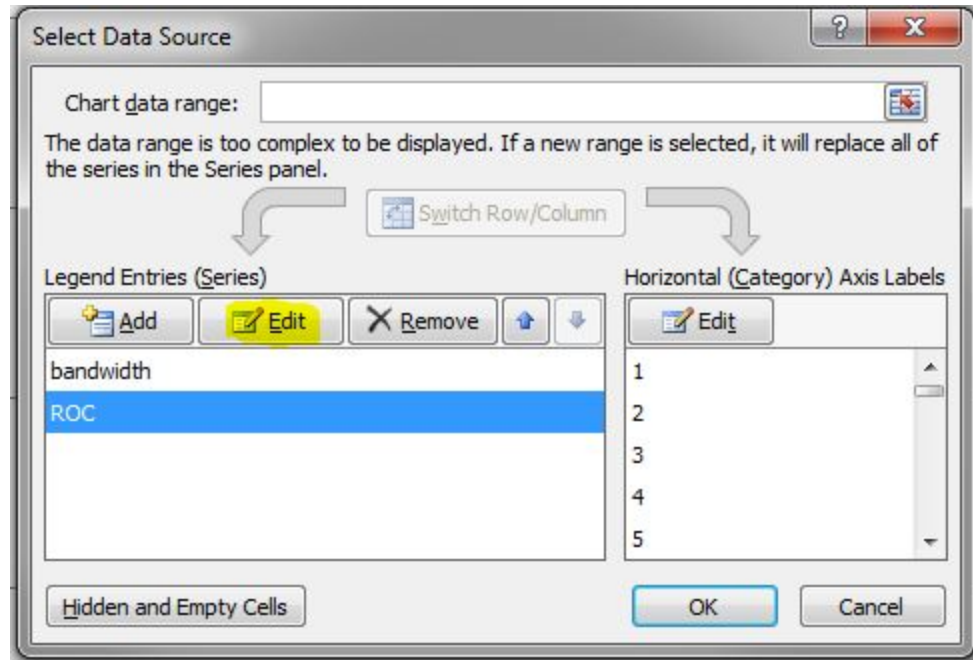
example: `"=diskstats.csv!B9:$R:$9"`



- iv. Click **OK**

b. Adjust **ROC Series**

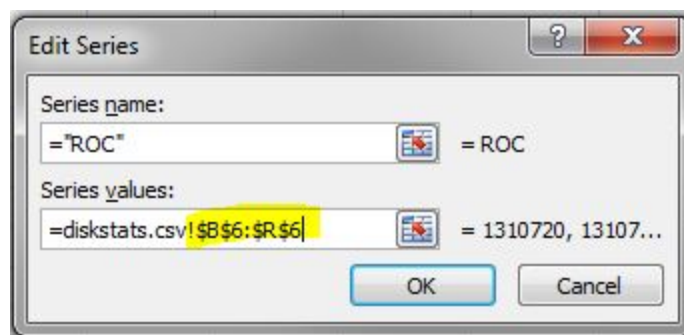
- i. From the **Series** list on the left, select **ROC**
- ii. Click **Edit**



iii. Adjust the **Series Values:** field with the following syntax:

"=diskstats.csv!\$B\$<row>:\$<final_column>\$<row>"

example: "=diskstats.csv!\$B\$6:\$R:\$6"



iv. Click **OK**

- c. Click **OK** to exit the Wizard
8. The Bandwidth vs ROC graph will update. Please analyze your results to determine if you have sufficient bandwidth to support replication of your data.

WAN Configuration

Using SteelEye DataKeeper in a WAN environment requires special configuration due to the nature of WAN networking. The following tips are recommended:

- To prevent false failover, you should enable manual failover confirmation. Because most WANs are somewhat less reliable than LANs and because typical WAN mirror configurations will have only one comm path, this is usually a good idea. With this option enabled, a LifeKeeper failover will proceed only if the user confirms the failover by using the **Ik_confirmso** command. Refer to the `Ik_confirmso` man page for more details.
- Determine the proper value for `LKDR_ASYNC_LIMIT`, based upon the latency and throughput of the WAN. The `LKDR_ASYNC_LIMIT` parameter (which is set in `/etc/default/LifeKeeper`) determines the number of outstanding asynchronous write operations (per mirror) that SteelEye DataKeeper will allow. The default value for this parameter is 256, but a larger number may increase write performance of the mirror. The disadvantage to increasing this value is that more data will be allowed to be out of sync between the primary and secondary at any given time.
- If you are mirroring a large amount of data over a slow WAN link, it may be desirable to avoid the initial full data resynchronization and instead ship or otherwise transport a copy of the source disk or partition to the remote (disaster recovery) site. To avoid the initial resynchronization, follow the steps in [Avoiding Full Resynchronizations](#).

Important: This procedure is not necessary if you created your hierarchy using the “New Replicated Filesystem” option in the LifeKeeper GUI. The “New Replicated Filesystem” option has been optimized to avoid the full initial resync.
- If the WAN link experiences periods of downtime in excess of 15 seconds on a regular basis, it may also be wise to tune the LifeKeeper heartbeat parameters. See [Tuning the LifeKeeper Heartbeat](#) for details.

Confirm Failover and Block Resource Failover Settings

Make sure you review and understand the following descriptions, examples and considerations before setting the **Confirm Failover** or **Block Resource Failover** in your LifeKeeper environment. These settings are available from the command line or on the **Properties** panel in the **LifeKeeper GUI**.

Confirm Failover On

Definition – Enables manual failover confirmation from System A to System B (where System A is the server whose properties are being displayed in the **Properties Panel** and System B is the system to the left of the checkbox). If this option is set on a system, it will require a manual confirmation by a system administrator before allowing LifeKeeper to perform a failover recovery of a system that it detects as failed.

Block Resource Failover On

Use the `lk_confirmso` command to confirm the failover. By default, the administrator has ten minutes to run this command. This time can be changed by modifying the CONFIRMSOTO setting in `/etc/default/LifeKeeper`. If the administrator does not run the `lk_confirmso` command within the time allowed, the failover will either proceed or be blocked. By default, the failover will proceed. This behavior can be changed by modifying the COMFIRMSODEF setting in `/etc/default/LifeKeeper`.

Example: If you wish to block automatic failovers completely, you should set the **Confirm Failover On** option in the **Properties** panel and also set CONFIRMSODEF to **1** (block failover) and CONFIRMSOTO to **0** (don't wait to decide on the failover action).

When to select this setting:

This setting is used in most Disaster Recovery, XenServer and other WAN configurations where the configuration does not include redundant heartbeat communication paths.

In a regular site (non-multi-site cluster and non-XenServer), open the **Properties** page from one server and then select the server that you want the **Confirm Failover flag** to be set on.

For a Multi-site **WAN** configuration: **Enable** manual failover confirmation

For a Multi-site **LAN** configuration: **Do not** enable manual failover confirmation

In a multi-site cluster environment – from the non-disaster system, select the DR system and check the set confirm failover flag. You will need to open the **Properties** panel and select this setting for each non-disaster server in the cluster.

In a XenServer environment, all servers in the list (not just the DR site) need to be checked.

Block Resource Failover On

Definition - By default, all resource failures will result in a recover event that will attempt to recover the failed resource on the local system. If local recovery fails or is not enabled, then LifeKeeper transfers the resource hierarchy to the next highest priority system for which the resource is defined. However, if this setting is selected on a designated system(s), all resource transfers due to a resource failure will be blocked from the given system.

When the setting is enabled, the following message is logged:

Local recovery failure, failover blocked, MANUAL INTERVENTION REQUIRED

Conditions/Considerations:

In a Multi-site configuration, **do not select** Block Failover for any server in the configuration.

In a XenServer environment, **select** Block Failover for each system in the cluster.

Remember: This setting will **not** affect failover behavior if there is a complete system failure. It will only block failovers due to local resource failures.

Setting the Flags on Each Server

1. Log in to the LifeKeeper GUI and select a server in your cluster. If the **Properties** panel option is selected on the **View** menu, the **Properties** panel will display (on the right side of the GUI).

On the **General** tab in the bottom of the panel, your system configuration will be displayed:

The screenshot shows a configuration panel with the following content:

Set Confirm Failover:
Configures the **confirmsotbuzzard.sc.steeleye.com** flag on each target system with the checkbox enabled.

Set Block Resource Failover:
Configures the **block_failover** flag on each target system with the checkbox enabled.

	Set Confirm Failover On	Set Block Resource Failover On
buzzard.sc.steeleye.com	<input type="checkbox"/>	<input type="checkbox"/>
alexander	<input type="checkbox"/>	<input type="checkbox"/>
vulture.sc.steeleye.com	<input type="checkbox"/>	<input type="checkbox"/>

2. In the **Set Confirm Failover On** column, select the checkbox for each of the other servers in the cluster.
3. In the **Set Block Resource Failover On** column, select the checkbox for each of the servers in the cluster as required.

IMPORTANT CONSIDERATION FOR MULTI-SITE CLUSTER CONFIGURATIONS: Do **not** check the **Block Resource Failover On** fields for the servers in a Multi-Site Cluster configuration.

4. Click **OK**.

SteelEye DataKeeper for Linux Resource Types

When creating your DataKeeper resource hierarchy, LifeKeeper will prompt you to select a resource type. There are several different DataKeeper resource types. The following information can help you determine which type is best for your environment.

Replicate New File System

Choosing a [New Replicated File System](#) will create/extend the NetRAID device, mount the given mount point on the NetRAID device and put both the LifeKeeper supported file system and the

Replicate Existing File System

NetRAID device under LifeKeeper protection. The local disk or partition will be formatted.

CAUTION: All data will be deleted.

Replicate Existing File System

Choosing [Replicate Existing File System](#) will use a currently mounted disk or partition and create a NetRAID device without deleting the data on the disk or partition. SteelEye DataKeeper will unmount the local disk or partition, create the NetRAID device using the local disk or partition and mount the mount point on the NetRAID device. It will then put both the NetRAID device and the LifeKeeper supported file system under LifeKeeper protection.

Important: If you are creating SteelEye Protection Suite for Linux Multi-Site Cluster hierarchies, your application will be stopped during the create process. You will need to restart your application once you have finished creating and extending your hierarchies.

DataKeeper Resource

Choosing a [DataKeeper Resource](#) will create/extend the NetRAID device and put it under LifeKeeper protection without a file system. You might choose this replication type if using a database that can use a raw I/O device.

In order to allow the user continued data access, SteelEye DataKeeper will not attempt to unmount and delete a NetRAID device if it is currently mounted. The user must manually unmount it before attempting a manual switchover and mount it on the other server after the manual switchover.

Note: After the DataKeeper resource has been created, should you decide to protect a manually mounted file system with LifeKeeper, you can do so as follows:

1. Format the NetRAID device with a LifeKeeper supported file system.
2. Mount the NetRAID device.
3. Create and extend a file system hierarchy using the NetRAID device as if it were a shared storage disk or partition.

LifeKeeper's file system recovery kit will now be responsible for mounting/unmounting it during failover.

Resource Configuration Tasks

You can perform all SteelEye DataKeeper configuration tasks via the LifeKeeper Graphical User Interface (GUI). The LifeKeeper GUI provides a guided interface to configure, administer and monitor SteelEye DataKeeper resources.

Overview

The following tasks are available for configuring SteelEye DataKeeper:

- **Create a Resource Hierarchy** - Creates a DataKeeper resource hierarchy.
- **Delete a Resource Hierarchy** - Deletes a DataKeeper resource hierarchy.
- **Extend a Resource Hierarchy** - Extends a DataKeeper resource hierarchy from the primary server to a backup server.
- **Unextend a Resource Hierarchy** - Unextends (removes) a DataKeeper resource hierarchy from a single server in the LifeKeeper cluster.
- **Create Dependency**- Creates a child dependency between an existing resource hierarchy and another resource instance and propagates the dependency changes to all applicable servers in the cluster.
- **Delete Dependency**- Deletes a resource dependency and propagates the dependency changes to all applicable servers in the cluster.
- **In Service** - Activates a resource hierarchy.
- **Out of Service** - Deactivates a resource hierarchy.
- **View/Edit Properties** - View or edit the properties of a resource hierarchy.

Creating a DataKeeper Resource Hierarchy

If you are creating a DataKeeper resource hierarchy in a [Multi-Site Cluster](#) environment, refer to the procedures at the end of this section after you select the **Hierarchy Type**.

Perform the following on your primary server:

1. Select **Edit > Server > Create Resource Hierarchy**

The **Create Resource Wizard** dialog will appear.

2. Select the **Data Replication** option from the drop down list and click **Next** to continue.
3. You will be prompted for the following information. When the **Back** button is active in any of the dialog boxes, you can go back to the previous dialog box. This is helpful should you encounter any error requiring you to correct the previously entered information. You may click **Cancel** at any time to cancel the entire creation process.

Field	Tips
Switchback Type	<p>You must select intelligent switchback. This means that after a failover to the backup server, an administrator must manually switch the DataKeeper resource back to the primary server.</p> <p>CAUTION: This release of SteelEye DataKeeper does not support Automatic Switchback for DataKeeper resources. Additionally, the Automatic Switchback restriction is applicable for any other LifeKeeper resource sitting on top of a DataKeeper resource.</p>
Server	<p>Select the name of the server where the NetRAID device will be created (typically this is your primary server). All servers in your cluster are included in the drop down list box.</p>
Hierarchy Type	<p>Choose the data replication type you wish to create by selecting one of the following:</p> <ul style="list-style-type: none"> • Replicate New File System • Replicate Existing File System • DataKeeper Resource
Bitmap File	<p>Select or edit the name of the bitmap file used for intent logging. If you choose None, then an intent log will not be used and every resynchronization will be a full resync instead of a partial resync.</p>
Enable Asynchronous Replication ?	<p>Select Yes to allow this replication resource to support asynchronous replication to target systems. Select No if you will use synchronous replication to all targets. You will be asked later to choose the actual type of replication, asynchronous or synchronous, when the replication resource is extended to each target server. (See Mirroring with SteelEye DataKeeper for a discussion of both replication types.) If you want the replication to any of these targets to be performed asynchronously, you should choose Yes here, even if the replication to other targets will be done synchronously.</p>

The next sequence of dialog boxes depends on which **Hierarchy Type** you have chosen. While some of the dialog boxes may be the same for each Hierarchy Type, their sequence and the required information may be slightly different. The next three topics take you through the remainder of the Hierarchy creation process.

- [DataKeeper Resource](#)
- [Replicate New File System](#)
- [Replicate Existing File System](#)

Extending Your Hierarchy

This operation can be started from the **Edit** menu or initiated automatically upon completing the **Create Resource Hierarchy** option, in which case you should refer to Step 2 below.

1. On the **Edit** menu, select **Resource**, then **Extend Resource Hierarchy**. The **Pre-Extend Wizard** appears. If you are unfamiliar with the Extend operation, click **Next**. If you are familiar with the LifeKeeper **Extend Resource Hierarchy** defaults and want to bypass the prompts for input/confirmation, click **Accept Defaults**.
2. The **Pre-Extend Wizard** will prompt you to enter the following information.

Note: The first two fields appear only if you initiated the Extend from the **Edit** menu.

Field	Tips
Template Server	<p>Select the TemplateServer where your <i>DataKeeper</i> 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 (activated) resource hierarchy.</p> <p>An error message will appear if you select a resource tag that is not in service on the template server you have selected. The drop down box in this dialog provides the names of all the servers in your cluster.</p>
Tag to Extend	<p>This is the name of the <i>DataKeeper</i> instance you wish to extend from the template server to the target server. The drop down box will list all the resources that you have created on the template server.</p>
Target Server	<p>Enter or select the server you are extending to.</p>
Switchback Type	<p>You must select intelligent switchback. This means that after a failover to the backup server, an administrator must manually switch the <i>DataKeeper</i> resource back to the primary server.</p> <p>CAUTION: This release of SteelEye <i>DataKeeper</i> does not support Automatic Switchback for <i>DataKeeper</i> resources. Additionally, the Automatic Switchback restriction is applicable for any other LifeKeeper resource sitting on top of a SteelEye <i>DataKeeper</i> resource.</p>
Template Priority	<p>Select or enter a Template Priority. This is the priority for the <i>DataKeeper</i> 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.</p> <p>Note: This selection will appear only for the initial extend of the hierarchy.</p>
Target Priority	<p>Select or enter the Target Priority. This is the priority for the new extended <i>DataKeeper</i> 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.</p>

After receiving the message that the pre-extend checks were successful, click Next.

Extending a DataKeeper Resource

Depending upon the hierarchy being extended, LifeKeeper will display a series of information boxes showing the Resource Tags to be extended, some of which cannot be edited.

3. Click **Next** to launch the **Extend Resource Hierarchy** configuration task.
4. The next section lists the steps required to complete the extension of a DataKeeper resource to another server.

Extending a DataKeeper Resource

1. After you have been notified that your pre-extend script has executed successfully, you will be prompted for the following information:

Field	Tips
Mount Point	Enter the name of the file system mount point on the target server. (This dialog will not appear if there is no LifeKeeper-protected filesystem associated with the DataKeeper Resource.)
Root Tag	Select or enter the Root Tag. This is a unique name for the filesystem resource instance on the target server. (This dialog will not appear if there is no LifeKeeper-protected filesystem associated with the DataKeeper Resource.)
Target Disk or Partition	<p>Select the disk or partition where the replicated file system will be located on the target server.</p> <p>The list of disks or partitions in the drop down box contains all the available disks or partitions that are <u>not</u>:</p> <ul style="list-style-type: none"> • already mounted • swap disks or partitions • LifeKeeper-protected disks or partitions <p>The drop down list will also filter out special disks or partitions, for example, root (/), boot (/boot), /proc, floppy and cdrom.</p> <p>Note: The size of the target disk or partition must be greater than or equal to that of the source disk or partition.</p>
DataKeeper Resource Tag	Select or enter the DataKeeper Resource Tag name.
Bitmap File	Select or edit the name of the bitmap file used for intent logging. If you choose none, then an intent log will not be used, and every resynchronization will be a full resync instead of a partial resync.
Replication Path	<p>Select the pair of local and remote IP addresses to use for replication between the target server and the other indicated server in the cluster. The valid paths and their associated IP addresses are derived from the set of LifeKeeper communication paths that have been defined for this same pair of servers. Due to the nature of DataKeeper, it is strongly recommended that you use a private (dedicated) network.</p> <p>If the DataKeeper Resource has previously been extended to one or more target servers, the extension to an additional server will loop through each of the pairings of the new target server with existing servers, prompting for a Replication Path for each pair.</p>
Replication Type	<p>Choose “synchronous” or “asynchronous” to indicate the type of replication that should be used between the indicated pair of servers.</p> <p>As for the previous Replication Path field, if the DataKeeper Resource has previously been extended to one or more target servers, the extension to an additional server will loop through each of the pairings of the new target server with existing servers, prompting for a Replication Type for each pair.</p>

2. Click **Extend** to continue. An information box will appear verifying that the extension is being performed.
3. Click **Finish** to confirm the successful extension of your DataKeeper resource instance.
4. Click **Done** to exit the **Extend Resources Hierarchy** menu selection.

Note: Be sure to test the functionality of the new instance on *all* servers by performing a manual switchover. See [Testing Your Resource Hierarchy](#) for details. At this point, SteelEye DataKeeper has initiated the data resynchronization from the source to the target disk or partition. In the LifeKeeper GUI, the state of the DataKeeper resource on the target server is set to “Resyncing”. Once the resynchronization is complete, the state will change to “Target” which is the normal Standby condition.

During resynchronization, the DataKeeper resource, and any resource that depends on it, will not be able to failover. This is to avoid data corruption.

Unextending Your Hierarchy

To remove a resource hierarchy from a single server in the LifeKeeper cluster, do the following:

1. On the **Edit** menu, select **Resource** then **Unextend Resource Hierarchy**.
2. Select the **Target Server** where you want to unextend the DataKeeper resource. It cannot be the server where the DataKeeper resource is currently in service (active).

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.

Click **Next**.

3. Select the **DataKeeper Hierarchy to Unextend** and click **Next**. (This dialog will not appear if you selected the **Unextend** task by right-clicking on a resource instance in either pane).
4. An information box appears confirming the target server and the DataKeeper resource hierarchy you have chosen to unextend. Click **Unextend**.
5. Another information box appears confirming that the DataKeeper resource was unextended successfully. Click **Done** to exit the **Unextend Resource Hierarchy** menu selection.

Note: At this point, data is not being replicated to the backup server.

Deleting a Resource Hierarchy

To delete a DataKeeper resource from all servers in your LifeKeeper configuration, complete the following steps.

Note: It is recommended that you take the DataKeeper resource out of service BEFORE deleting it. Otherwise, the **md** and **NetRAID** devices will not be removed, and you will have to unmount the file system manually. See [Taking a DataKeeper Resource Out of Service](#).

1. On the **Edit** menu, select **Resource**, then **Delete Resource Hierarchy**.
2. Select the name of the **TargetServer** where you will be deleting your DataKeeper resource

hierarchy.

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.

3. Select the **Hierarchy to Delete**. (This dialog will not appear if you selected the **Delete Resource** task by right-clicking on a resource instance in the left or right pane.) Click **Next**.
4. An information box appears confirming your selection of the target server and the hierarchy you have selected to delete. Click **Delete**.
5. Another information box appears confirming that the DataKeeper resource was deleted successfully. Click **Done** to exit.

Note: If the NetRAID device was mounted prior to the resource deletion then it will remain mounted. Otherwise, the NetRAID device will also be deleted.

Taking a DataKeeper Resource Out of Service

Taking a DataKeeper resource out of service removes LifeKeeper protection for the resource. It breaks the mirror, unmounts the file system (if applicable), stops the **md** device and kills the **nbd** server and client.

WARNING: Do not take your DataKeeper resource out of service unless you wish to stop mirroring your data and remove LifeKeeper protection. Use the **Pause** operation to temporarily stop mirroring.

1. In the right pane of the LifeKeeper GUI, right-click on the **DataKeeper resource** that is in service.
2. Click **Out of Service** from the resource popup menu.
3. A dialog box confirms the selected resource to be taken out of service. Any resource dependencies associated with the action are noted in the dialog. Click **Next**.
4. An information box appears showing the results of the resource being taken out of service. Click **Done**.

Bringing a DataKeeper Resource In Service

Bringing a DataKeeper resource in service is similar to creating the resource: LifeKeeper starts the **nbd** server and client, starts the **md** device which synchronizes the data between the source and target devices, and mounts the file system (if applicable).

1. Right-click on the **DataKeeper resource instance** from the right pane.
2. Click **In Service** from the popup menu. A dialog box appears confirming the server and resource that you have selected to bring into service. Click **In Service** to bring the resource into service.
3. An information box shows the results of the resource being brought into service. Any resource dependencies associated with the action are noted in the confirmation dialog. Click **Done**.

Testing Your Resource Hierarchy

You can test your DataKeeper resource hierarchy by initiating a manual switchover. This will simulate a failover of the resource instance from the primary server to the backup server.

Performing a Manual Switchover from the LifeKeeper GUI

You can initiate a manual switchover from the LifeKeeper GUI by selecting **Edit, Resource, and InService**. For example, an in-service request executed on a backup server causes the DataKeeper resource hierarchy to be taken out-of-service on the primary server and placed in-service on the backup server. At this point, the original backup server is now the primary server and original primary server has now become the backup server.

After the switchover, the state of the DataKeeper resource on the target server is set to **“Resyncing”** in the LifeKeeper GUI. Once the resynchronization is complete the state will change to **“Target”**, which is the normal **Standby** condition.

Note: Manual failover is prevented for DataKeeper resources during resynchronization.

If you execute the **Out of Service** request, the resource hierarchy is taken out of service without bringing it in service on the other server. The resource can only be brought in service on the same server if it was taken out of service during resynchronization.

Chapter 3: Administration

Administering SteelEye DataKeeper for Linux

The following topics provide information to help in understanding and managing SteelEye DataKeeper for Linux operations and issues after DataKeeper resources are created.

[Viewing Mirror Status](#)

[GUI Mirror Administration](#)

[Pause and Resume Mirror](#)

[Force Mirror Online](#)

[Set Rewind Log Location](#)

[Set Rewind Log Max Size](#)

[Create/View Rewind Bookmarks](#)

[Rewind and Recover Data](#)

[Command Line Mirror Administration](#)

[Monitoring Mirror Status via Command Line](#)

[Server Failure](#)

[Resynchronization](#)

[Avoiding Full Resynchronizations](#)

Viewing Mirror Status

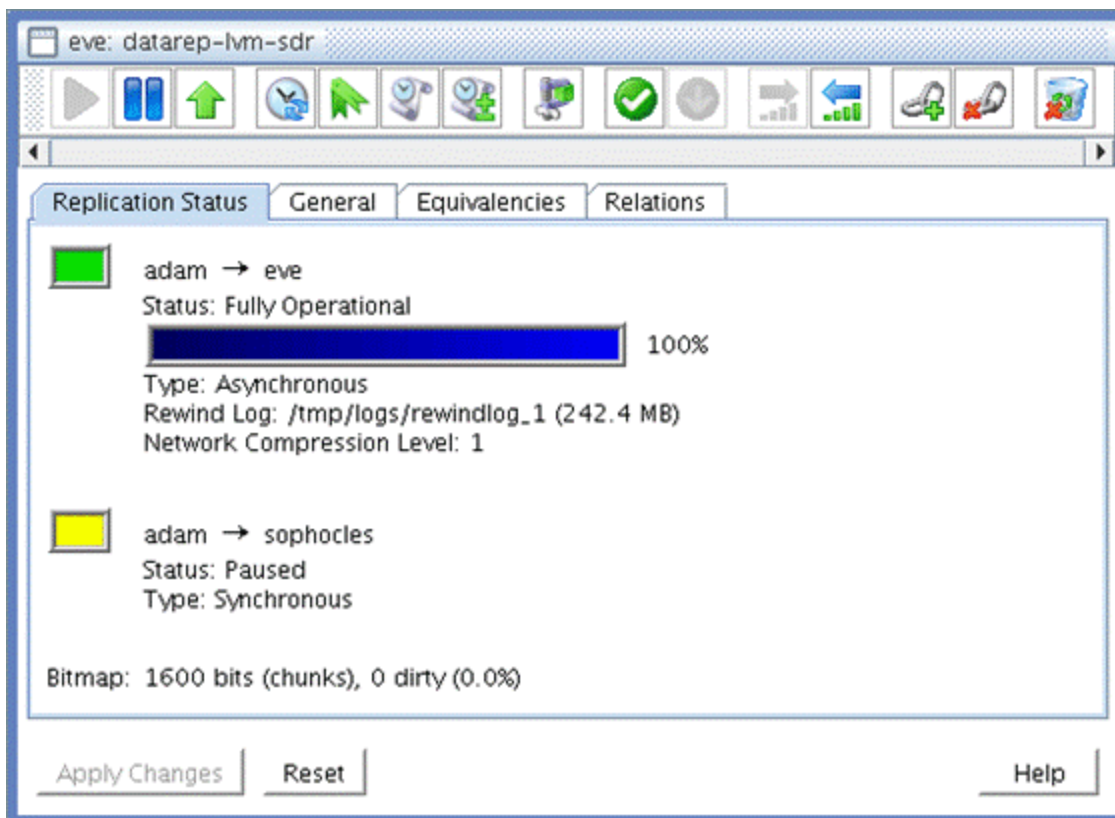
You can view the **Replication Status** dialog to see the following information about your mirror:

- **Mirror status:** Fully Operational, Paused, Resyncing, or Out Of Sync
- **Synchronization status:** percent complete
- **Replication type:** synchronous or asynchronous
- **Replication direction:** from source server to target server
- **Bitmap:** the state of the bitmap/intent log

- **Rewind Log**: the location and size of the rewind log (if enabled)
- **Network Compression Level**: the compression level (if enabled)

To view the **Replication Status** dialog, do the following:

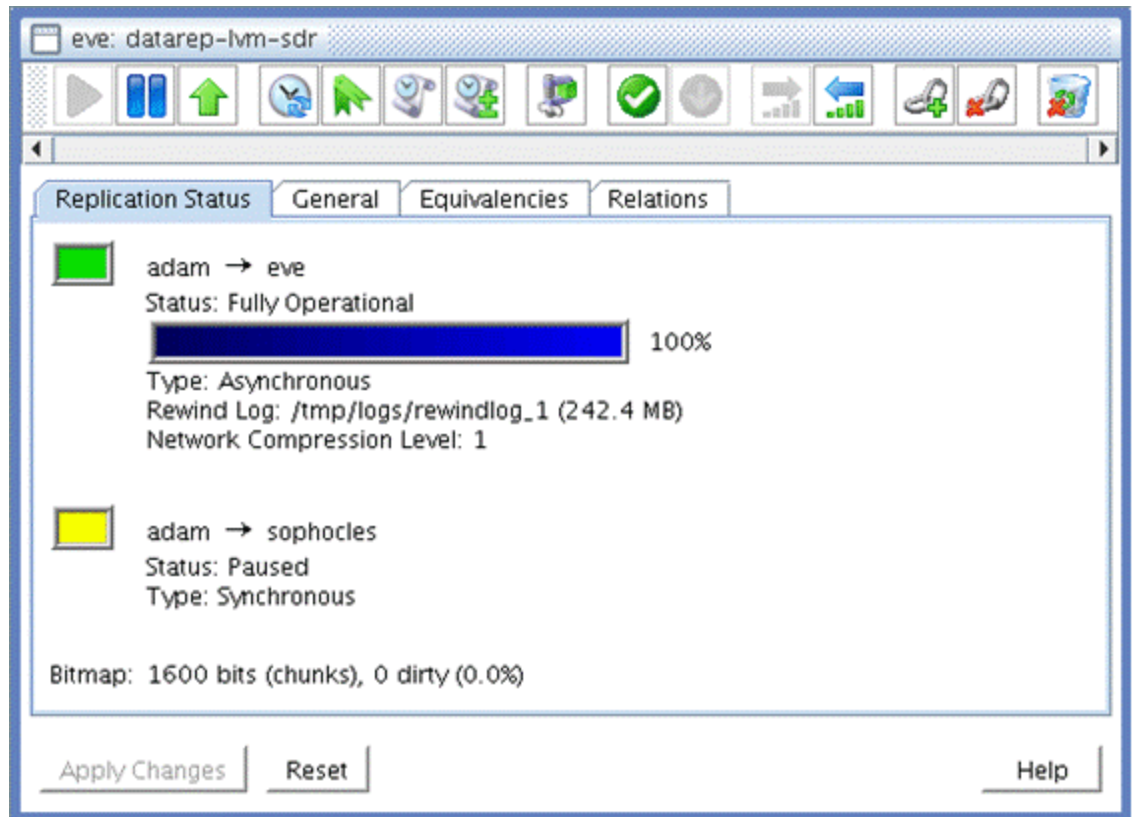
1. Click the **View** menu, and select **Properties Panel**.
 2. Click the **DataKeeper resource** in the **LifeKeeper status** display.
- or,
1. Right-click the **DataKeeper resource** in the LifeKeeper status display.
 2. From the pop-up menu, select **Properties**.



GUI Mirror Administration

A SteelEye DataKeeper mirror can be administered through the LifeKeeper GUI in two ways:

1. By enabling the **Properties Panel** and clicking the toolbar icons (shown in the screenshot).



Click on each icon below for a description



or,

2. By right-clicking the **data replication resource** and selecting an action from the popup menu.

Pause and Resume

Pause Mirror



Resume Mirror



You may pause a mirror to temporarily stop all writes from being replicated to the target disk. For example, you might pause the mirror to take a snapshot of the target disk or to increase I/O performance on the source system during peak traffic times.

When the mirror is paused, it will be mounted for read (or read/write with kernel 2.6.19 or higher) access at the normal filesystem mount point on the target system. Any data written to the target while the mirror is paused will be overwritten when the mirror is resumed.

Force Mirror Online



Force Mirror Online should be used only in the event that both servers have become inoperable and the primary server cannot bring the resource in service after rebooting. Selecting **Force Mirror Online** removes the *data_corrupt* flag and brings the DataKeeper resource in service. For more information, see Primary server cannot bring the resource ISP in the [Troubleshooting](#) section.

Note: `Mirror_settings` should be run on the target system(s) (or on all systems, if you want the settings to take effect regardless of which system becomes the mirror source). The mirror must be **paused** and **restarted** before any settings changes will take effect.

Set Compression Level



The Network Compression Level may be set to a value from 0 to 9. A value of 0 disables compression entirely. Level 1 is the fastest but least aggressive compression level, while Level 9 is the slowest but best. Network compression is typically effective only on WANs.

Set Rewind Log Location



Select the directory where the rewind log file should be stored (this is only applicable when the system is a mirror target). There should be adequate space in this location to store the desired

amount of history in the log¹. The log cannot be located on a mirror or shared disk and should, for optimal performance, be located on a separate physical disk from any mirrors. An empty setting disables rewind logging.

Note: The mirror must be paused and restarted before any setting changes will take effect.

¹The log file contains a copy of every disk block that is written to the mirrored disk so the log file can grow larger than the mirrored disk itself if the same disk blocks are written multiple times, as is the case when a file is modified or appended to.

Set Rewind Log Max Size



Enter the maximum log file size in megabytes (MB). An empty value or zero (0) disables the file size limit. There should be adequate space on the log file disk to accommodate the log file growing to the maximum size. However, the log will wrap around and overwrite the earliest entries when it detects that it has run out of disk space.

Create and View Rewind Bookmarks



A bookmark is an entry that is placed in the rewind log file. Bookmarks are useful for keeping track of important system events (such as upgrades) in case a rewind needs to be performed. When you perform a rewind, all bookmarked log entries will be displayed as choices for the rewind point.

Rewind and Recover Data



The rewind feature allows the data on the target disk to be rewound back to any previous disk write.

The steps involved are:

1. The mirror is paused.
2. A timestamp associated with previous disk write is selected and the disk is rewound to that time.

Rewind and Recover Data

3. The user is asked to verify the rewind data and indicate its condition (good or bad).
4. The user then has the option to use the current data (go to Step 5) or continue rewinding by selecting another timestamp (go to Step 2).
5. The user has the choice of recovering the data manually and then resuming the mirror (erasing the rewind data) or switching the mirror and any protected applications to the target system and using the rewind data as the new production data.

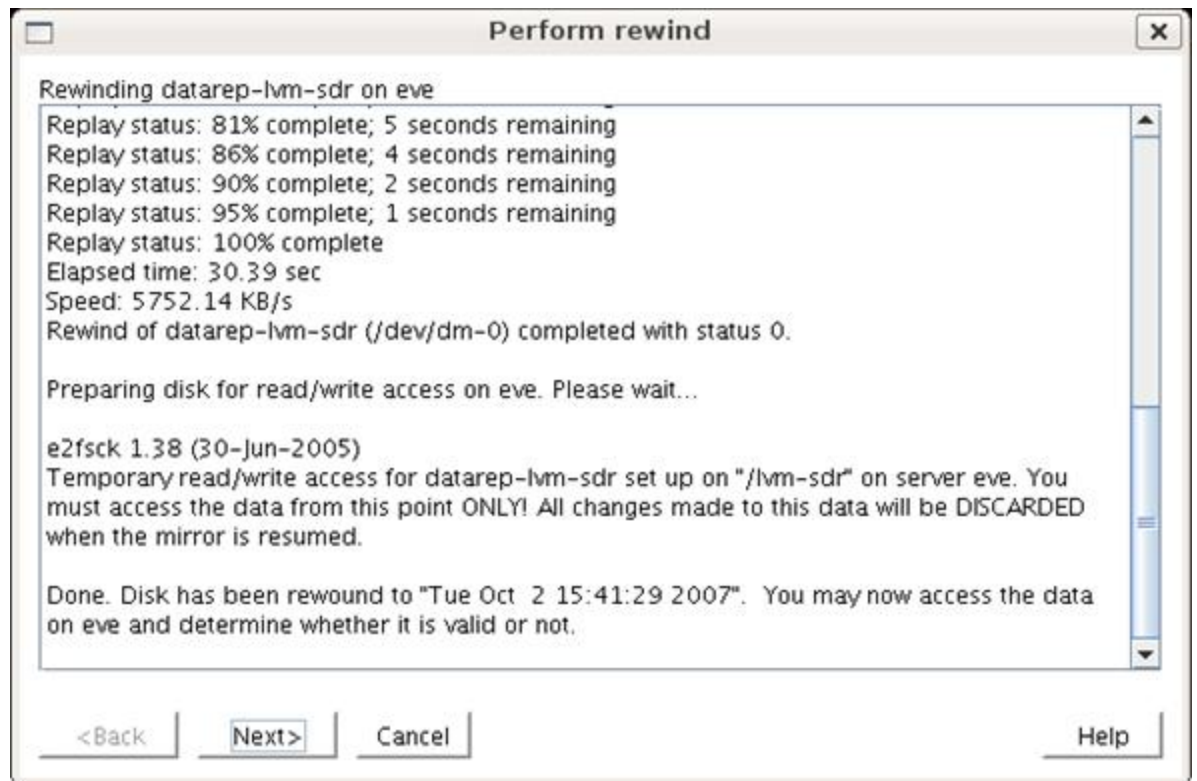
The user is led through the steps above with a series of wizard dialogs. The dialogs are explained below:

1. Confirm that you wish to rewind the data. Click **Continue**.
2. The mirror is being paused in preparation for the rewind. Click **Next**.
3. Select or type in a timestamp that you wish to rewind to. Bookmarked log entries as well as a random sampling of other log entries appear in the dropdown list. The progress bar at the bottom of the dialog displays which data is good (green), bad (red) or unknown (yellow). So, at the beginning of the rewind process, the progress bar is all yellow. Once the data has been rewind and you have indicated that the data is good or bad, the progress bar is updated with green and red sections accordingly.



Dialog 3

4. The data is being rewound. After the data is rewound, the target disk is mounted for read-only access so that the data can be verified. Click **Next**.



Dialog 4

5. You are asked for comments on the data. Enter any comments (not mandatory) and click **Next**.
6. You are now asked to indicate whether the data is valid or not. Answer **Yes** or **No** and click **Next**.
7. You are now asked if you wish to continue rewinding (go back to Dialog 3) or accept the current rewound data and begin recovery (go on to Dialog 8).
8. You are now asked to choose a recovery method. The choices are:
 - a. **Move applications to <target system>** (go on to Dialog 9)
 - b. **Manually copy data to the source system** (go on to Dialog 10)
 - c. Make your selection and click **Next**.
9. The hierarchy is now being switched over to the target server. The rewound data will be resynced to the old source disk. Click **Finish**. *Rewind is complete.*
10. You are asked to manually copy files to the source system. Copy any rewound data that you

wish to keep to a safe location, then click **Next**.

11. The mirror is now being resumed. A full resynchronization will occur from the source to target. Click **Finish**. *Rewind is complete*.

Command Line Mirror Administration

In addition to performing actions through the LifeKeeper GUI, the mirror can also be administered using the command line. There are several commands (found in the `$LKROOT/bin` directory) that can be used to administer a DataKeeper resource.

Mirror Actions

```
mirror_action <tag> <action> <source> [target(s)]
```

<tag> is the LifeKeeper resource tag of the DataKeeper resource

<action> is one of: pause, resume, force, fullresync

<source> is the current source system

<target> is the target system (or list of systems) that the action should affect

Examples:

To pause the mirror named `datarep-ext3` from source system, `adam`, to target system, `eve`:

```
mirror_action datarep-ext3 pause adam eve
```

To resume replication from `adam` to both `eve` and `sophocles`:

```
mirror_action datarep-ext3 resume adam eve sophocles
```

To force the mirror online on system `eve`:

```
mirror_action datarep-ext3 force eve
```

To resume replication and force a full resynchronization from `adam` to `sophocles`:

```
mirror_action datarep-ext3 fullresync adam sophocles
```

Mirror Settings

```
mirror_settings <tag> <setting> <value>
```

<tag> is the LifeKeeper resource tag of the DataKeeper resource

<setting> is one of: `logdir`, `logmax`, `compress`

<value> is the value to be set

Note: `mirror_settings` should be run on the target system(s) (or on all systems, if you want the settings to take effect regardless of which system becomes the mirror source). The mirror must be paused and restarted before any settings changes will take effect.

Examples:

To set the network compression level to 5:

```
mirror_settings datarep-ext3 compress 5
```

To disable network compression:

```
mirror_settings datarep-ext3 compress 0
```

To set the rewind logging directory (and enable rewind logging):

```
mirror_settings datarep-ext3 logdir /tmp/logs
```

To disable rewind logging:

```
mirror_settings datarep-ext3 logdir ""
```

To set the rewind log maximum size to 1GB:

```
mirror_settings datarep-ext3 logmax 1073741824
```

To disable the rewind log maximum size limit:

```
mirror_settings datarep-ext3 logmax 0
```

Bitmap Administration

```
bitmap -a <num>|-c|-d|-X <bitmap_file>
```

-a <num> adds the asynchronous write parameter to the bitmap file. It is needed if a synchronous mirror is upgraded to include an asynchronous target. The default value for <num> is 256.

-c cleans the bitmap file (zeroes all the bits). This can be used to avoid a full resync in case an exact replica of the source disk exists on the target. Use this option with extreme caution.

-d dirties the bitmap file (sets all the bits to ones). This option can be used to force a full resync, for example after a split-brain situation has occurred.

-X<bitmap file> examines the bitmap file and displays useful information about the bitmap and the mirror.

In addition, the `mdadm` command may also be used to administer a DataKeeper resource, as the DataKeeper resource is actually an md device. Refer to the `mdadm(8)` man page for details. **Note:**

When using `mdadm`, be sure to use the version that is located in `$LKROOT/bin`, as it is more up-to-date than the version included with the operating system.

Monitoring Mirror Status via Command Line

Normally, the mirror status can be checked using the **Replication Status** tab in the **Resource Properties** dialog of the LifeKeeper GUI. However, you may also monitor the status of your mirror by executing:

```
$LKROOT/bin/mirror_status <tag>
```

Example:

```
# mirror_status datarep-ext3-sdr
```

```
[-] eve -> adam
```

```
    Status: Paused
```

```
    Type: Asynchronous
```

```
[-] eve -> sophocles
```

```
    Status: Resynchronizing
```

```
    [=> ] 11%
```

```
    Resync Speed: 1573K/sec
```

```
    Type: Synchronous
```

```
Bitmap: 4895 bits (chunks), 4895 dirty (100.0%)
```

The following command may also be helpful:

```
cat /proc/mdstat
```

A sample *mdstat* file is shown below:

```
eve:~ # cat /proc/mdstat
```

```
Personalities : [raid1]
```

```
md1 : active raid1 nbd10[1] nbd8[3](F) sdb1[0]
```

```
    313236 blocks super non-persistent [3/2] [UU_]
```

```
    bitmap: 3/3 pages [12KB], 64KB chunk, file:  
    /opt/LifeKeeper/bitmap_ext3-sdr
```

```
unused devices: <none/></tag>
```

Server Failure

If both your primary and backup servers become inoperable, your DataKeeper resource will be brought into service/activated only when **both** servers are functional again. This is to avoid data corruption that could result from initiating the resynchronization in the wrong direction. If you are certain that the only operable server was the last server on which the resource was “**In Service Protected**” (ISP), then you can force it online by right-clicking the DataKeeper resource and then selecting **Force Mirror Online**.

Resynchronization

During the resynchronization of a DataKeeper resource, the state of this resource instance on the target server is “**Resyncing**”. However, the resource instance is “**Source**” (ISP) on the primary server. The LifeKeeper GUI reflects this status by representing the DataKeeper resource on the target server with the following icon:



and the DataKeeper resource on the primary server with this icon:



As soon as the resynchronization is complete, the resource state on the target becomes “**Target**” and the icon changes to the following:



The following points should be noted about the resynchronization process:

- A SteelEye DataKeeper resource and its parent resources cannot fail over to a target that was in the synchronization process when the primary failed.
- If your DataKeeper resource is taken out of service/deactivated during the synchronization of a target server, that resource can only be brought back into service/activated on the same system or on another target that is already in sync (if multiple targets exist), and the resynchronization will continue.
- If your primary server becomes inoperable during the synchronization process, any target server that is in the synchronization process will not be able to bring your DataKeeper resource into service. Once your primary server becomes functional again, a resynchronization of the mirror will continue.

Avoiding Full Resynchronizations

When replicating large amounts of data over a WAN link, it is desirable to avoid full resynchronizations which can consume large amounts of network bandwidth and time. With newer kernels, SteelEye DataKeeper can avoid almost all full resyncs by using its bitmap technology. However, the initial full resync, which occurs when the mirror is first set up, cannot be avoided when existing data is being replicated. (For brand new data, SteelEye does not perform a full resync, so the steps below are not necessary.)

There are a couple of ways to avoid an initial full resync when replicating existing data. Two recommended methods are described below.

Method 1

The first method consists of taking a raw disk image and shipping it to the target site. This results in minimal downtime as the mirror can be active on the source system while the data is in transit to the target system.

Procedure

1. Create the mirror (selecting Replicate Existing Filesystem), but do not extend the mirror to the target system.
2. Take the mirror out of service.
3. Take an image of the source disk or partition. For this example, the chosen disk or partition is /dev/sda1:

```
root@source# dd if=/dev/sda1 of=/tmp/sdr_disk.img bs=65536
```

(The block size argument of 65536 is merely for efficiency).

This will create a file containing the raw disk image of the disk or partition.

Note that instead of a file, a hard drive or other storage device could have been used.

4. Optional Step – Take a checksum of the source disk or partition:

```
root@source# md5sum /dev/sda1
```

5. Optional Step – Compress the disk image file:

```
root@source# gzip /tmp/sdr_disk.img
```

6. Clear the bitmap file, e.g.:

```
root@source# /opt/LifeKeeper/bin/bitmap -c
/opt/LifeKeeper/bitmap_sdr
```

7. Bring the mirror and dependent filesystem and applications (if any), into service. The bitmap file will track any changes made while the data is transferred to the target system.
8. Transfer the disk image to the target system using your preferred transfer method.
9. Optional Step – Uncompress the disk image file on the target system:

```
root@target# gunzip /tmp/sdr_disk.img.gz
```

10. Optional Step – Verify that the checksum of the image file matches the original checksum taken in Step 4:

```
root@target# md5sum /tmp/sdr_disk.img
```

11. Transfer the image to the target disk, for example, /dev/sda2:

```
root@target# dd if=/tmp/sdr_disk.img of=/dev/sda2 bs=65536
```

12. Set LKDR_NOFULL_SYNC=1 in /etc/default/LifeKeeper on both systems:

```
root@source# echo 'LKDR_NO_FULL_SYNC=1' >>
/etc/default/LifeKeeper
```

```
root@target# echo 'LKDR_NO_FULL_SYNC=1' >>
/etc/default/LifeKeeper
```

13. Extend the mirror to the target. A partial resync will occur.

Method 2

This method can be used if the target system can be easily transported to or will already be at the source site when the systems are configured. This method consists of temporarily modifying network routes to make the eventual WAN mirror into a LAN mirror so that the initial full resync can be performed over a faster local network. In the following example, assume the source site is on subnet 10.10.10.0/24 and the target site is on subnet 10.10.20.0/24. By temporarily setting up static routes on the source and target systems, the "WAN" traffic can be made to go directly from one server to another over a local ethernet connection or loopback cable.

Procedure

1. Install and configure the systems at the source site.
2. Add static routes:

```
root@source# route add -net 10.10.20.0/24 dev eth0
```

```
root@target# route add -net 10.10.10.0/24 dev eth0
```

The systems should now be able to talk to each other over the LAN.

Procedure

3. Configure the communication paths in LifeKeeper.
4. Create the mirror and extend to the target. A full resync will occur.
5. Pause the mirror. Changes will be tracked in the bitmap file until the mirror is resumed.
6. Delete the static routes:

```
root@source# route del -net 10.10.20.0/24
```

```
root@target# route del -net 10.10.10.0/24
```

7. Shut down the target system and ship it to its permanent location.
8. Boot the target system and ensure network connectivity with the source.
9. Resume the mirror. A partial resync will occur.

Chapter 4: Multi-Site Cluster

SteelEye Protection Suite for Linux Multi-Site Cluster

The SteelEye Protection Suite for Linux Multi-Site Cluster is a separately licensed product that uses a LifeKeeper shared storage configuration between two or more servers with the additional ability to replicate the shared disk(s) to one or more target servers using SteelEye DataKeeper for Linux.

Please see the following topics for information on multi-site clusters:

Overview

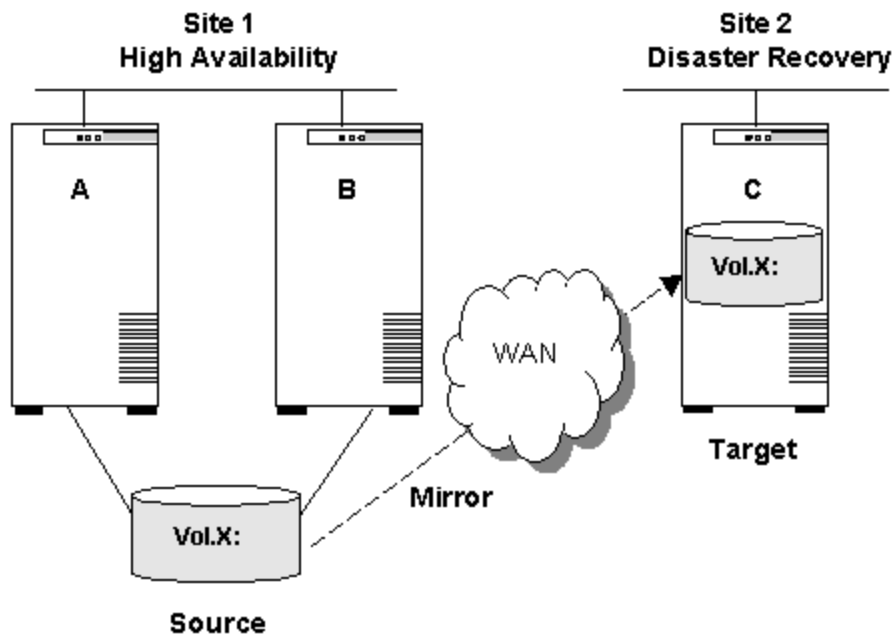
- [Multi-Site Cluster Overview](#)
- [Multi-Site Cluster Configuration Considerations](#)
- [Creating a SteelEye Protection Suite for Linux Multi-Site Cluster Resource Hierarchy](#)
 - [Replicate New File System](#)
 - [Replicate Existing File System](#)
 - [DataKeeper Resource](#)
- [Extending Your Hierarchy](#)
- [Extending a Hierarchy to a Disaster Recovery System](#)
- [Configuring the Restore and Recovery Setting for your IP Resource](#)
- [Troubleshooting](#)

Migrating to a Multi-Site Cluster Environment

- [Migrating to a Multi-Site Cluster Environment](#)
 - [Requirements](#)
 - [Before You Start](#)
 - [Performing the Migration](#)
 - [Successful Migration](#)

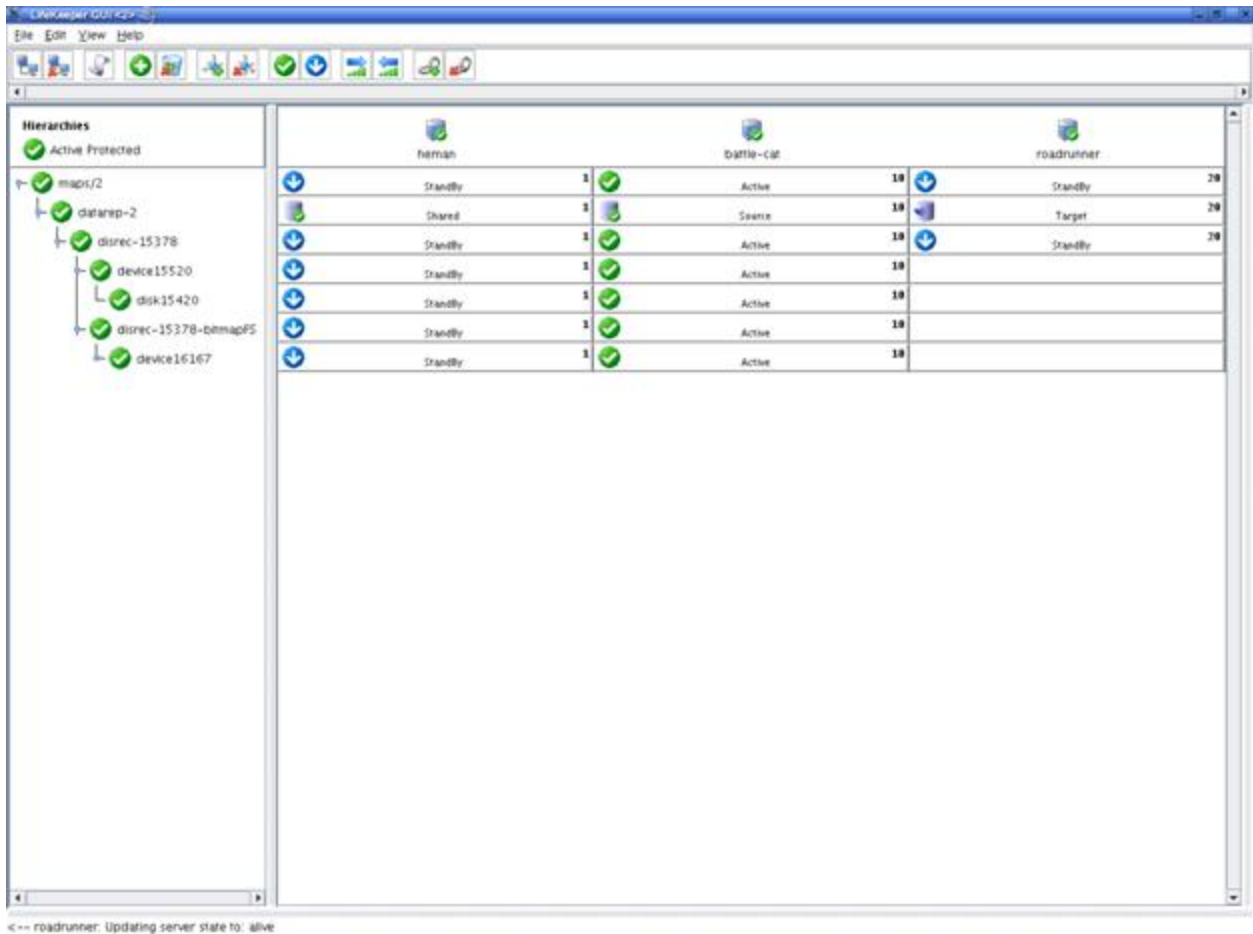
SteelEye Protection Suite for Linux Multi-Site Cluster

The **SteelEye Protection Suite for Linux Multi-Site Cluster** is a separately licensed product that uses a LifeKeeper shared storage configuration between two or more servers with the additional ability to replicate the shared disk(s) to one or more target servers using SteelEye DataKeeper.



SteelEye Protection Suite for Linux Multi-Site Cluster can be built upon a Wide Area Network that is configured to provide failover of IP addresses across multiple network segments that are on different subnets. This configuration involves either a virtual network (Virtual LAN (VLAN)) or Virtual Private Network (VPN).

Following is an image of the SteelEye LifeKeeper GUI after the SteelEye Protection Suite for Linux Multi-Site Cluster product has been configured. Although the hierarchies appear unbalanced, they are configured properly and will function correctly. If you are an existing SteelEye DataKeeper customer and are familiar with the SteelEye LifeKeeper graphical user interface, the SteelEye Protection Suite Multi-Site Cluster resource hierarchy display in the LifeKeeper GUI will appear differently from previous releases of SteelEye DataKeeper.



Multi-Site Cluster Configuration Considerations

Before you begin configuring your systems, it's important to understand what hierarchy configurations you should avoid in the Linux Multi-Site Cluster hierarchy environment.

Below are three examples of hierarchy configurations that should be avoided in the Linux Multi-Site Cluster environment. In all these cases, a Linux Multi-Site Cluster hierarchy shares an underlying device with another hierarchy. Failure or switchover of either hierarchy will impact the associated hierarchy. This could possibly produce unintended consequences such as application failure or mirror breakage; which would require a full-resync process later. In addition, complications could result when switching the mirror sources to the DR site allowing it to mirror back to the primary site since the mirror target system will have the lower level disk resources in service. Any shared resources must also be operational (ISP) on the same node as the mirror target.

Example	Description
1	Using the Multi-Site Cluster hierarchy's mirror disk resource more than once in the same or different hierarchies.
2	Using the same Multi-Site Cluster file system or disk resource for the mirror bitmap in more than one Multi-Site Cluster hierarchy. (Each mirror's bitmap file must reside on a unique LUN and can't be shared.)
3	Using the bitmap file system, device or disk resource with another hierarchy (Multi-Site or non-Multi-Site).

Multi-Site Cluster Restrictions

- The SteelEye Logical Volume Manager Recovery Kit should not be installed on the Disaster Recovery node when using Linux Multi-Site Cluster.

Creating a SteelEye Protection Suite for Linux Multi-Site Cluster Resource Hierarchy

Perform the following on your primary server:

1. Select **Edit > Server > Create Resource Hierarchy**

The **Create Resource Wizard** dialog will appear.

2. Select the **Data Replication** option from the drop down list and click **Next** to continue.
3. You will be prompted for the following information. When the **Back** button is active in any of the dialog boxes, you can go back to the previous dialog box. This is helpful should you encounter any error requiring you to correct the previously entered information. You may click **Cancel** at any time to cancel the entire creation process.

Field	Tips
Switchback Type	<p>You must select intelligent switchback. This means that after a failover to the backup server, an administrator must manually switch the Multi-Site Cluster resource back to the primary server.</p> <p>CAUTION: This release of SteelEye DataKeeper does not support Automatic Switchback for DataKeeper resources. Additionally, the Automatic Switchback restriction is applicable for any other LifeKeeper resource that becomes part of the Multi-Site Cluster hierarchy. This includes anything sitting above the hierarchy or becomes a child within the hierarchy.</p>
Server	Select the name of the server where the NetRAID device will be created (typically this is your primary server). All servers in your cluster are included in the drop down list box.
Hierarchy Type	<p>Choose the data replication type you wish to create by selecting one of the following:</p> <ul style="list-style-type: none"> • Replicate New File System • Replicate Existing File System • DataKeeper Resource

The next sequence of dialog boxes depends on which **Hierarchy Type** you have chosen. While some of the dialog boxes may be the same for each Hierarchy Type, their sequence and the required information may be slightly different. The following three topics will take you through the remainder of the Hierarchy creation process:

- [Replicate New File System](#)
- [Replicate Existing File System](#)
- [DataKeeper Resource](#)

Replicate New File System

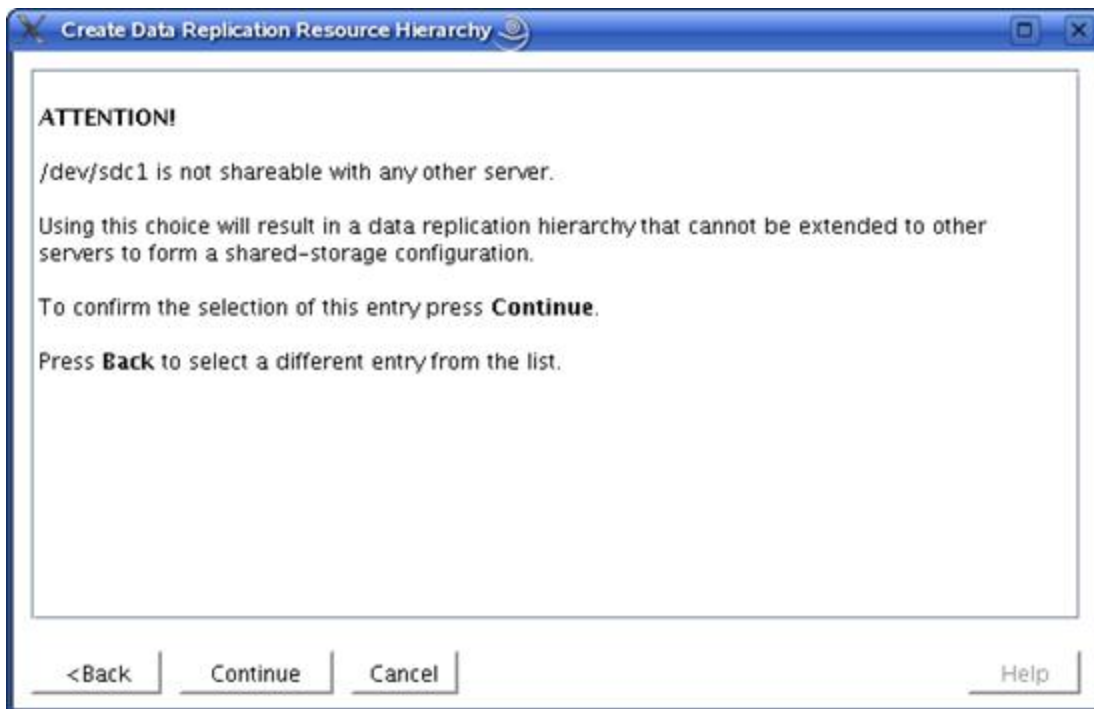
This option will create a NetRAID device, format it with a LifeKeeper supported file system type, mount the file system on the NetRAID device and place both the mounted file system and the NetRAID device under LifeKeeper protection. The NetRAID device and the local disk or partition will be formatted causing existing data to be deleted. You should select this option if you want to create a mirror on a new file system and place it under LifeKeeper protection. You will need one free disk or partition for this resource type.

CAUTION: This option will cause your local disk or partition to be formatted and all existing data will be deleted.

1. Enter the following information when prompted:

Field	Tips
Source Disk or Partition	<p>The list of Source Disks or Partitions in the drop down box contains all the available disks or partitions that are <u>not</u>:</p> <ul style="list-style-type: none"> • currently mounted • swap disks or partitions • LifeKeeper-protected disks or partitions <p>The drop down list will also filter out special disks or partitions, for example, root (/), boot (/boot), /proc, floppy and cdrom.</p>

2. The following screen will display if you select a source disk or partition that is not shared.



3. Select **Back** to select a different source disk or partition that is shared. Provide the remaining information to finish configuring the SteelEye Protection Suite for Linux Multi-Site Cluster resource

Field	Tips
New Mount Point	Enter the New Mount Point of the new file system. This should be the mount point where the replicated disk or partition will be located.
New File System Type	Select the File System Type . You may only choose from the LifeKeeper supported file system types.
DataKeeper Resource Tag	Select or enter a unique DataKeeper Resource Tag name for the DataKeeper resource instance.
File System Resource Tag	Select or enter the File System Resource Tag name for the file system resource instance.
Bitmap File	Select the bitmap file entry from the pull down list. Displayed in the list are all available shared file systems that can be used to hold the bitmap file. The bitmap file must be placed on a shared device that can switch between the local nodes in the cluster.

- Click **Next** to continue to the **Confirmation** Screen.
- A confirmation screen noting the location where the new file system will be created and a warning indicating the pending reformat of the local disk or partition will display. Click **Create** to begin **Resource Creation**.
- LifeKeeper will verify that you have provided valid data to create your resource on a new file system. If LifeKeeper detects a problem, an ERROR will appear in the information box. If the validation is successful, your resource will be created. Note that the creation of the file system may take several minutes depending upon the disk or partition size.

Click **Next** to continue.

- An information box appears announcing the successful creation of your new replicated file system resource hierarchy. You must **Extend** the hierarchy to another server in your cluster to begin data replication and in order to place it under LifeKeeper protection.

Click **Next** to extend the resource or click **Cancel** if you wish to extend your resource at another time.

If you click **Continue** LifeKeeper will launch the **Pre-extend Wizard**. Refer to Step 2 under [Extending Your Hierarchy](#) for details on how to extend your resource hierarchy to another server.

Replicate Existing File System

This option will unmount a currently mounted file system on a local disk or partition, create a NetRAID device, then re-mount the file system on the NetRAID device. Both the NetRAID device and the mounted file system are placed under LifeKeeper protection. You should select this option if

you want to create a mirror on an existing file system and place it under LifeKeeper protection.

1. Enter the following information when prompted:

Field	Tips
Existing Mount Point	This should be the mount point to be mounted on the NetRAID device on the primary server. The local disk or partition should already be mounted on this mount point.

2. The following screen will display if you select a mount point that is not shared.

```
Deleting resource hierarchy...
Successfully removed
ins_remove[701,lraci.C]Thu Jun 1 07:06:54 EDT 2000:
    fletch,priv_globact(1,delete): Running Post Global delete
    Machine cornfed
ins_remove[714,lraci.C]Thu Jun 1 07:06:56 EDT 2000:
    fletch,priv_globact(1,delete): Post Global delete Scripts F
    Exiting 0 On Machine cornfed With Output Following:
lcdrecover[701,lraci.C]Thu Jun 1 07:12:15 EDT 2000:
```

3. Select **Back** to select a different mount point that is shared. Provide the remaining information to finish configuring the SteelEye Protection Suite for Linux Multi-Site Cluster resource.

Field	Tips
DataKeeper Resource Tag	Select or enter a unique DataKeeper Resource Tag name for the DataKeeper resource instance.
File System Resource Tag	Select or enter the File System Resource Tag name.
Bitmap File	Select the bitmap file entry from the pull down list. Displayed in the list are all available shared file systems that can be used to hold the bitmap file. The bitmap file must be placed on a shared device that can switch between the local nodes in the cluster.

4. Click **Next** to create your DataKeeper resource on the primary server.
5. LifeKeeper will verify that you have provided valid data to create your DataKeeper resource. If LifeKeeper detects a problem, an ERROR will appear in the information box. If the validation is successful, your resource will be created. Click **Next**.

- An information box appears announcing that you have successfully created an existing replicated file system resource hierarchy. You must **Extend** the hierarchy to another server in your cluster to begin replication and to place it under LifeKeeper protection. Click **Next** to extend the resource, or click **Cancel** if you wish to extend your resource at another time.

If you click **Continue** LifeKeeper will launch the *Pre-extend Wizard*. Refer to Step 2 under Extending Your Hierarchy for details on how to extend your resource hierarchy to another server.

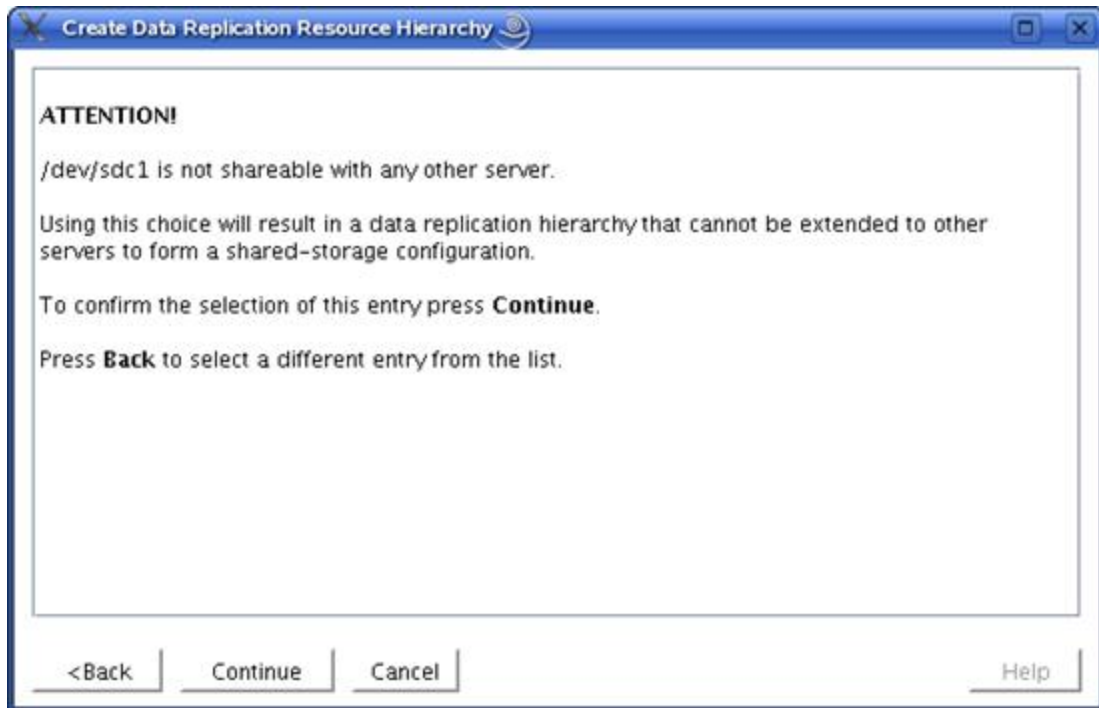
DataKeeper Resource

This option will create only the NetRAID device (not a file system) and place the device under LifeKeeper protection. You should select this option if you only want to create a DataKeeper device on a disk or partition and place the device under LifeKeeper protection. You will need to manually make and mount a file system on this device in order to create a readable mirror. You will need one free disk or partition for this resource type.

- Enter the following information when prompted:

Field	Tips
Source Disk or Partition	<p>The list of Source Disks or Partitions in the drop down box contains all the available disks or partitions that are <u>not</u>:</p> <ul style="list-style-type: none"> • currently mounted • swap type disks or partitions • LifeKeeper-protected disks or partitions <p>The drop down list will also filter out special disks or partitions, for example, root (/), boot (/boot), /proc, floppy and cdrom.</p>

- The following screen will display if you select a source disk or partition that is not shared.



3. Select **Back** to select a different source disk or partition that is shared. Provide the remaining information to finish configuring the SteelEye Protection Suite for Linux Multi-Site Cluster resource.

Field	Tips
DataKeeper Resource Tag	Select or enter a unique DataKeeper Resource Tag name for the DataKeeper resource instance.
Bitmap File	Select the bitmap file entry from the pull down list. Displayed in the list are all available shared file systems that can be used to hold the bitmap file. The bitmap file must be placed on a shared device that can switch between the local nodes in the cluster.

4. Click **Next**.
5. An information window appears notifying you that you will have to manually make the file system and mount the NetRAID device (*/dev/mdX*) before being able to use it.
Click **Create** to create your DataKeeper device on the local disk or partition.
6. An information box appears and LifeKeeper will verify that you have provided valid data to create your DataKeeper resource. If LifeKeeper detects a problem, an ERROR will appear in the information box. If the validation is successful, your resource will be created.
Click **Next** to continue.

7. An information box appears announcing the successful creation of your DataKeeper resource device. You must **Extend** the hierarchy to another server in your cluster to begin data replication and in order to place it on the backup/target server and under LifeKeeper protection.

Click **Continue** to extend the resource, or click **Cancel** if you wish to extend your resource at another time.

If you click **Continue** LifeKeeper will launch the ***Pre-extend Wizard***. Refer to Step 2 under [Extending Your Hierarchy](#) for details on how to extend your resource hierarchy to another server.

Extending Your Hierarchy

This operation should be started on the Primary Server to the Secondary Server from the **Edit** menu or initiated automatically upon completing the **Create Resource Hierarchy** option in which case you should refer to Step 2 below.

1. On the **Edit** menu, select **Resource** then **Extend Resource Hierarchy**. The ***Pre-Extend Wizard*** appears. If you are unfamiliar with the **Extend** operation, click **Next**. If you are familiar with the LifeKeeper **Extend Resource Hierarchy** defaults and want to bypass the prompts for input/confirmation, click **Accept Defaults**.
2. The ***Pre-Extend Wizard*** will prompt you to enter the following information.

Note: The first two fields appear only if you initiated the Extend from the **Edit** menu.

Field	Tips
Template Server	<p>Select the Template Server where your DataKeeper 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 (activated) resource hierarchy.</p> <p>An error message will appear if you select a resource tag that is not in service on the template server you have selected. The drop down box in this dialog provides the names of all the servers in your cluster.</p>
Tag to Extend	<p>This is the name of the DataKeeper instance you wish to extend from the template server to the target server. The drop down box will list all the resources that you have created on the template server.</p>
Target Server	<p>Enter or select the server you are extending to.</p>
Switchback Type	<p>You must select intelligent switchback. This means that after a failover to the backup server, an administrator must manually switch the Multi-Site Cluster hierarchy resource back to the primary server.</p> <p>CAUTION: This release of DataKeeper for Linux does not support Automatic Switchback for DataKeeper resources. Additionally, the Automatic Switchback restriction is applicable for any other LifeKeeper resource that becomes part of the Multi-Site Cluster hierarchy. This includes anything sitting above the hierarchy or becomes a child within the hierarchy.</p>
Template Priority	<p>Select or enter a Template Priority. This is the priority for the DataKeeper 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.</p> <p>Note: This selection will appear only for the initial extend of the hierarchy.</p>
Target Priority	<p>Select or enter the Target Priority. This is the priority for the new extended DataKeeper 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.</p>

3. After receiving the message that the pre-extend checks were successful, click .
4. Depending upon the hierarchy being extended, LifeKeeper will display a series of information boxes showing the Resource Tags to be extended, some of which cannot be edited. Click **Next** to launch the Extend Resource Hierarchy configuration task.

The next section lists the steps required to complete the extension of a DataKeeper resource to another server.

Extending a DataKeeper Resource

1. After you have been notified that your pre-extend script has executed successfully, you will be prompted for the following information:

Field	Tips
Mount Point	Enter the name of the file system mount point on the target server. (This dialog will not appear if there is no LifeKeeper-protected filesystem associated with the DataKeeper Resource.)
Root Tag	Select or enter the Root Tag . This is a unique name for the filesystem resource instance on the target server. (This dialog will not appear if there is no LifeKeeper-protected filesystem associated with the DataKeeper Resource.)
DataKeeper Resource Tag	Select or enter the DataKeeper Resource Tag name.
Bitmap File	Select the name of the bitmap file used for intent logging. If you choose None , then an intent log will not be used and every resynchronization will be a full resync instead of a partial resync.

2. Click **Next** to continue. An information box will appear verifying that the extension is being performed.
3. Click **Finish** to confirm the successful extension of your DataKeeper resource instance.
4. Click **Done** to exit the **Extend Resources Hierarchy** menu selection.

Note: Be sure to test the functionality of the new instance on *all* servers by performing a manual switchover. See [Testing Your Resource Hierarchy](#) for details. At this point, DataKeeper has initiated the data resynchronization from the source to the target disk or partition. In the LifeKeeper GUI, the state of the DataKeeper resource on the target server is set to “**Resyncing**”. Once the resynchronization is complete, the state will change to “**Target**” which is the normal **Standby** condition.

During resynchronization, the DataKeeper resource and any resource that depends on it will not be able to fail over. This is to avoid data corruption.

Extending a Hierarchy to a Disaster Recovery System

This operation can only occur from an ISP node or as the continuation of the creation process for multiple nodes from the **Edit** menu or initiated automatically upon completing the **Create Resource Hierarchy** option, in which case you should refer to Step 2 below.

1. On the **Edit** menu, select **Resource** then **Extend Resource Hierarchy**. The **Pre-Extend Wizard** appears. If you are unfamiliar with the Extend operation, click **Next**. If you are familiar with the LifeKeeper Extend Resource Hierarchy defaults and want to bypass the prompts for

2. The **Pre-Extend Wizard** will prompt you to enter the following information.

Note: The first two fields appear only if you initiated the **Extend** from the **Edit** menu.

Field	Tips
Target Server	Enter or select the server you are extending to.
Switchback Type	You must select intelligent switchback. This means that after a failover to the backup server, an administrator must manually switch the Multi-Site Cluster hierarchy resource back to the primary server. CAUTION: This release of SteelEye DataKeeper for Linux does not support Automatic Switchback for DataKeeper resources. Additionally, the Automatic Switchback restriction is applicable for any other LifeKeeper resource that becomes part of the Multi-Site Cluster hierarchy. This includes anything sitting above the hierarchy or becomes a child within the hierarchy.
Target Priority	Select or enter the Target Priority . This is the priority for the new extended DataKeeper 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.
Template Priority	Select or enter a Template Priority . This is the priority for the DataKeeper 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.

3. After receiving the message that the pre-extend checks were successful, click **Next**.

Note: Depending upon the hierarchy being extended, LifeKeeper will display a series of information boxes showing the Resource Tags to be extended, some of which cannot be edited.

4. Click **Next** to launch the **Extend Resource Hierarchy** configuration task.

The next section lists the steps required to complete the extension of a DataKeeper resource to another server.

1. After you have been notified that your pre-extend script has executed successfully, you will be prompted for the following information:

Field	Tips
Mount Point	Enter the name of the file system mount point on the target server. (This dialog will not appear if there is no LifeKeeper-protected filesystem associated with the DataKeeper Resource.)
Root Tag	Select or enter the Root Tag . This is a unique name for the filesystem resource instance on the target server. (This dialog will not appear if there is no LifeKeeper-protected filesystem associated with the DataKeeper Resource.)
Target Disk or Partition	<p>Select the disk or partition where the replicated file system will be located on the target server.</p> <p>The list of disks or partitions in the drop down box contains all the available disks or partitions that are <u>not</u>:</p> <ul style="list-style-type: none"> • already mounted • swap disks or partitions • LifeKeeper-protected disks or partitions <p>The drop down list will also filter out special disks or partitions, for example, root (/), boot (/boot), /proc, floppy and cdrom.</p> <p>Note: The size of the target disk or partition must be greater than or equal to that of the source disk or partition.</p>
DataKeeper Resource Tag	Select or enter the DataKeeper Resource Tag name.
Bitmap File	Select or edit the name of the bitmap file used for intent logging. If you choose None , then an intent log will not be used, and every resynchronization will be a full resync instead of a partial resync.
Replication Path	<p>Select the pair of local and remote IP addresses to use for replication between the target server and the other indicated server in the cluster. The valid paths and their associated IP addresses are derived from the set of LifeKeeper communication paths that have been defined for this same pair of servers. Due to the nature of DataKeeper, it is strongly recommended that you use a private (dedicated) network.</p> <p>If the DataKeeper Resource has previously been extended to one or more target servers, the extension to an additional server will loop through each of the pairings of the new target server with existing servers, prompting for a Replication Path for each pair.</p>
Replication Type	<p>Choose “synchronous” or “asynchronous” to indicate the type of replication that should be used between the indicated pair of servers.</p> <p>As for the previous Replication Path field, if the DataKeeper Resource has previously been extended to one or more target servers, the extension to an additional server will loop through each of the pairings of the new target server with existing servers, prompting for a Replication Type for each pair.</p>

2. Click **Next** to continue. An information box will appear verifying that the extension is being performed.
3. Click **Finish** to confirm the successful extension of your DataKeeper resource instance.
4. Click **Done** to exit the **Extend Resources Hierarchy** menu selection.

Configuring the Restore and Recovery Setting for Your IP Resource

To complete this configuration, you will need configure the **Restore** and **Recovery** setting for your IP resource to **Disable**. This option is displayed in the **Properties** pane. When the **Properties** pane is open for an IP resource or the properties for an IP resource are being displayed, this setting is one of three button options. Refer to the IP Recovery Kit for more information regarding this option.

Note: Be sure to test the functionality of the new instance on *all* servers by performing a manual switchover. See [Testing Your Resource Hierarchy](#) for details. At this point, SteelEye DataKeeper has initiated the data resynchronization from the source to the target disk or partition once the extend to the disaster recovery node is completed. In the LifeKeeper GUI, the state of the DataKeeper resource on the target server is set to **“Resyncing”**. Once the resynchronization is complete, the state will change to **“Target”** which is the normal Standby condition.

During resynchronization, the DataKeeper resource and any resource that depends on it will not be able to fail over. This is to avoid data corruption.

If you haven't done so already, make sure you set the confirm failover flags. Refer to the section [Confirm Failover and Block Resource Failover Settings](#) for more information about this procedure.

Multi-Site Cluster Troubleshooting

The following table lists possible problems and explanations.

Error Message Example	Explanation
ERROR 104108: The disk containing /dev/sde2, protected by resource disk6485, is already in use by another protected LifeKeeper hierarchy. Died at /opt/LifeKeeper/lkadm/subsys/scsi/disrec/bin/create line 107	These messages are generated when trying to protect a Linux Multi-Site Cluster resource that is not on a unique LUN.
Mon Oct 27 16:01:28 EDT 2008 create: ERROR 104103: Cannot create a new disaster-recovery resource instance. Died at /opt/LifeKeeper/lkadmin/subsys/scsi/netraid/bin/create line 184.	

Migrating to a Multi-Site Cluster Environment

The SteelEye Multi-Site Migrate feature is included in the SteelEye Protection Suite for Linux Multi-

Site Cluster product. This additional feature enables an administrator to migrate an existing SteelEye Linux LifeKeeper environment to a Multi-Site Cluster Environment. The migration procedure allows selected shared file system's resources to be safely migrated and replicated with minimum hierarchy downtime.

Following are a few important considerations when creating a Multi-Site resource from an existing file system:

- The Multi-Site migrate procedure will un-mount the file system during the creation process and remount it on a NETRAID device.
- Any applications that depend on this file system will need to be stopped during the create resource procedure. This action is handled by the **Migrate** procedure; no administration action is required.
- Hierarchies containing the following resource types **cannot** be migrated using the Multi-Site migration feature— **NAS** (scsi/netstorage), **DRBD** (scsi/drbd), **SDR** (scsi/netraid) and **Multi-Site Cluster resource** (scsi/disrec).

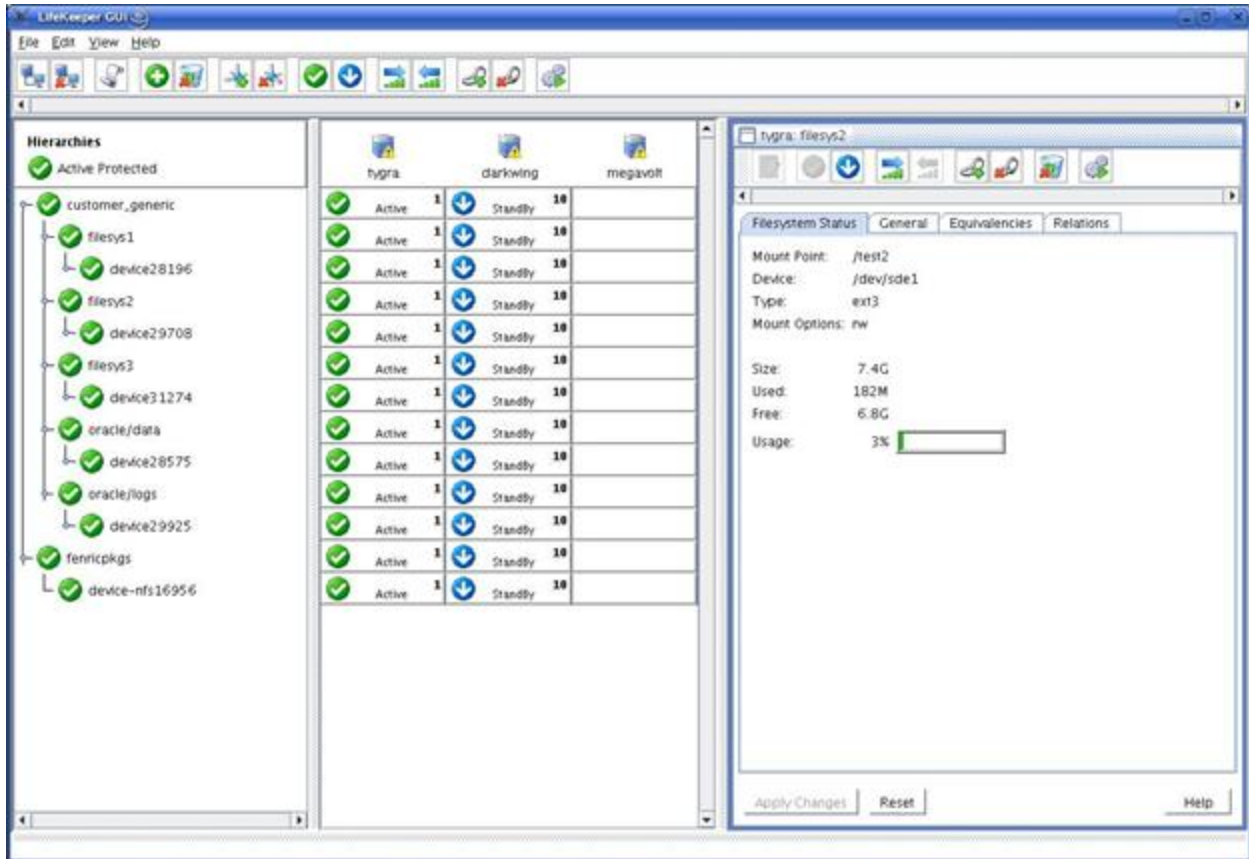
Requirements

Prior to performing a migration, make sure your systems meet the requirements described in the [Installing/Configuration](#) section of this document. In addition to the more general SDR requirements outlined in the Installing SDR section, you must have Novell's SLES 11, SLES 10 or Red Hat Enterprise Linux 5 installed on each system in your cluster. This feature is defined for configurations that have two servers that share a storage device. One of the servers is considered the primary and is located at a primary site. A third server is remote and located at a disaster recovery site.

After you have installed the SteelEye Protection Suite for Linux Multi-Site Cluster on the primary node and other shared storage nodes, there is no additional installation or configuration required to take advantage of the **Migrate** feature.


Before You Start

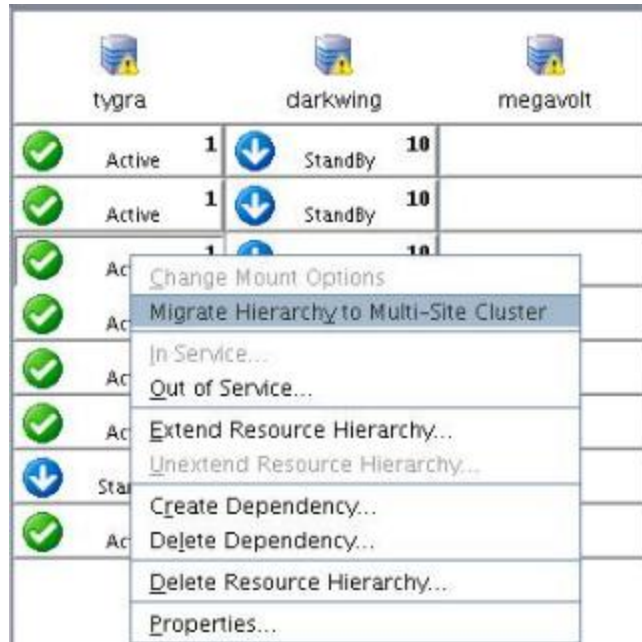
The following image depicts a file system resource hierarchy prior to performing a migrate.



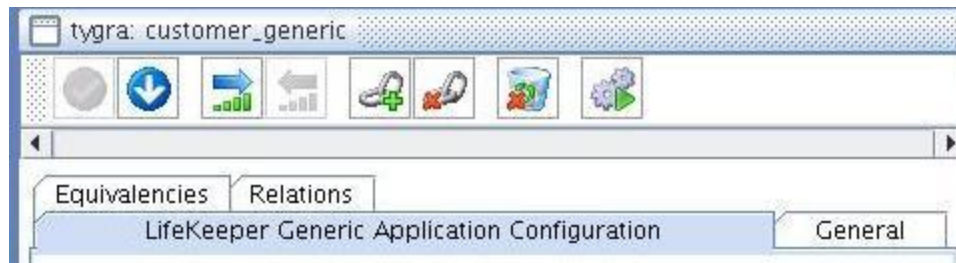
Performing the Migration

There are three methods for configuring and performing a **Multi-Site Migrate**. You can:

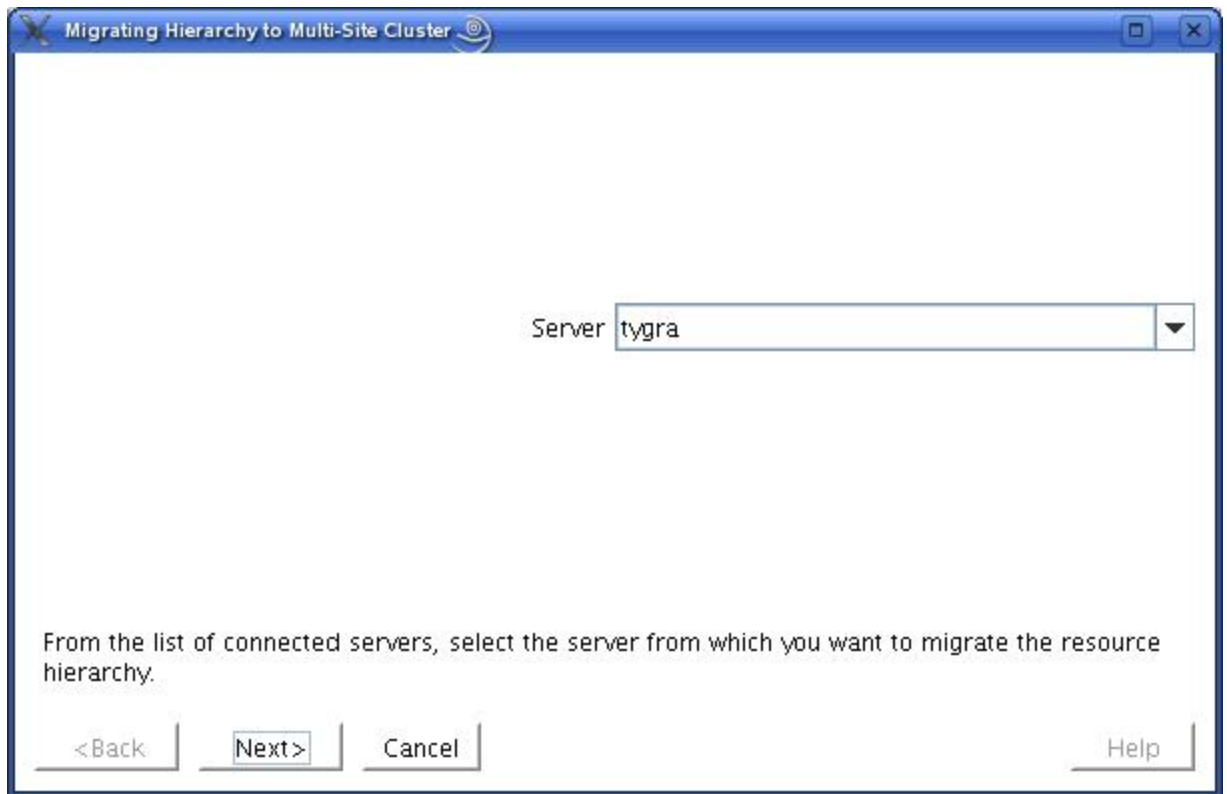
- Select the **Migrate** icon from the LifeKeeper GUI toolbar  and then select the resource to migrate.
- Select the file system resource and right-click the mouse to display the **Migrate Hierarchy to Multi-Site Cluster** menu option.



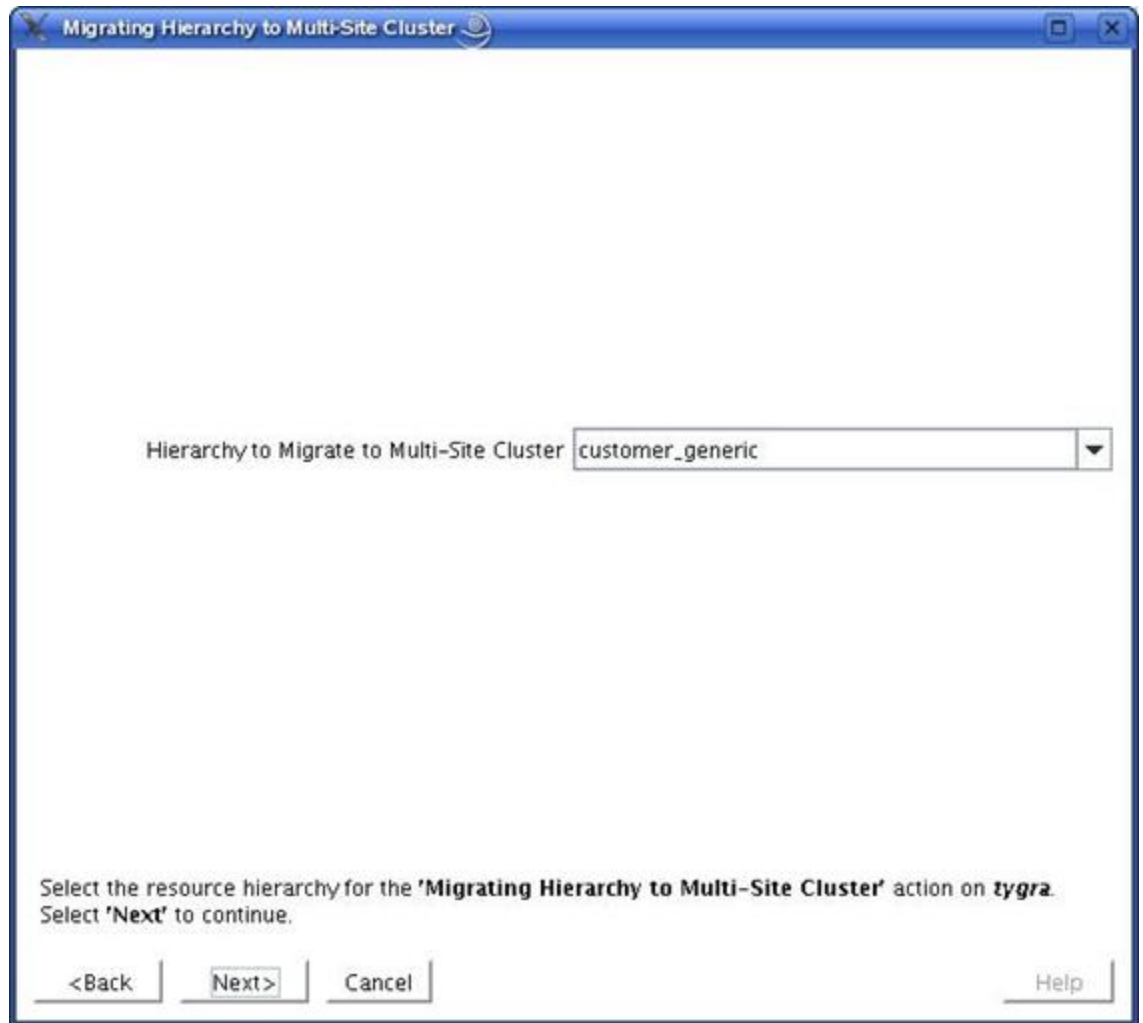
- Select the file system resource and select the **Migration** icon from the **Properties Panel** toolbar.



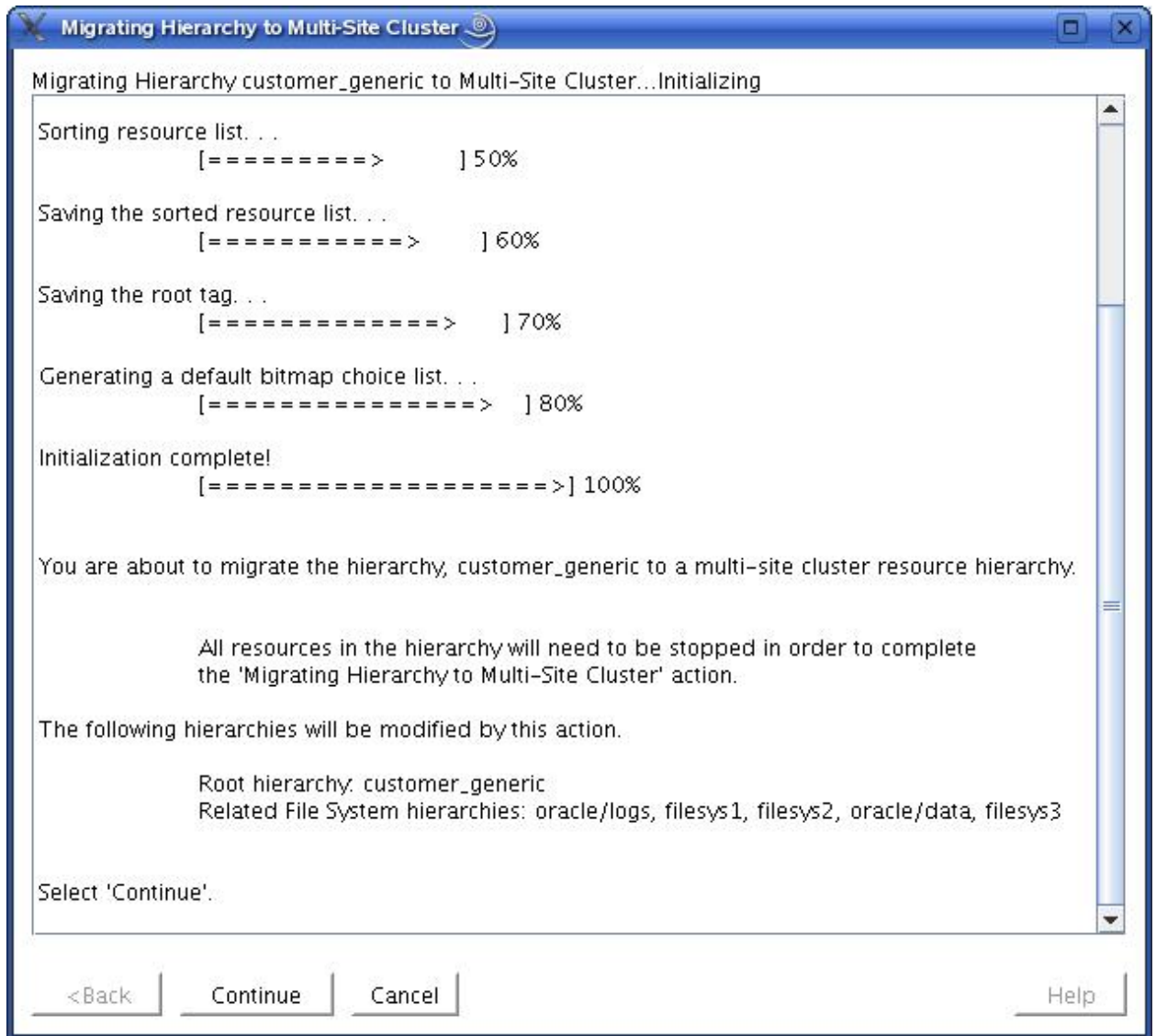
If you initiate the **Migrate** from the **global toolbar** icon, the following dialog box will display:



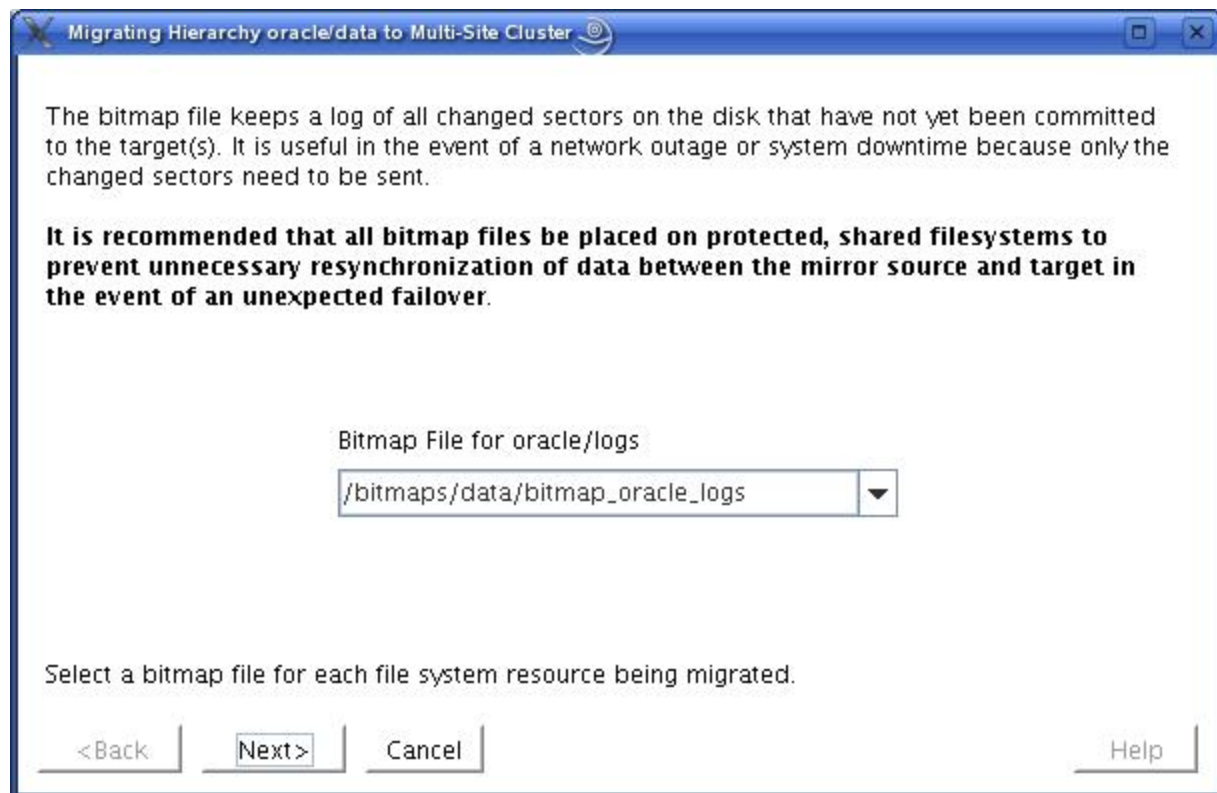
1. Select the server where the **hierarchy to migrate** exists and is in-service. Click **Next**.



2. Select the **root hierarchy tag** that will be migrated and click **Next**. The root tag can be a file system or other application resource. The tag selected (for non-file system resources) must contain a file system dependent resource. If you select a File System in the LifeKeeper GUI window and select **Migrate Hierarchy to Multi-Site Cluster** from the pop-up window or the **Migrate** icon in the **Properties Panel Migrate** icon, the following initialization screen displays.

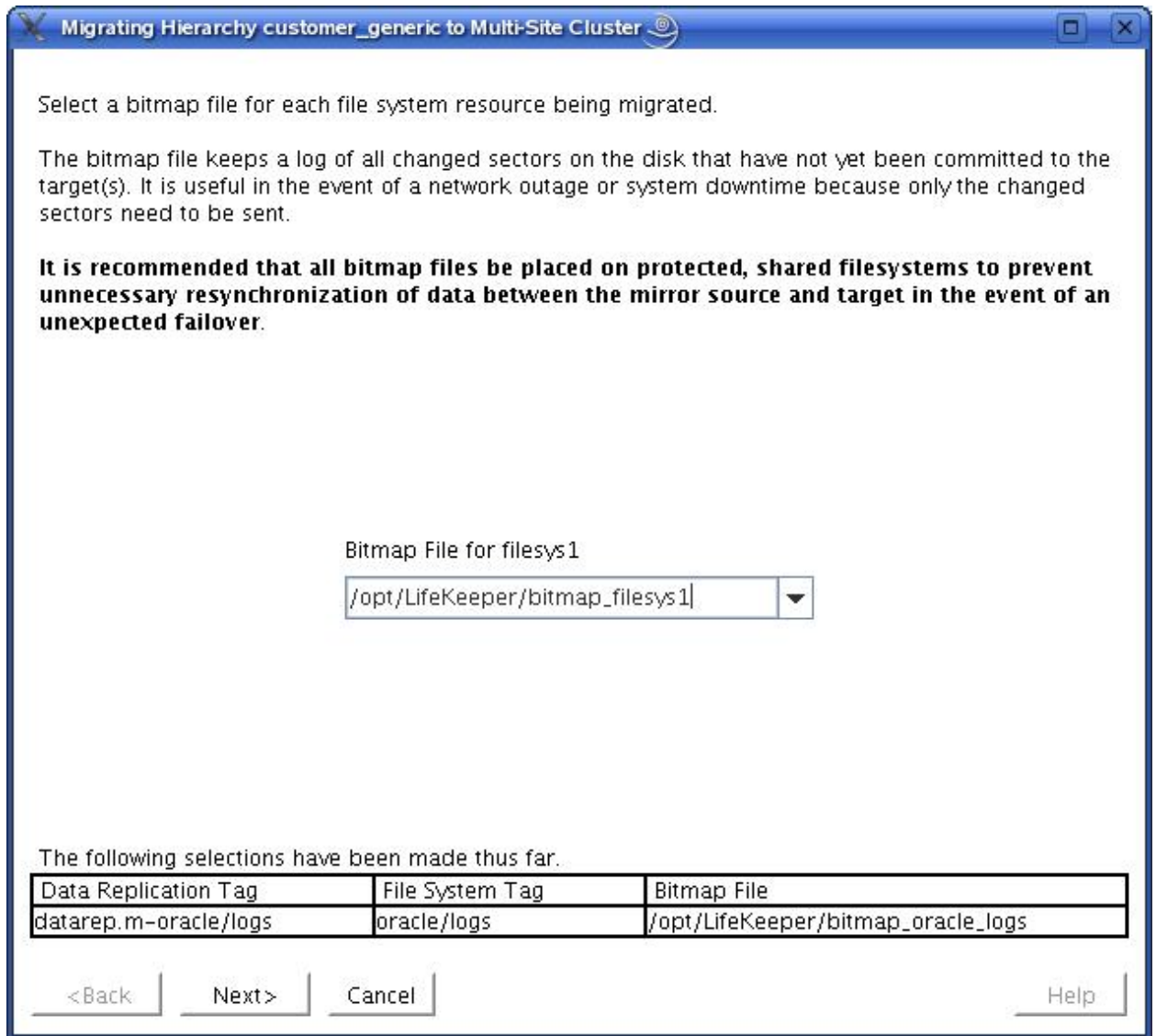


3. Press **Continue** when the Continue button is enabled. The following bitmap dialog will display.

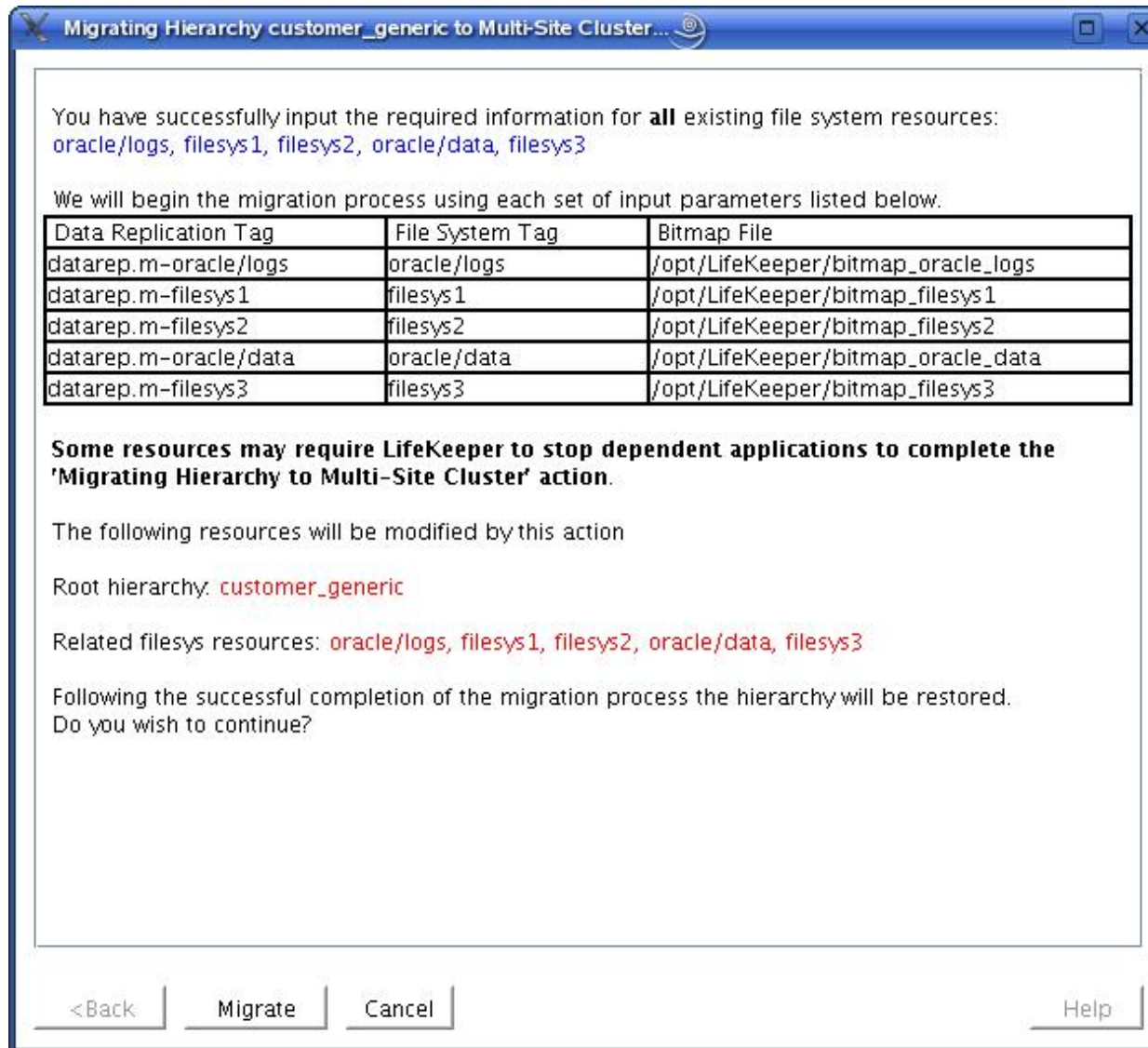


4. Select a bitmap file for the file system you are migrating. Select **Next**.

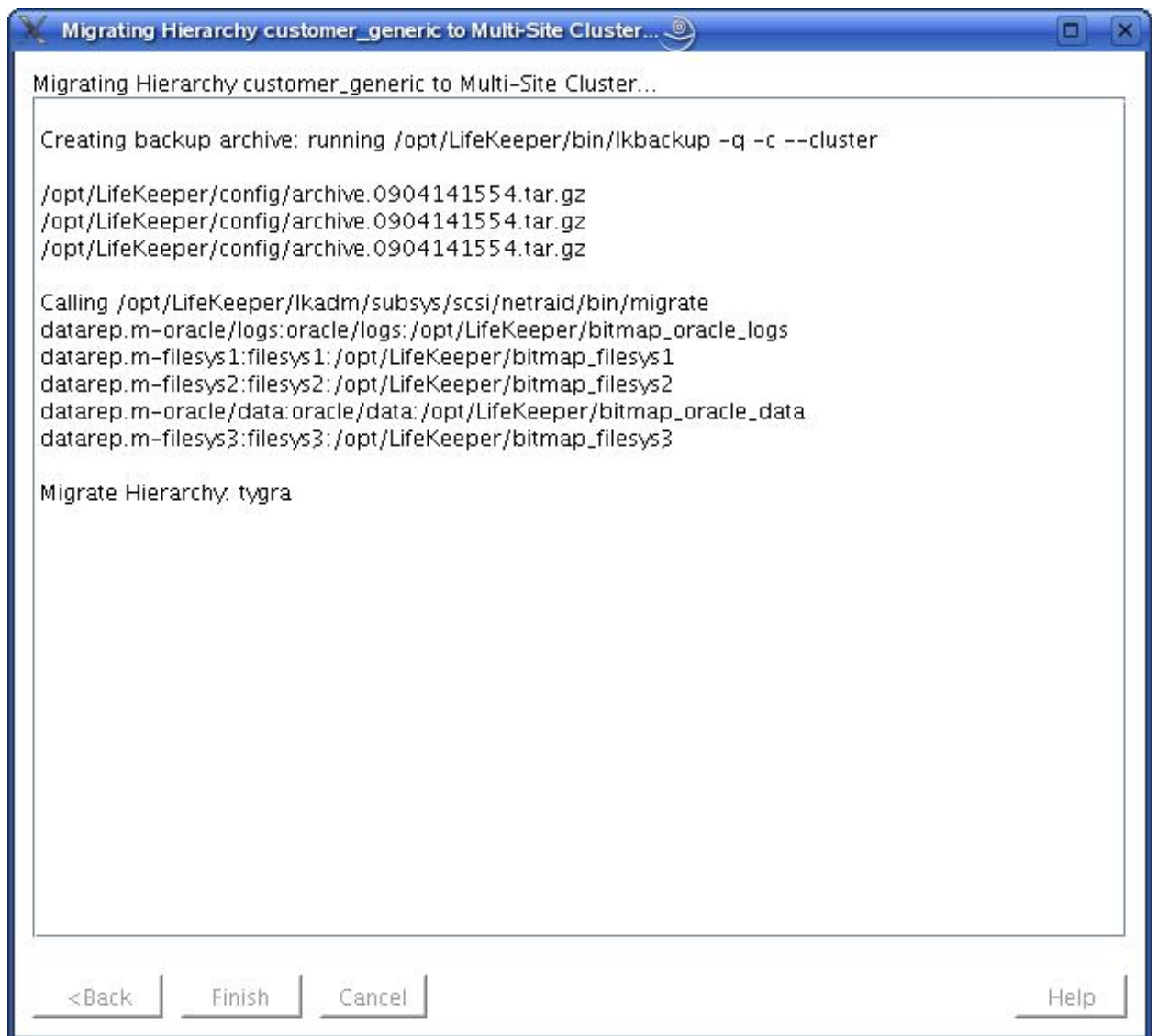
Important: Once you select **Next**, you will not be able to change the **Bitmap File Selection** for this file system resource.



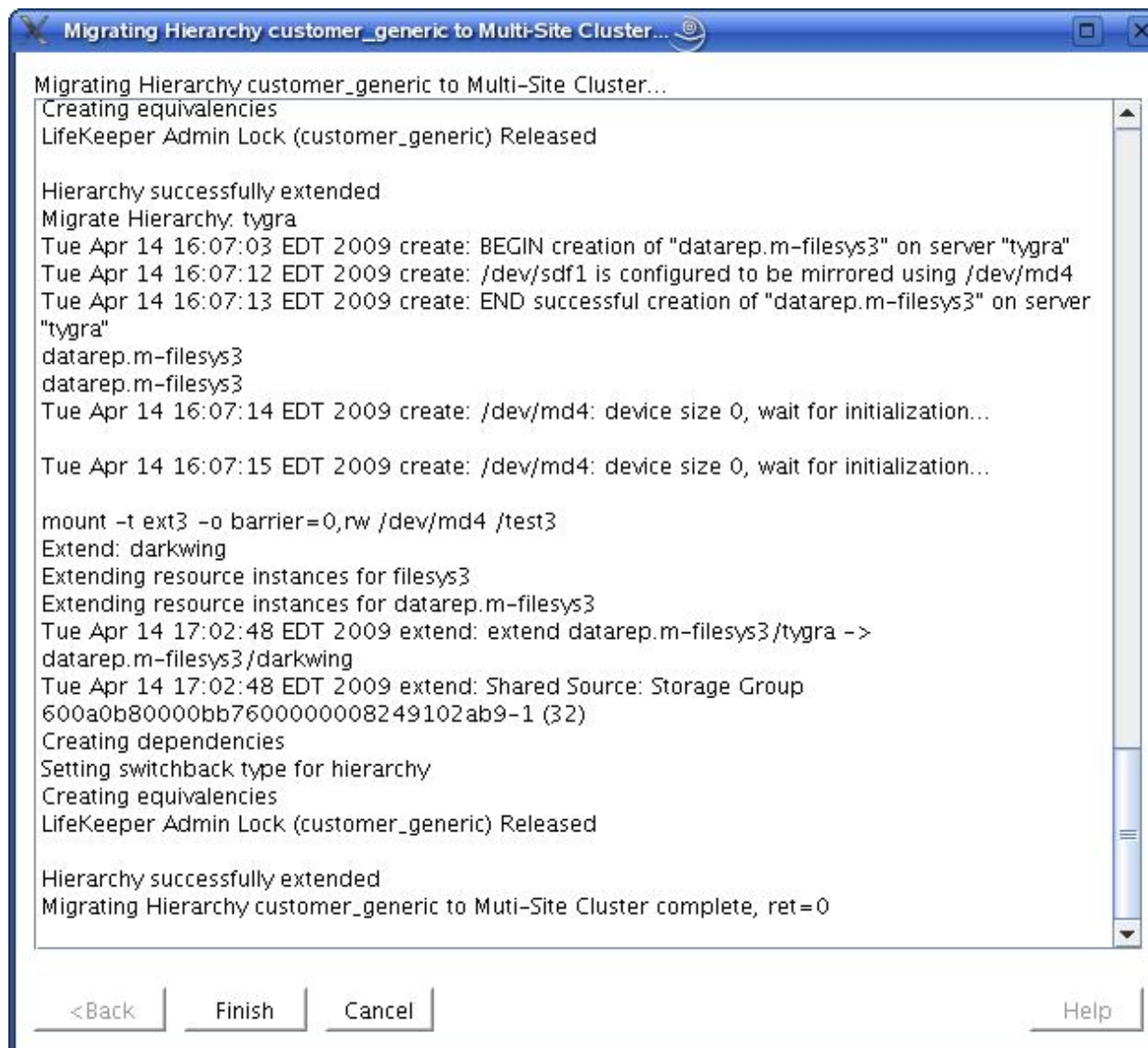
5. Select the second bitmap file for the second file system being migrated within the hierarchy. After selecting the first bitmap file in the previous dialog box, any additional file system tags will be displayed so that the user can enter a unique bitmap file for each additional file system tag.
Note: This screen will not appear if there is only one file system being migrated. Also, multiple screens similar to this will exist if there are more than two file systems being migrated.
6. Select **Next**, a summary screen similar to the one below will display.



7. This **Summary** screen displays all the configuration information you've submitted during the Migrate procedure. Once you select **Migrate**, the following screen displays.



8. The **Migration status** will display in this window. Press **Finish** when the Finish button is enabled.



Successful Migration

The following image is an example of a file system resource hierarchy after the Multi-Site migration is completed. At this time, the hierarchy can be extended to the non-shared node (megavolt).

Successful Migration

The screenshot displays the LifeKeeper GUI interface. On the left, a 'Hierarchies' tree shows a multi-site cluster structure under 'Active Protected'. The tree includes nodes for 'customer_generic', 'filesys1', 'datarep.m-filesys1', 'disrec-26168', 'device28196', 'disk28113', 'disrec-26168-bitmapFS', 'device26688', 'disk28113', 'filesys2', 'datarep.m-filesys2', 'disrec-29088', 'device29708', 'disk29624', 'disrec-29088-bitmapFS', 'device29673', 'disk29624', 'filesys3', and 'datarep.m-filesys3'. The 'datarep.m-filesys1' resource is highlighted.

The central pane shows a table of resources for 'tygra', 'darkwing', and 'megavolt'. The 'tygra' and 'darkwing' columns show 'Active' status with a green checkmark and a '1' in the '1' column. The 'megavolt' column shows 'Standby' status with a blue arrow and a '10' in the '10' column. The 'Source' row for 'tygra' and 'darkwing' shows 'Source' and 'Shared' status respectively.

The right pane shows the 'tygra-datarep.m-filesys1' configuration window. The 'Mirror Configuration' tab is active, showing a green checkmark for 'tygra = darkwing'. The status is 'Shared Storage (600a0b80000bb760000008349102aca-1)'. The bitmap is '4800 bits (chunks, 4800 dirty (100.0%))'. Buttons for 'Apply Changes', 'Reset', and 'Help' are visible at the bottom.

Resource Tag= datarep.m-filesys1, Resource ID= /dev/md1

Chapter 4: Troubleshooting

The following table lists possible problems and suggestions.

Symptom	Suggested Action
NetRAID device not deleted after DataKeeper resource deletion.	Deleting a DataKeeper resource will not delete the NetRAID device if the NetRAID device is mounted. You can manually unmount the device and delete it by executing: <i>mdadm -S <md_device> (cat /proc/mdstat to determine the <md_device>).</i>
Installation/HADR rpm fails	See the Installation section for complete instructions on manually installing these files.
Errors during failover	Check the status of your device. If resynchronization is in progress you cannot perform a failover.
After primary server panics, DataKeeper resource goes ISP on the secondary server, but when primary server reboots, the DataKeeper resource becomes OSF on both servers.	Check the “switchback type” selected when creating your DataKeeper resource hierarchy. Automatic switchback is not supported for DataKeeper resources in this release. You can change the Switchback type to “Intelligent” from the resource properties window.
Primary server cannot bring the resource ISP when it reboots after both servers became inoperable.	If the primary server becomes operable before the secondary server, you can force the DataKeeper resource online by opening the resource properties dialog, clicking the Replication Status tab, clicking the Actions button, and then selecting Force Mirror Online . Click Continue to confirm, then Finish .
Error creating a DataKeeper hierarchy on currently mounted NFS file system	You are attempting to create a DataKeeper hierarchy on a file system that is currently exported by NFS. You will need to replicate this file system before you export it.
DataKeeper GUI wizard does not list a newly created partition	The Linux OS may not recognize a newly created partition until the next reboot of the system. View the <i>/proc/partitions</i> file for an entry of your newly created partition. If your new partition does not appear in the file, you will need to reboot your system.
Resources appear green (ISP) on both	This is a “split-brain” scenario that can be caused by a temporary communications failure. After communications are resumed, both systems assume they are primary.

Symptom	Suggested Action
<p>primary and backup servers.</p>	<p>DataKeeper will not resync the data because it does not know which system was the last primary system. Manual intervention is required.</p> <p>If not using a bitmap:</p> <p>You must determine which server was the last backup, then take the resource out of service on that server. DataKeeper will then perform a FULL resync.</p> <p>If using a bitmap (2.6.18 and earlier kernel):</p> <p>You should take both resources out of service, starting with the original backup node first. You should then dirty the bitmap on the primary node by executing: \$LKROOT/lkadm/subsys/scsi/netraid/bin/bitmap -d /opt/LifeKeeper/bitmap_filesys</p> <p>(where <i>/opt/LifeKeeper/bitmap_filesys</i> is the bitmap filename). This will force a full resync when the resource is brought into service. Next, bring the resource into service on the primary node and a full resync will begin.</p> <p>If using a bitmap (2.6.19 and later kernel or with RedHat Enterprise Linux 5.4 kernels 2.6.18-164 or later (or a supported derivative of RedHat 5.4 or later):</p> <p>You must determine which server was the last backup, then take the resource out of service on that server. DataKeeper will then perform a partial resync.</p>
<p>Installation - Package check errors (<code>rpm -V steeleye-lk</code>) will occur on the core when installed on SUSE</p>	<p>The following errors will occur:</p> <p>Because of the way SUSE runs shutdown scripts (versus other Linux distributions), the following scripts are moved to another location after installation, so LifeKeeper will be shut down when changing run levels or rebooting. These should be the only errors that occur when verifying the steeleye-lk package.</p> <p>Missing <i>/etc/rc.d/rc0.d/K01lifekeeper</i></p> <p>Missing <i>/etc/rc.d/rc1.d/K01lifekeeper</i></p> <p>Missing <i>/etc/rc.d/rc6.d/K01lifekeeper</i></p>
<p>Core - Language Environment Effects</p>	<p>Some LifeKeeper scripts parse the output of Linux system utilities and rely on certain patterns in order to extract information. When some of these commands run under non-English locales, the expected patterns are altered and LifeKeeper scripts fail to retrieve the needed information. For this reason, the language environment variable <code>LC_MESSAGES</code> has been set to the POSIX "C" locale (<code>LC_MESSAGES=C</code>) in <i>/etc/default/LifeKeeper</i>. It is not necessary to install Linux with the language set to English (any language variant available with your installation media may be chosen); the setting of <code>LC_MESSAGES</code> in <i>/etc/default/LifeKeeper</i> will only influence LifeKeeper. If you change the value of <code>LC_MESSAGES</code> in <i>/etc/default/LifeKeeper</i>, be</p>

Symptom	Suggested Action
	<p>aware that it may adversely affect the way LifeKeeper operates. The side effects depend on whether or not message catalogs are installed for various languages and utilities and if they produce text output that LifeKeeper does not expect.</p>
Core - Shutdown hangs on SLES10 systems	<p>When running shutdown on an AMD64 system with SLES10, the system locks up and the shutdown does not complete. This has been reported to Novell via bug #294787. The lockup appears to be caused by the SLES10 powersave package.</p> <p>Workaround: Remove the SLES10 powersave package to enable shutdown to complete successfully.</p>
GUI - GUI login prompt may not re-appear when reconnecting via a web browser after exiting the GUI	<p>When you exit or disconnect from the GUI applet and then try to reconnect from the same web browser session, the login prompt may not appear.</p> <p>Workaround: Close the web browser, re-open the browser and then connect to the server. When using the Firefox browser, close all Firefox windows and re-open.</p>
GUI - lkGUIapp on RHEL5 reports unsupported theme errors	<p>When you start the GUI application client, you may see the following console message:</p> <pre><i>/usr/share/themes/Clearlooks/gtk-2.0/gtkrc:60: Engine "clearlooks" is unsupported, ignoring</i></pre> <p>This message comes from the RHEL 5 and FC6 Java platform look and feel and will not adversely affect the behavior of the GUI client.</p>
Data Replication - GUI does not show proper state on SLES 10 SP2 system	<p>On SLES 10 SP2, netstat is broken due to a new format in <code>/proc/<PID>/fd</code>. This issue is due to a SLES 10 SP2 kernel bug and has been fixed in kernel update version 2.6.16.60-0.23.</p> <p>Solution: Please upgrade to kernel version 2.6.16.60-0.23 if running on SLES 10 SP2.</p>
Data Replication - Size limitation on 32-bit machines	<p>When trying to replicate a drive larger than 2 TB on a 32-bit machine, the following error may occur:</p> <pre><i>Negotiation: ...Error: Exported device is too big for me. Get 64-bit machine</i></pre> <p>Solution: If using SteelEye DataKeeper on a 32-bit machine, you cannot replicate a driver that is greater than 2 TB in size.</p>

Chapter 4: Glossary

SteelEye DataKeeper for Linux Glossary of Terms

Asynchronous mirroring

A type of mirroring in which the primary system writes to the local source device and queues a copy of that write to be transmitted to the target device on the backup system. The data need not reach the target system before the write is acknowledged as complete to the operating system.

Backup system/server

This is a synonym for alternate, spare, or secondary system/server. In a DataKeeper environment, the target device resides on the backup system unless there is a failure of the primary system.

Intent log

A bitmap indicating which data blocks are out of sync between the target and source devices. The intent log can be used to avoid a full resynchronization following a failure.

md

md is the multiple disk driver in the Linux kernel. It is a software RAID manager that provides various RAID levels. SteelEye DataKeeper for Linux uses **md** raid1 in conjunction with **NBD** to provide network replication as a shared storage replacement.

mdadm

mdadm (multiple devices admin) is a raid management package

Mirror

A disk or partition on a primary server whose contents are synchronized across a network to a corresponding disk or partition on a backup server.

Mirror status

The current status of a mirror. Possible states are: Fully operational, Paused, Resyncing, or Out Of Sync. The mirror status can be viewed from the Resource Properties dialog of the DataKeeper resource.

NetRAID device

A DataKeeper device represented as an **md** raid1 device, which consists of a local disk or partition and a Network Block Device (**NBD**).

Network Block Device (NBD)

A device driver in the Linux kernel that lets Linux use a disk on a remote server as one of its local

block devices.

Primary system/server

This is the name of the server in a LifeKeeper configuration with the highest priority for a given resource hierarchy. In a DataKeeper environment, the source device resides on the primary system. All client reads and writes are made to the primary server unless there is a failure.

Replication

The mirroring of data from a source device to a target device across a network.

Resynchronization (Resync)

The process by which SteelEye DataKeeper synchronizes the mirrored data between the source and target devices. This process takes place when the mirror is created, after a pause operation, or after a break in the connection between the source and target. A resync may be full or partial. A full synchronization is performed when the DataKeeper resource is created, and after a failover (with 2.6.18 and earlier kernels – with 2.6.19 and later kernels or with RedHat Enterprise Linux 5.4 kernels 2.6.18-164 or later (or a supported derivative of RedHat 5.4 or later), a partial resync occurs after failover). Otherwise, as long as an intent log file is designated, a partial resynchronization is performed when needed.

Source

In a DataKeeper (mirroring) environment, the source device resides on the primary server. Clients read from and write to the source device only, and writes are replicated across the network to a target device located on a backup server.

Synchronous mirroring

A type of mirroring in which writes are acknowledged only after the data has been written to both the source and target devices.

Target

In a DataKeeper (mirroring) environment, the target device resides on the backup server. Writes are replicated across the network from the source device to the target.

