LifeKeeper® for Linux
Module 6: SteelEye Data Replication and Multi-Site Cluster
Learning Objectives

At the end of this module, you will understand:

- SteelEye Data Replication features
- Product architecture
- Failover scenarios
- SDR Troubleshooting
- LinuxMulti-Site Cluster Configuration
SteelEye Data Replication:

- Provides data replication across existing LAN/WAN infrastructure
- Supports synchronous or asynchronous mirroring
- Intent log alleviates need for full resynchronization
- Provides the netraid resource type
- Fully integrated with LifeKeeper Administrative Interface
- Eliminates the need for an expensive external storage device
- Automated failover protection of replicated data
Choice of protection based on availability needs:
- Different between organizations
- Different between applications
- Requirement will evolve over time
Synchronous vs Asynchronous

**Synchronous:**
- High level of data protection
- Writes to SOURCE and TARGET at the same time
- Network latency impacts application performance
- Available for 2.4 (SDR v4) and 2.6 (SDR v6) kernels

**Asynchronous:**
- Better performance
- Less reliable; data can be out of sync
- Ideal for WAN environments
- Requires 2.6.16+ kernel

**Intent Log:**
- Partial vs Full Re-synchronization
- Available with Synchronous or Asynchronous Mirrors

**Bitmap Merging:**
- Practically Eliminates Full Resyncs
- Requires 2.6.19+ kernel (or back-ported kernel fix – RHEL 5 Update 4)
## Feature/Distribution Matrix

| SDR v6 Feature                          | Red Hat | | | SUSE | | |
|----------------------------------------|---------|---------|---------|---------|---------|
|                                        | RHEL 4  | RHEL 5  | SLES 9  | SLES 10 | SLES 11 |
| Multiple target support (2.6.7+ kernel)| X       | X       |         | X       | X       |
| Bitmap intent logging (2.6.16+ kernel)|         | X       |         | X       | X       |
| Asynchronous (WAN) replication (2.6.16+ kernel) | X       |         | X       | X       |         |
| Bitmap Merging (2.6.19+ kernel)        | X*      |         |         |         | X       |

* RHEL 5 Update 4 (2.6.18-164)
Terminology and Definitions

- **Primary Server**: Per resource, based on server priority (1).
- **Backup Server(s)**: Per resource, based on server priority (>1)
- **Source Server**: Server with data replication resource in-service.
- **Target Server(s)**: Server(s) providing failover protection.
- **Template Server**: Server from which data replication resource attributes are copied during extend of resource.
- **Bitmap**: A file that SDR 6/7 uses as backing store for the replication intent log. Each bit in the file represents one block on the device being replicated.
How Data Replication Works

- RAID1 implemented by Linux MD (Multiple Device) kernel extension
- NBD (Network Block Device) used to transmit/receive data
Failover Scenarios

Scenario 1:
- Server1 (Source) fails
- Server2 becomes Source
- Additional writes occur (on System2 only)
- Server1 returns to service
- Depending on kernel version (and replication type selected), Server2 initiates either full or partial resynchronization to Server1
- “lkcheck” confirms resync complete
- Resource fully protected, eligible to be switched back to Server1
Failover Scenarios

Scenario 2:
- Server1 (Source) fails
- Server2 becomes Source
- Additional writes occur (on System2 only)
- Server1 returns to service
- Server2 initiates a full or partial resynchronization to Server1
- Server2 fails during resync
- Resource failed until Server2 comes up and completes resync
Failover Scenarios

Scenario 3:
- Server1 & Server2 fail concurrently
- Server1 returns to service
- Resource remains out of service until an administrator performs “Force Mirror Online” via the GUI (removes $TAG_data_corrupt flag file)
- Resource last ISP on Server1
  - Confirmed by presence of last_owner flag file

Scenario 4:
- Server1 & Server2 fail concurrently
- Server2 returns to service
- Resource remains out of service (last ISP on Server1)
Hardware Requirements

Servers
- Two or more supported computers configured in accordance with LifeKeeper requirements

Network Interface Cards
- One required
- Best practice: two NICs - replication traffic does not compete with client traffic

Disk Partitions
- Target must be at least as large as source
- Best practice: source and target contain same number of blocks
Software Requirements

- 2.6 kernel (see previous matrix for features)
- LifeKeeper Installation Support
  - HADR-generic-2.6-<version>.noarch.rpm
- LifeKeeper Core RPMs
- SteelEye Data Replication RPM
  - steeleye-lkDR-<version>.noarch.rpm
  - SDR RPM must be version 6.4.0 or above
Configurations

Single Target

Replicate Any Data. Protect Any Application
N+1

Replicate Any Data. Protect Any Application
Multiple Targets

Server1 (primary)

Server2 (backup)

Server3 (backup)

Device Partition

Mirror source

Data replication

Mirror target

Mirror target

Replicate Any Data. Protect Any Application
Configuration Considerations

- Cascading failover supported with 2.6.7+ kernels
  - Up to 7 targets
- Replication over NIC associated with IP resource having local recovery enabled not recommended
- Non-shared storage only with standard SDR (see Multi-Site)
- No automatic switchback
- Replicate a New File System reformats the local partition selected
  - Existing data on target partition is lost
- Target partition is not locked
  - Mounting target partition is not supported and strongly discouraged
  - Device contents are inconsistent while replication in progress
  - Replication occurs below the file system layer
- Target partition size must be \( \geq \) Source partition
Configuration Considerations

- **Bitmap location**
  - If possible, locate the bitmap file on a disk that is not being replicated
  - This will prevent lots of head seeks and can improve performance
  - Due to kernel Bug, no bitmaps on Reiser FS.

- **Replicated Disk Configuration**
  - Replicating (to or from) RAID 5 disks can occasionally become a bottleneck (that can be larger than network bandwidth)
  - RAID 1 configurations usually yield better performance
Packet Compression

- Used when rate of change exceeds network capacity (WAN).
- Implemented via intermediary binary.
- 9 levels of compression (none to maximum)
  - Understand the trade-offs between CPU overhead and network consumption when using higher compression settings.
- Configuring compression
  - Not enabled by default
  - Property is per node per mirror
  - Property found under settings for source and target datarep resources.
    - Select resource, then right click to view menu.
Continuous Data Protection/Rewind

- Default = disabled
- Target-specific settings
- When enabled, changes may be rolled back to address application data corruption or inadvertent deletion of critical data or files.
- Implemented on target
- Circular rewind log must be sized appropriately for the level of protection desired.
- Consider I/O overhead when determining placement of rewind log.
  - The log should be on separate physical disk from target disk.
**Data recovery process:**

- Select mirror target resource
- Right click on resource and select “Rewind and Recover Data”
- Mirror is paused.
- List of rewind points is displayed
- Select and commit the rewind point.
- Data is rewound.
- File system is mounted on target.
- Manually determine validity of data on target.
- If data is valid, commit changes and either switchover to target, or manual copy data from target back to source.
- If data is not valid, choose a different rewind point to validate.
- Iterate through rewind points until a valid point is found.
Set failover confirmation (manual failover):

- Use LifeKeeper GUI to set target server property
  - General tab
  - Select “Confirm Failover From” checkbox
- `lk_confirmso` command line interface
/etc/default/LifeKeeper

- LKDR_ASYNC_LIMIT
  - Sets number of outstanding asynchronous write operations that SDR will allow (default = 256)
  - May increase throughput
  - More data out of sync

If WAN interruptions of >15 seconds are common, use caution adjusting heartbeat parameters
Avoiding initial resync over slow WAN:

- Create initial mirror over LAN
- Take SDR resource out of service
- Execute command on primary server:
  
  ```bash
  $LKROOT/bin/bitmap -c/opt/LifeKeeper/bitmap_<SDR_resource_tag>
  ```
- Transport and install target device at remote site
- Bring source resource in-service
- Extend resource to backup server

Procedure not necessary if replication resource created using “New Replicated File System” option

- Optimized to eliminate initial full resync
WAN Bandwidth Sizing Considerations

- A critical factor in a successful SDR deployment is correctly sizing the network bandwidth to accommodate the rate of data change.

- To measure the rate of change:
  - Determine partition(s) being used (e.g. /dev/sda3). Then, measure the amount of data written in a day:

    ```bash
    MB_START=`awk '/sda3 / { print $7 / 2 / 1024 }' /proc/diskstats`
    ...
    ...
    MB_END=`awk '/sda3 / { print $7 / 2 / 1024 }' /proc/diskstats`
    
    - The daily rate of change, in MB, is then MB_END – MB_START
```
The following approximate rates of change can be handled by SDR with various WAN connections:

<table>
<thead>
<tr>
<th>Connection</th>
<th>Bandwidth</th>
<th>MB per Day</th>
<th>GB per Day</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>1.5 Mbps</td>
<td>15000</td>
<td>15</td>
</tr>
<tr>
<td>T3</td>
<td>45 Mbps</td>
<td>450000</td>
<td>450</td>
</tr>
<tr>
<td>Gigabit LAN</td>
<td>1 Gbps</td>
<td>5000000</td>
<td>5000</td>
</tr>
</tbody>
</table>
For Systems using 2.6.19+ Kernels (SLES 11 & RHEL 5.4) and SDR 6:

- A full resync is required the first time a mirror is created.
- No other scenarios should cause a full resync. All other resyncs should be partial resyncs only.

For systems using 2.6.16 thru 2.6.18 Kernels (RHEL 5.3; SLES 10) and SDR 6:

- A full resync is required the first time a mirror is created.
- A full resync is required anytime there is a hard failure of the source system and a failover occurs to the target. When the original source system comes back up, a full resync will occur back to the original source system (it is now the target).
- No other scenarios should cause a full resync. All other resyncs should be partial resyncs only.
For systems using 2.6 Kernels (< 2.6.16) and SDR 6 or SDR 5 or (RHEL 3, RHEL 4, or SLES 9) or systems using 2.4 Kernels and SDR 4:

- These systems have no bitmap support so full resyncs occur a lot more often.
- A full resync is required the first time a mirror is created.
- A full resync is required anytime there is a hard failure of the source system and a failover occurs to the target. When the original source system comes back up, a full resync will occur back to the original source system (it is now the target).
- Anytime the data replication resource is manually switched over to the target from the source, a full resync will occur.
- Anytime the mirror is paused a full resync will occur.
- A full resync occurs if the target system fails and comes back up.
- A full resync will occur if the nbd client or server processes fail or there is a communications failure on the path used for replication traffic, even if other communication paths are still active.
Data Replication Resource:

- Creates a data replication resource between the specified disk partitions
- Used when LVM is to be configured over SDR or with databases that don't use a file system (raw disk access).

New Replicated File System:

- Creates data replication resource on the specified partition
- Creates file system on the disk partition
- Mounts file system on specified mount point
- Creates File System resource

Replicate Existing File System:

- Replicates a mounted file system
Create a Data Replication Resource:

- Create NetRAID device (/dev/mdN)
- Put resource under LifeKeeper protection

After Resource is created:

- Use with LVM or as a Raw device for a database.
- Extend the hierarchy
Replicated File Systems

Create a New Replicated File System:
- Select existing partition
- Format/mount file system
- Put under LifeKeeper protection

Replicate an Existing File System:
- Select currently mounted partition
- Build NetRAID device
- Begin synchronizing existing data
Deleting an SDR Resource

Take the Data Replication resource out-of-service BEFORE deleting it:

- Out-of-Service: unmounts file system
- Delete: removes the MD devices and adds entry back to /etc/fstab
- Deleting the resource before taking it out of service will leave the file system mounted on /dev/mdN with no entry in /etc/fstab
  - To resolve:
    - unmount file system
    - execute:
      ```
      mdadm --stop /dev/md0
      ```
    - add proper file system mount information to /etc/fstab
    - mount file system
**SDR Resource Configuration Tasks:**
- In-Service
- Out-of-Service
- Extend a Resource Hierarchy
- Unextend a Resource Hierarchy
- Create Dependency
- Delete Dependency
- Delete a Resource Hierarchy
- Properties - View information about resource.

**SDR Mirror Tasks:**
- Pause Mirror
- Resume Mirror
- Force Mirror online
  - both servers have become inoperable
  - Removes data corrupt flag
GUI Sync State Indicators

- Target state during resync
- Source state during resync
- Target state when source and target are in-sync
NetRAID Device Monitoring

Replication Status/Properties (LifeKeeper GUI):

- Mirror and resync status
- Mirror type (sync/async)
- Intent logging
- Source & Target
- Force Mirror Online

![Resource Properties for datarep-ext3-sdr](image)

- **eve → adam**
  - Status: Paused
  - Type: Asynchronous

- **eve → sophocles**
  - Status: Fully Operational
  - Type: Synchronous
  - Bitmap: 4895 bits (chunks), 4895 dirty (100.0%)
$LKROOT/bin/mirror_status <resource_tag>

- Monitor the progress of resynchronization
- Obtain general status

`cat /proc/mdstat`

- View status of MD RAID devices

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

- `<tag>` is the LifeKeeper resource tag of the SDR resource
- `<action>` is one of: pause, resume, force
- `<source>` is the current source system
- `<target>` is the target system (or list of systems) that the action should affect
SDR/LVM2 Compatibility:

- Replication of physical volumes is not recommended.
  - Requires both the LVM ARK and SDR.
  - Data integrity can only be guaranteed if there is one physical volume in a volume group and one volume group in a logical volume.
  - Similar issues with DRBD.

- Replication of logical volumes is fully supported.
  - No LVM ARK. LVM configurations must be managed manually.
  - Use SDR rewind, not LVM snapshots.

- Modification of /etc/lvm.conf recommended.
  - Add “r |/dev/nbd*|” to the filter= parameter.
  - Prevents LVM from detecting multiple physical volumes with same signature.
SDR/LVM2 Compatibility (cont.):

- Logical volumes defined by /dev/mapper devices are automatically registered in a LifeKeeper configuration file.

  
  /opt/LifeKeeper/subsys/scsi/resources/DEVNAME/device_pattern
<table>
<thead>
<tr>
<th>Symptom</th>
<th>Suggested Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>NetRAID device not deleted when Data Replication resource is deleted.</td>
<td>Deleting an SDR resource will not delete the NetRAID device if NetRAID device is mounted. Manually unmount the device and delete it.</td>
</tr>
<tr>
<td>Installation of HADR rpm fails.</td>
<td>See Planning and Installation Guide for complete instructions on manually installing files.</td>
</tr>
<tr>
<td>Errors during failover.</td>
<td>Check status of device. If resynchronization is in progress failover can not be performed.</td>
</tr>
</tbody>
</table>
| After failure of primary/source server, resource goes ISP on backup server, but when primary server reboots, SDR resource becomes OSF on both servers | Check “switchback type” selected when Data Replication resource was created. Automatic switchback is not supported with Data Replication resources. Change Switchback type to “Intelligent”.

Replicate Any Data. Protect Any Application
## Troubleshooting

<table>
<thead>
<tr>
<th>Symptom</th>
<th>Suggested Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary server cannot bring resource ISP after both servers become inoperable.</td>
<td>If primary server becomes operable before the secondary server, force the Data Replication resource online by right clicking the SDR resource and selecting Force Mirror Online. Click Continue to confirm, then Finish.</td>
</tr>
<tr>
<td>Error creating Data Replication resource on currently mounted NFS file system.</td>
<td>Attempting to create Data Replication resource on file system currently exported by NFS. Replicate this file system before exporting it.</td>
</tr>
<tr>
<td>SDR GUI wizard does not list newly created partition.</td>
<td>Linux OS may not recognize newly created partition until system is rebooted. View /proc/partitions file for entry related to newly created partition. If new partition does not appear in the list, reboot the system.</td>
</tr>
<tr>
<td>Resources appear green (ISP) both primary and backup servers.</td>
<td>“split-brain” has occurred; caused by temporary failure of all communication paths. SDR will not resync data because it does not know which system was last primary. Manual intervention is required. Either use manual confirm of failover or a STONITH device to prevent this condition.</td>
</tr>
</tbody>
</table>
New Optional Feature added in LifeKeeper 6.4.0

- License for Multi-Site is not included with SDR

Allows Creation of a Hybrid Cluster

- Normal Shared Storage Cluster at the primary site
- Replicated server (or servers) as an off-site backup system for disaster recovery
The Best of Both Worlds – Shared and Replicated Storage

- The File System resource acts normal at the primary site
  - Local shared storage failovers and switchovers
  - Built on a single LUN (just like any normal LifeKeeper File System Resource) with Fibre Channel or iSCSI storage
  - Multi-path solutions available for most popular Storage Devices
  - SCSI Reservations for I/O Fencing
- The shared storage file system acts as the source of the SDR mirror
  - SteelEye Data Replication uses LifeKeeper protected shared storage resources as the primary source of the replication
  - The source of the mirror can move with local switchovers or failovers
- Shared storage is not required at the mirrored site
- Extremely cost effective Disaster Recovery protection when compared to storage-based replication
Shared Bitmaps

- A configurable feature of Multi-Site
- Allows the SDR replication bitmap file to exist on a shared storage LUN that then can be switched within the local shared storage cluster.
- The location of the shared bitmap is LifeKeeper protected (as a standard file system resource).
- Setting up a shared bitmap prevents full resyncs to the disaster recovery site whenever there is a switchover or failover at the local shared storage site.
- Since the bitmap exists on switchable storage, the system bringing the Multi-Site file system in service can access the same bitmap that was in use before the switchover or failover.
- Configuration of this feature is highly recommended to prevent full resyncs.
Unbalanced Hierarchies

- A Multi-Site configuration will result in a “unbalanced hierarchy” in LifeKeeper
- It is not bad or wrong; it is just an unusual configuration for longtime LifeKeeper users.
- The shared file system devices and the shared bitmap resources do not need to exist at the replicated node.
Shared Storage Setup

- Just like setting up a normal shared storage cluster, make sure to configure your storage before you start building a Multi-Site cluster.
- Refer to the tips in Module 5 on configuration of shared storage.
- If you plan to use the Shared Bitmap feature (and it is strongly recommended) be sure to set aside a very small LUN for each Multi-Site configured file system you plan to create.
- The Shared Bitmap LUN can be very small. Each bit (in the file) represents one block that is being replicated. (1 TB of replicated storage would need a bitmap of less than 3 MB) It needs a really small LUN!
- Partition and create a file system on your shared bitmap LUN.
- Mount the shared bitmap file system on the primary host in the local cluster.
- Create a resource in LifeKeeper for the shared bitmap file system. Extend this to all the local cluster systems.
Shared Storage Setup (continued)

• When creating a “New Replicated File System” with Multi-Site for the critical data:
  • Make sure you can see the LUN from all the local systems
  • Partition the LUN

• When “Replicating an Existing File System” with Multi-Site
  • Make sure you can see the LUN from the other local systems
  • Mount the partition on the primary system

Replication Site Setup

• Create a partition that is the same size (or slightly bigger) than the one that is to be protected.
• Don’t bother to build a file system or mount this partition.
• Ensure that you will have enough network bandwidth between the shared storage site and the disaster recovery site to handle all of the data change that may occur. (Just like with standard SDR).
Create and Extend your Multi-Site Resources

- Creation is slightly different than creating a shared storage file system resource.
- Let’s walk through an example …
There are three systems in the example cluster:

- Putnam and Rose share a storage device
- Cabell is located at the disaster recovery site
- The shared bitmap resource has already been created and mounted on /bitmap2 (small, 16 MB partition)
- Putnam will be the primary node
Create a Multi-Site Resource

- /dev/sdf1 will be the protected partition
- There is a partition of similar size on third machine at the remote site
- Begin the resource creation by clicking
- Select the Data Replication Recovery Kit
- Select “Replicate New Filesystem”
- The mount point is /s2
Create a Multi-Site Resource

- The wizard asks for a Bitmap File location
- Select the location on the shared storage device created and mounted earlier (/bitmap2)
Create a Multi-Site Resource

- The Create action completes
- The Extend action to the next (shared storage) node is automatically invoked
Create a Multi-Site Resource

- Complete the extend operation to the second shared storage node
Create a Multi-Site Resource

- Extend the resource hierarchy.
- Supply the host name (fully qualified) of the machine at the disaster recovery site (AKA “Target Server”)

![Pre-Extend Wizard](image)
Select the target disk partition from the available options.
Create a Multi-Site Resource

- If the same bitmap path is used (as on the shared storage side) a warning may be given.
- Just as with standard SDR, it is best practice to locate the bitmap on a separate device, eliminating disk head contention that can significantly impact performance.

The bitmap file keeps a log of all changed sectors on the disk that have not yet been committed to the target(s). It is useful in the event of a network outage or system downtime because only the changed sectors need to be sent. By default, the bitmap file will contain one bit per 64KB of data on the disk (this can be changed with the LKDR_CHUNK_SIZE variable).

On the target, the bitmap file is necessary when replication changes roles (i.e., on switchover or failover). Without a bitmap file, it is impossible to switch back again to the source without transmitting all the data from the target.
Select the replication path, or network endpoint

- Best practice: use a separate NIC for replication to eliminate contention with application-related traffic.
Create a Multi-Site Resource

- Select the replication type
  - Best practice: use asynchronous replication when performing replication over a high latency network.

![Image of Replication Resource](image-url)
Create a Multi-Site Resource

- Select another replication path or network endpoint to be used for the second shared storage node
- The Extend action will complete
The File System resource can now be used with other resources
- Note unbalanced equivalencies
Multi-Site Configuration Considerations

Configurations to Avoid

- Multi-Site hierarchy shares underlying device (or devices) with another hierarchy
  - Not a good idea anytime, but will not work with Multi-Site
  - Causes problems whenever resources are switched to the Disaster Recovery site
  - Can easily trigger full resyncs

- Examples
  - Using the Multi-Site Cluster hierarchy’s mirror disk resource more than once in the same or different hierarchies.
  - 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 not be shared.)
  - Using the bitmap file system, device or disk resource with another hierarchy (Multi-Site or non-Multi-Site).
Multi-Site Configuration Considerations

Confirm Failover Flags

- As with standard SteelEye Data Replication configurations, confirming failover to a WAN connected site is often prudent.
- Set the Confirm Failover Flag for the Disaster Recovery site (from the LifeKeeper GUI)

```
Set Confirm Failover:
  Configures the confirmso!cabell.dsims.us 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
  cabell.dsims.us   [ ]  [ ]
  rose.dsims.us    [ ]  [ ]
  putnam.dsims.us  [ ]  [ ]
Confirm Failover Flags (continued)

- If the flag is set, LifeKeeper will delay failover until the failover is confirmed (via command line)
  
  - `lk_confirms -y system` < Confirms failover
  - `lk_confirms -n system` < Blocks failover

- Two flags in the `/etc/default/LifeKeeper` control this behavior
  
  - **CONFIRMSODEF**
    - If set to zero, proceed with failover after waiting
    - If set to one, block failover after waiting
  
  - **CONFIRMSOTO**
    - Specifies the time in seconds to wait before taking the automatic action specified by `CONFIRMSODEF`.
    - Note: The default setting is 10 minutes.
Multi-Site Networking Considerations

Client Application Connectivity to the Disaster Recovery site

- There are several potential solutions
  - One Size Does NOT Fit All!
- VLAN
  - From a LifeKeeper point of view, this is the easiest solution.
  - The IP at the primary site can also work at the disaster recovery site with no modification.
  - Usually best implemented with router hardware (e.g. Cisco Local Area Mobility)
Multi-Site Networking Considerations

- **LifeKeeper Shared Equivalency**
  - Used when there is no way to implement the same subnets at both sites.
  - Create an IP address resource at the primary site and a different IP resource at the disaster site.
  - `eqv_create` – (from module 5) create an equivalency between specified resources. The equivalency must be created on ALL cluster members.

- **Use the “Disable Restore and Recovery” feature in the IP kit**
  - This is even easier than building shared equivalencies
  - Use IP address at primary site. When extending IP resource to DR site, selecting this option makes sure it is not brought in-service (no monitoring).

- **Dynamic DNS Update**
  - Can be used in conjunction with Shared Equivalency or Disable Restore techniques.
  - Requires creation of a generic application kit to perform the update.
Protecting an Existing File System

- When creating a Multi-Site resource from an existing file system
  - SDR/Multi-Site 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.
Multi-Site Upgrade Considerations

Upgrades to an Existing Shared Storage LifeKeeper Hierarchy

- Still a manual process in LifeKeeper 6.4
- Due to the nature of how MD/NBD works in SteelEye Data Replication, file systems have to be un-mounted and re-mounted on a NETRAID device.
- A small amount of downtime for the hierarchy is unavoidable.
- Current best practice (to limit downtime):
  - Have the resource hierarchy in-service on the Primary system.
  - Delete the dependency (in LifeKeeper) from the file system (to be protected) and the rest of the hierarchy (the application portion).
  - Un-extend the file system hierarchy from the other local cluster nodes.
  - Stop (out of service) the application portion of the hierarchy.
  - Delete the file system resource from LifeKeeper (the file system will remain mounted.)
Upgrades to an Existing Shared Storage LifeKeeper Hierarchy

- Current best practice (continued):
  - Create the Multi-Site resource for the existing file system. Do not Extend at this time.
  - Restart the application portion of the hierarchy (in service).
  - Extend the Multi-Site resource to the other local cluster nodes.
  - Extend the Multi-Site resource to the disaster recovery node.
  - Extend the application portion of the hierarchy to the disaster recovery node.
  - Create a dependency in LifeKeeper between the application portion of the hierarchy and the new Multi-Site resource.
  - Test the new Multi-Site enabled cluster carefully.
Multi-Site Upgrade Considerations

Easy Upgrade to Multi-Site Cluster

- Automated Upgrade in LifeKeeper 7.0
- Due to the nature of how MD/NBD works in SteelEye Data Replication, file systems have to be un-mounted and re-mounted on a NETRAID device.
  - Upgrade Process Warns that this will occur
  - A small amount of downtime for the hierarchy is unavoidable.
- Upgrade Process kicked off from the LifeKeeper GUI
- Collects a lot of Information first
  - There can be a lot going on here!
    - Shared Bitmaps, etc.
- Takes some time….
  - Process Errs on the Side of Being SAFE
  - Easy and Safe, not so fast.
Easy Upgrade to Multi-Site Cluster

- Process does “Prep” for Upgrade First
- Prepares the Underlying meta-device
  - On the Primary first
  - On the Shared Backup system(s) next
- The Extend to the Replicated Leg
  - The Upgrade process will walk the Admin through this
  - Or you can stop and do the Extend to the Replicated Leg later
    - Just kick off the Extend manually (via the GUI)