LifeKeeper for Windows

v7.2.1

Technical Documentation

January 2012

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

Table of Contents

Chapter 1: About LifeKeeper for Windows

LifeKeeper for Windows provides a fault resilient software solution to provide high availability for data, applications, and communications resources. LifeKeeper does not require any customized, fault-tolerant hardware. You simply group two or more systems and enter site-specific configuration data. Then, LifeKeeper automatically provides fault detection and recovery.

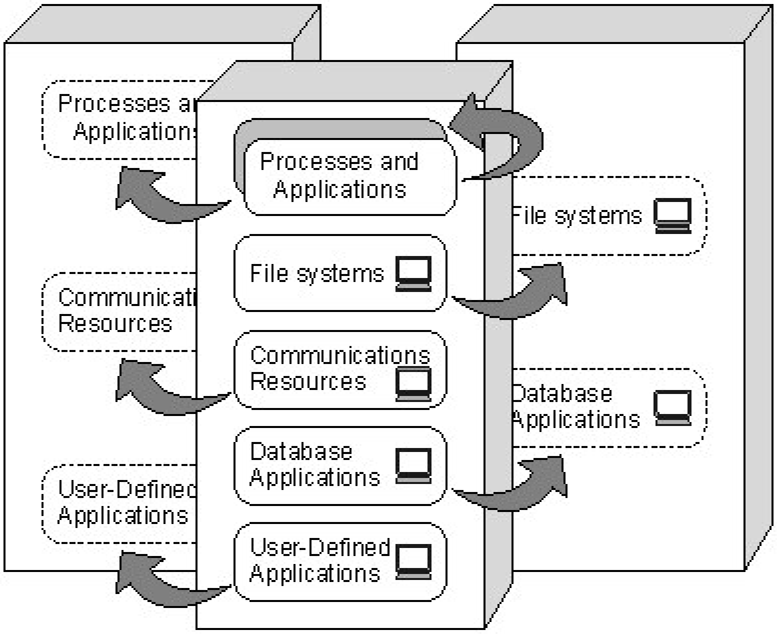
In case of a failure, LifeKeeper migrates protected resources from the failed system to the backup system. Users experience a brief interruption during the actual switchover, but LifeKeeper automatically restores operations on the backup system when it completes the failover recovery.

Protected Resources

The LifeKeeper family of products includes software that allows you to provide failover protection for a range of system resources. The figure below demonstrates LifeKeeper's flexibility and identifies the resource types you can specify for automatic recovery:

- **Volume**. With LifeKeeper’s Volume resource type, you can protect data and applications on shared SCSI peripherals or replicated volumes by creating a resource in the LifeKeeper hierarchy for the disk volume containing those resources.
- **File share**. LifeKeeper's LAN Manager Recovery Kit lets you protect a specific folder or directory on a shared drive.
- **Computer alias name**. LifeKeeper's LAN Manager Recovery Kit enables automatic failover of the computer alias name for applications that communicate to the server via NetBEUI.
- **Communications resources**.
    - LifeKeeper’s IP Recovery Kit allows you to create resources that enable switchover of IP addresses.
    - LifeKeeper's DNS Recovery Kit provides a mechanism to update DNS A and PTR records.
- **Web Server resources**. LifeKeeper's Microsoft IIS Recovery Kit allows you to protect Microsoft Internet Information Services (IIS) resources.
- **Mail Server resources**. LifeKeeper provides optional Microsoft Exchange Server Recovery Kits for mail server resources.
- **Database applications**. LifeKeeper provides optional Recovery Kits for major RDBMS products, such as Microsoft SQL Server, and Oracle.
- **Generic applications**. LifeKeeper's Generic Application Recovery Kit allows the creation of resources for an application that has no predefined recovery kit.

LifeKeeper supports N-Way recovery for a range of resource types. N-way recovery allows different resources to failover to different backup servers in a cluster.



See LifeKeeper Core Software for the core components available with LifeKeeper.

LifeKeeper Core Software

LifeKeeper for Windows Core includes the basic LifeKeeper software packages, plus the Core Recovery Kits.

Core LifeKeeper Software

- **LifeKeeper Configuration Database (LCD)** - The LCD stores information about the LifeKeeper-protected resources. This includes information on resource instances, dependencies, shared equivalencies, recovery direction, and LifeKeeper operational flags. The data is cached in shared memory and stored in files so that the data can be remembered over system boots.

- **LCD Interface (LCDI)** - The LCDI queries the configuration database (LCD) to satisfy requests for data or modifications to data stored in the LCD. The LCDI may also be used by the Application Recovery Kit to obtain resource state or description information.

- **LifeKeeper Communications Manager (LCM)** - The LCM is used to determine the status of servers in the cluster and for LifeKeeper inter-process communication (local and remote). Loss of LCM communication across all communication paths on a server in the cluster indicates the server has failed.

- **LifeKeeper Alarm Interface** - The LifeKeeper Alarm Interface provides the infrastructure for triggering an event. The sendevent program is called by application daemons when a failure is detected in a LifeKeeper-protected resource. The sendevent program communicates with the LCD to determine if recovery scripts are available.

- **LifeKeeper Recovery Action and Control Interface (LRACI)** - The LRACI determines the appropriate recovery script to execute for a resource and invokes the appropriate restore / remove scripts for the resource.
- **LifeKeeper GUI** - The LifeKeeper GUI is a client/server application that provides a graphical administration interface to LifeKeeper and its configuration data. The LifeKeeper GUI client is implemented as both a stand-alone application and as a Java applet invoked from a web browser.

LifeKeeper Core Recovery Kits

The Core Recovery Kits provide all fault detection and recovery mechanisms and the definitions necessary to provide LifeKeeper protection for volumes (drive letters), file shares, communications resources, and Microsoft Internet Information Server.

- **Volume Recovery Kit** - Allows you to create a resource to protect an entire shared or mirrored drive (for instance, the K: drive).
- **LAN Manager Recovery Kit** - Enables automatic recovery of the computer alias name and Windows file share lists. The LAN Manager resource lets you create a"switchable" computer name for applications that communicate with the server via NetBEUI, or system name. It includes the File Share recovery component.
- **IP Recovery Kit** - Provides a mechanism to recover a "switchable" IP address from a failed primary server to one or more backup servers in a LifeKeeper environment. A switchable IP address is a virtual IP address that can switch between servers and is separate from the IP address associated with the network interface card of each server. Applications under LifeKeeper protection are associated with the switchable IP address, so if there is a failure on the primary server, the switchable IP address becomes associated with the backup server. This kit also provides local recovery of the IP resource.
- **DNS Recovery Kit** - Provides a mechanism to update DNS A and PTR records of the primary server or an alias name. After a failover or switchover to the backup server, the A record and the associated PTR record (if exists) of the primary server or alias name will be updated with the IP address of the backup server.
- **Microsoft IIS Recovery Kit** - Protects Web, FTP and SMTP services of the Microsoft Internet Information Services (IIS). It continuously monitors the health of your Internet servers, and if a problem arises, provides automatic failover of the affected web server to a backup server.
- **Generic Application Recovery Kit** - Allows the creation of resources for an application that has no predefined recovery kit.

LifeKeeper Recovery Kits

LifeKeeper Application Recovery Kits (ARKs) include tools and utilities that allow LifeKeeper to manage and control a specific application. When a recovery kit is installed for a specific application, LifeKeeper is able to monitor the health of the application and automatically recover the application if it fails. The LifeKeeper Recovery Kit is non-intrusive and requires no changes within the application in order for LifeKeeper to protect it.

Note: All separately packaged LifeKeeper Recovery Kits now require a software license key in order to function with LifeKeeper v4.3 and higher.

LifeKeeper for Windows offers the following optional Recovery Kits:

- **Microsoft SQL Server** - This recovery kit provides menu-driven definition of resources for automatic switchover of a Microsoft SQL Server instance. The kit provides options that allow you to easily create a complete resource hierarchy so that the recovery operation can include all disk resources used by the SQL Server as well as the Named Pipe or IP socket resources used to access the database.
- **Microsoft Exchange Server** - This Protection Suite for Exchange provides a comprehensive set of tools to address all areas of Exchange business continuity, including Exchange data protection, disaster recovery and high availability.
    - Real-time CDP protects Exchange data and provides the highest RPO
    - Classic failover and recovery for Exchange
    - Support for shared storage and data replication
    - Complete monitoring of Exchange processes
    - Support for Exchange 2003 and 2007
    - Standard or Enterprise versions supported
- **Oracle Recovery** - This recovery kit provides system administrators with default recovery scripts for automatic startup and shutdown of Oracle databases that are protected by LifeKeeper for Windows. By customizing these recovery kits, a system administrator can tailor their recovery actions for specific applications utilizing the Oracle database.

Product Requirements and FAQ information is delivered in the OS specific Release Notes.

Documentation for each recovery kit is included in the LifeKeeper Application Recovery Kits section.

Communication Path Overview

LifeKeeper monitors resource operations and provides failover using shared communication paths (comm paths) between servers. It is critical to LifeKeeper fault detection and resource recovery that communication between the servers remains active. As a result, you need to define multiple comm paths, using different protocols, to guard against a system failover simply because a communication line fails.

Before you can define resources or resource hierarchies in LifeKeeper, you must define your comm paths on each of the servers. LifeKeeper uses the comm paths to coordinate resource definitions and states between the nodes and for fault detection and recovery.

The Communication Path Types section describes these comm path topics:

- **Comm path types**. LifeKeeper supports three types of comm paths for two-server configurations: TCP/IP, TTY, and Shared Disk. Configurations with greater than two servers support only TCP/IP comm paths.
- **LifeKeeper heartbeat**. A key function of the comm path between the servers is the LifeKeeper heartbeat. This periodic signal between the servers assures each server that the other is still alive and processing.
- **Safety check**. If all comm paths die, LifeKeeper performs a safety check to verify system failure before switching over resources.

Communication Path Types

LifeKeeper provides three different types of comm paths so that you can define redundant comm paths using different protocols. Although there is less value in defining multiple paths of the same type over the same media, redundant paths using different protocols or different media provide good protection against failover due to a comm path failure.

When you define your comm paths, you establish a priority for each path.LifeKeeper uses the paths in priority order for inter-node communication. However, LifeKeeper sends heartbeat signals over all active comm paths. These are the comm paths supported by LifeKeeper and the default priority range assigned to each:

- **TCP/IP (socket)**. The LAN comm path is given the highest priority. The default priority range for the socket path is 1 to 30. You can define multiple LAN comm paths between a pair of servers by defining each one over a different adapter to prevent false failovers.

  Note: The LifeKeeper GUI uses TCP/IP for communicating status information about protected resources; if there are two TCP/IP comm paths configured, LifeKeeper uses the comm path with the highest priority for communicating resource status.

- **Shared disk**. LifeKeeper allows you to define a raw disk partition on a shared disk as a communication location for a pair of servers in the cluster. The shared disk path must be identified with the same drive letter on both servers and the drive letter must identify the same disk partition. The disk partition is usually small, typically one megabyte. The default priority range for the shared disk comm path is 61 to 99 (not supported in greater than two-server configurations).

  Note: TTY and Shared Disk comm paths are used by LifeKeeper only for detecting whether other servers in the cluster are alive. Therefore if the TCP/IP comm path used by the LifeKeeper GUI is down, the GUI will show hierarchies on other servers in an UNKNOWN state, even if the TTY, shared disk or secondary TCP/IP comm path is operational.

More About the Shared Disk Comm Path

The shared disk comm path can be used as a channel of last resort in the case where all other communication has been severed. If the shared disk comm path were to be lost as well, it is very likely that at least one of the servers would not be able to access the storage subsystem, thereby preventing a "split-brain" situation where both servers may access the same disk resource simultaneously.

CAUTIONS:

- A LifeKeeper configuration should include no more than one shared disk comm path between any two servers in the cluster.
- Before using shared disk comm paths on JBOD or Host-based RAID, be sure to test the comm path for reliability when a member of the cluster is shut down or out of service. Sometimes, in configurations using JBOD or Host-based RAID, the comm path will fail to go down when a cluster member goes down, and therefore a failover is not initiated.

Heartbeat Interval

The LifeKeeper heartbeat interval is the number of seconds between heartbeat signals that verify the servers are alive. The default (and recommended) interval is six seconds.

- If you wish to set the interval at the minimum allowed value of four seconds, then you should ensure that the communication path is configured on a private network and tested thoroughly since values lower than five seconds create a risk of false failovers due to network interruptions.
- The heartbeat interval works in conjunction with the maximum heartbeat misses, which has a default value of five (recommended). Setting the maximum heartbeat misses to a lower number (3 or 4) can create a risk of false failovers. Be sure to test thoroughly in your environment.
- Setting these values too high can effectively disable LifeKeeper's ability to detect a failure.

LifeKeeper Heartbeat

The heartbeat is a key LifeKeeper fault detection mechanism. The heartbeat is a periodic signal sent over the comm path between a pair of servers. The regular signals tell each server that the other is still active. When you define your comm path, the definition sets the heartbeat signal interval in seconds and specifies the number of consecutive heartbeats a server can miss before marking the comm path as dead.

When the LifeKeeper servers mark a comm path as dead, inter-node communications immediately commence over the comm path with the next highest priority. Only when the server fail to receive the heartbeat signal on all comm paths does LifeKeeper initiate the safety check to determine the need for failover recovery.

Safety Check

When all the communications paths on a server are DEAD, LifeKeeper assumes that the paired system is DEAD (or down) and attempts to failover. However,LifeKeeper performs a safety check to ensure that the failure occurred in the server, rather than just the comm paths.

The safety check queries on the network (through LAN Manager) to see if the machine still exists. One of two events can occur:

- **System is alive**. If the check receives a response that the system does exist on the network, it aborts the failover and reports the following message to the LifeKeeper event log:

  ```
  SAFETY CHECK FAILED: COMM_DOWN ABORTED
  ```

- **System is dead**. If the check does not receive a response within a specified time-out period (default 8 seconds), the machine is assumed to be down and the failover proceeds.

LifeKeeper performs this check only once, after all comm paths go down. If the safety check detects that the system is alive, failover is aborted.LifeKeeper does not re-initiate failover until all of the following events happen in sequence:

1. At least one of the comm paths comes back ALIVE.
2. All comm paths again go DEAD.
3. The safety check activates and does not detect that the paired system is alive.

Resource Hierarchies

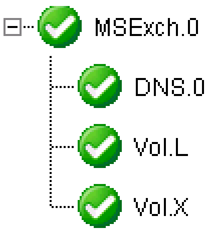
The LifeKeeper GUI enables you to create a resource hierarchy on one server and extend that hierarchy to one or more backup servers. LifeKeeper then automatically builds the designated hierarchies on all servers specified. LifeKeeper maintains hierarchy information in a database on each server. If you use the command line interface, you must explicitly define the hierarchy on each server.

After you create the resource hierarchy, LifeKeeper manages the stopping and starting of the resources within the hierarchy. The following topics provide background for hierarchy definition tasks:

- Resource States
- Hierarchy Relationships
- Shared Equivalencies
- Resource Hierarchy Information

Hierarchy Relationships

LifeKeeper allows you to create relationships between resource instances.The primary relationship is a dependency. For example, one resource instance depends on another resource instance for its operation . The combination of resource instances and dependencies is the resource hierarchy.

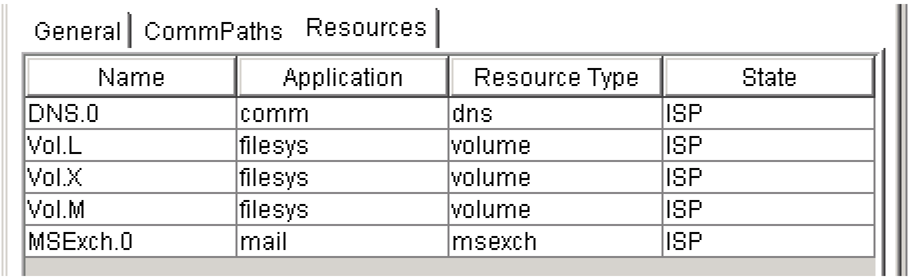


In the example above, MSExch.0 is an Exchange resource, which has three dependencies - a DNS resource (DNS.0) and two volume resources (Vol.Land Vol.X).

The dependency relationships specified by the resource hierarchy tell LifeKeeper the appropriate order for bringing resource instances in service and out-of-service. In the example resource hierarchy, LifeKeeper cannot bring the MSExch.0 resource into service until it successfully brings into service the DNS and volume instances.

Resource Hierarchy Information

A snapshot of information about all the resources defined for a server can be displayed in the Server Properties Dialog Box.



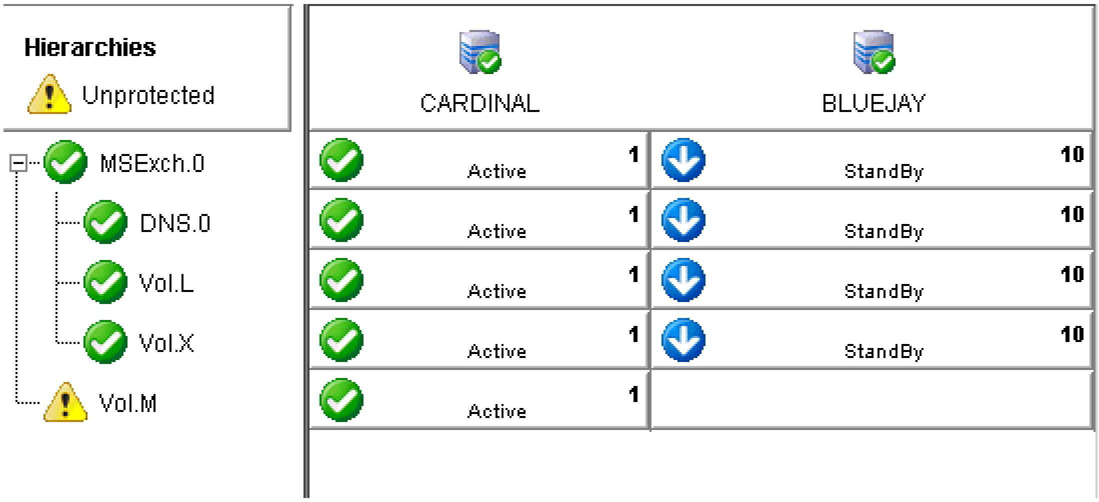
Name	Application	Resource Type	State
DNS.0	comm	dns	ISP
Vol.L	filesys	volume	ISP
Vol.X	filesys	volume	ISP
Vol.M	filesys	volume	ISP
MSExch.0	mail	msexch	ISP

Other resource information can be viewed in the Status Table (main GUI window) or in the Viewing Resource Properties topics.

Resource States

The LifeKeeper GUI status display shows the resources that are defined across all servers to which it is connected. The left pane of the status window displays the Resource Hierarchy Tree which reflects the global resource status (that is, the status of the resource across all servers).

The right pane of the status window contains columns showing the status of each individual resource on each server.



Hierarchies ⚠ Unprotected	CARDINAL		BLUEJAY	
✓ MSExch.0	✓ Active	1	↓ StandBy	10
✓ DNS.0	✓ Active	1	↓ StandBy	10
✓ Vol.L	✓ Active	1	↓ StandBy	10
✓ Vol.X	✓ Active	1	↓ StandBy	10
⚠ Vol.M	✓ Active	1		

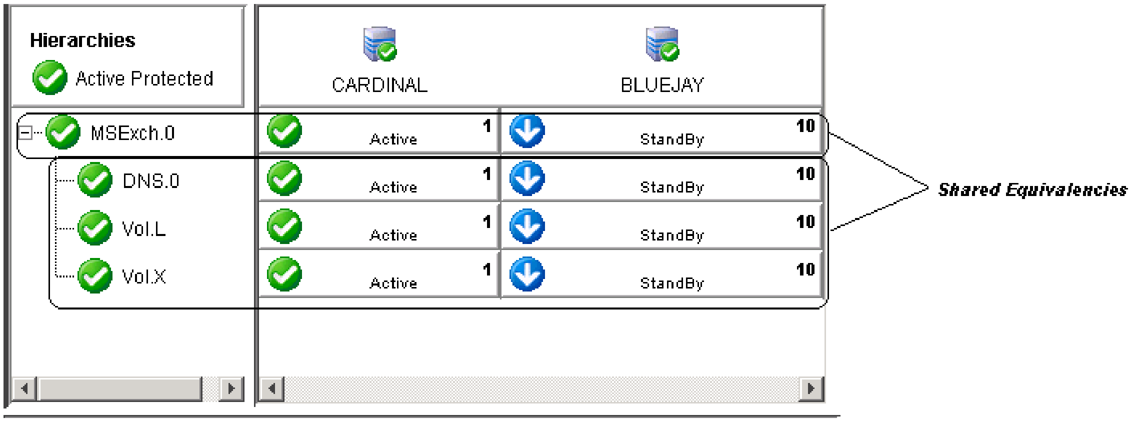
The sample shows above a hierarchy MSExch.0 with a status of In Service,Protected (ISP). The resource Vol.M exists only on CARDINAL. Thus it is In Service, but it has no failover protection, which is indicated by the yellow triangle.

For more details on resource states, see Viewing the Status of Resources.

Shared Equivalencies

When you create a LifeKeeper resource hierarchy, you create the hierarchy initially on the primary server and extend the hierarchy to a backup server. Most resource instances can be active on only one server at a time. For such resources, LifeKeeper defines a second kind of relationship called a shared equivalency that ensures that when the resource is in-service on one server, it is out-of-service on the other servers on which it is defined.

In the example below, a shared equivalency exists between each hierarchy level on a pair of servers. For example, the MSExch.0 resource exists on both servers, and there is a shared equivalency between the two instances of MSExch.0 (just as there is between the one DNS instance and the two volume instances).



Hierarchies ✓ Active Protected	CARDINAL		BLUEJAY	
✓ MSExch.0	✓ Active	1	↓ StandBy	10
✓ DNS.0	✓ Active	1	↓ StandBy	10
✓ Vol.L	✓ Active	1	↓ StandBy	10
✓ Vol.X	✓ Active	1	↓ StandBy	10

Chapter 2: Installation

Planning Your LifeKeeper Environment

This section will assist you in defining your LifeKeeper cluster environment enabling you to successfully achieve your high availability goals quickly and effectively.

Planning Server Communication

Determine and document server communication in a configuration map similar to the one below, using the following guidelines:

- Cluster requirements - To avoid a single point of failure, LifeKeeper requires at least two communication paths (also called "comm paths" or "heartbeats") between servers in a cluster. See **Communication Path Considerations** below for more details.

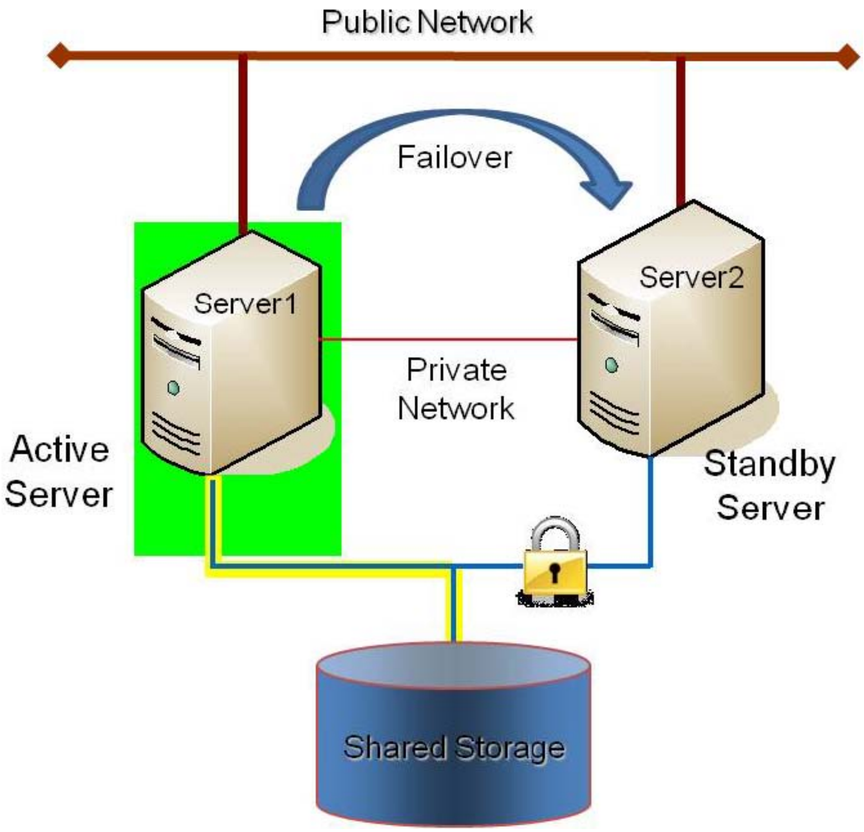


Figure 1 - Sample Configuration Map for LifeKeeper Pair

This is a very simple configuration map depicting a pair of LifeKeeper servers sharing a disk array subsystem. Under normal circumstances, Server 1 runs the application(s) and is considered the primary or active server. Server2 is the secondary or standby server. In this case, there is no contention for disk resources because only one server at a time reserves an entire volume of the disk array.

This sample cluster also shows TCP/IP communication paths configured on the public network and on the private network. On your configuration map, label the IP addresses associated with each TCP/IP comm path.

A pair of servers is the simplest LifeKeeper configuration. When planning a cluster consisting of more than two servers, a configuration map is even more critical to ensure that the appropriate connections exist between and among servers. Each server must have a physical communication path to every other server in the cluster in order to provide cascading failover capability.

Note: If using replicated storage rather than shared storage, refer to SteelEye DataKeeper for additional information on configuring hardware and software for replication.

Communication Path Considerations

LifeKeeper comm paths are used to communicate the state of protected resources in a cluster and to manage failovers. Each comm path is assigned a priority number with the lowest number designated as the “highest”priority.

The recommended configuration is two separate LAN-based (TCP/IP) comm paths configured on independent subnets. The primary comm path should be configured on the private network. **A switchable IP address should not be configured on the Network Interface Card (MIC) carrying the primary comm path**.

Redundant Comm Paths

LifeKeeper strongly recommends redundant comm paths whenever possible. If a single comm path is used and that comm path fails, then LifeKeeper hierarchies may come into service on multiple systems simultaneously. This is known as a false failover or a “split-brain” scenario. In the split-brain scenario, each server believes it is in control of the application and thus can access and write data to the shared storage device.

Primary Comm Path (Private Network)

A private TCP/IP comm path provides reliable communication between systems that is not affected by any communication occurring on the public network. For this reason, it is recommended that the primary comm path be configured on a private network and the secondary comm path on the public network.

TCP/IP comm paths are configured in LifeKeeper using static IP addresses and subnet masks. The cabling may consist of either a crossover cable for a two-node cluster or a small hub for clusters of three or more nodes.

Note: It is very important that private network connections are not registered with DNS. DNS should normally publish only the public network connection for each server. This is essential when connecting a local LifeKeeper GUI admin client to a remote LifeKeeper system. Refer to Verifying Network Configuration for network configuration details.

Recovery Kit Requirements

Each of the LifeKeeper recovery kits has requirements that you should consider in planning and connecting all the components of your LifeKeeper cluster. While the *LifeKeeper Release Notes* provides technical requirements for each kit such as program versions and disk space requirements, you will find detailed configuration information in the Recovery Kit section.

The Core recovery kits (Volume, IP, LAN Manager, File Share, DNS, Microsoft Internet Information Services (IIS), and Generic Application) are documented throughout this wiki documentation site.

Note: All separately packaged (optional) LifeKeeper recovery kits require a software license key in order to function with LifeKeeper v4.3 and higher. You can install the license key by running the **License Key** utility from ***Start->All Programs->SteelEye->LifeKeeper->License Key Installer***.

Storage and Adapter Requirements

LifeKeeper configurations may use the facilities of shared SCSI host adapters and shared disk hardware to switch resources from a failed server to a designated backup server. A Fibre Channel Storage Area Network (SAN) may also be used to switch resources from a failed server to a designated backup server.

Determine your storage and host adapter requirements using the following guidelines:

Storage Devices - Based on your application's data storage requirements,you will need to determine the type and number of data storage devices required by your configuration. Your shared files should reside on a disk array subsystem (Redundant Array of Inexpensive Disks, or RAID). LifeKeeper supports a number of hardware RAID peripherals for use in LifeKeeper configurations. The primary requirement is that the device is supported by Microsoft. See the Microsoft Hardware Compatibility List.

IMPORTANT: Consider the following issues when planning the configuration of your storage devices:

- LifeKeeper manages resources at the volume level, making the resources on each volume available to only one server in the configuration at a time. As a result, it is a good idea to plan disk allocations before you begin to configure LifeKeeper.

Adapters - Based upon the type of configuration and the number of peripherals, determine the types and number of SCSI or Fibre Channel Host Adapters required. It is important that any adapter you choose be supported by Microsoft so that there is a driver available. See the "Cluster" categories in the Microsoft Hardware Compatibility List for Microsoft-supported adapters and peripherals.

For reference purposes, you should add the host adapter specifications to your configuration map.

Verifying Server Specifications

Ensure that you have the correct version and/or capacity of the following components for each LifeKeeper server:

- Windows 2003 R1 and R2 Operating System (32- or 64-bit)

Note: If you plan to use the LAN Manager Recovery Kit, be sure that the File and Printer Sharing for Microsoft Networks component is installed. This component is installed and enabled by default.

- Windows 2008 R1 and R2 Operating System (32- or 64-bit)

  Note: File and Printer Sharing is enabled for Lan Manager AND for use with DataKeeper replicated volumes. During the installation procedure, LifeKeeper can automatically configure the Windows 2008 firewall so that ports it needs are opened, and so that ICMP is enabled.

  Note: The Local Security Policy "Network Access: Let Everyone permissions apply to anonymous users" must be Enabled if you plan to use DataKeeper replicated volumes with LifeKeeper. This policy will be enabled by LifeKeeper installation.

  Note: By default, firewall is enabled. During installation, if a firewall is detected, the appropriate rules will be added to windows firewall. However,if the firewall is disabled during installation and re-enabled at a future time, the setup firewall script needs to run to add the rules. This script is installed as ***%LKROOT%\support\firewallSetup.bat***. To run the command from the command line, type `firewallSetup.bat%LKROOT%\jre1.5`.

- Ethernet TCP/IP-supported network interface card(s) for LAN-based cluster heartbeat(s)

- Disk arrays and storage adapters (SCSI or Fibre Channel) if you are using shared storage.

- Memory. See the *LifeKeeper Release Notes* for minimum memory requirements for LifeKeeper.

  Note: Additional memory (beyond that required for LifeKeeper)is required to run user applications.

- Disk space. See the *LifeKeeper Release Notes* for minimum disk space requirements for LifeKeeper and recovery kits.

- LifeKeeper Graphical User Interface (GUI) platforms and browsers

- Power Requirements. To maximize the availability of your LifeKeeper servers, it is strongly recommended that you use Uninterruptible Power Supplies (UPSs), or at a minimum, separate the electrical sources to your servers.

- Application software to be protected by LifeKeeper.

Determine the server names, processor types, memory and other I/O devices for your configuration. When you specify a backup server, you should ensure that the server you select has the capacity to perform the processing should a failure occur on the primary server.

Setting Up Your LifeKeeper Environment

Now that you have determined your requirements and mapped your LifeKeeper configuration, you can start setting up the components of your LifeKeeper environment.

The major tasks of this topic are:

Configuring Your Storage

Safe Creation of Shared Disk Volume Instances

Verifying Network Configuration

DNS Resource Requirements

Installing and Setting Up Database Applications

Note: Although it is possible to perform some setup tasks in a different sequence, this list provides the recommended sequence.

Configuring Your Storage

LifeKeeper may be used with either shared storage or with replicated storage. Follow the instructions that apply to your configuration below.

Shared Storage Configuration

If you are using shared storage, then after your Windows environment is installed, you should set the host adapter and shared peripheral addressing. Refer to the documentation accompanying your adapter and storage device for specific details. Perform the following tasks to configure your shared storage for access by all servers in the LifeKeeper cluster:

1. Because all disks placed under LifeKeeper protection must be partitioned, your shared disk arrays must now be configured into partitions (volumes) using the Windows Disk Management utility. You should also format the partitions with the NTFS file system.

   Note: To safely configure your shared storage, it is recommended that you follow the procedure in Safe Creation of Shared Disk Volume Instances.

   You should refer to your disk array software documentation for detailed instructions.

2. If you plan to use a Shared Disk comm path, designate a small raw (unformatted) partition to use for the comm path. One MB should be a sufficient size.

3. Power on the other server(s) in the cluster and verify that all servers recognize the shared disks. From the backup server(s), make drive assignments for the shared volumes exactly the same as the first server. It is recommended that you have the Disk Management utility open on only one server at a time.

4. If you have created file shares on the shared volumes, you will need to turn on the file sharing attribute of these folders on each server in the cluster.

Replicated Volume Configuration

If you are using SteelEye DataKeeper for Windows, create your disk partitions (volumes) to be replicated using the Windows Disk Management utility. You should also format the partitions with the NTFS file system.

Be sure to assign the same drive letter to the source volume (on the primary server) and target volume (on the backup server).

Verifying Network Configuration

It is important to ensure that your network is configured and working properly before you install LifeKeeper. There are several tasks you should do at this point to verify your network operation:

1. You must ensure that every network interface card (NIC) has one permanent IP address in order to create a TCP/IP comm path or protect an IP address.
2. If your server has more than one NIC (recommended), you should configure them to be on different subnets. If the adapters are on the same subnet, TCP/IP cannot effectively utilize the second adapter.
3. Your IP addresses should be configured as follows, assuming at least two NICs in each server (one on a private network and one on the public network):

    a. In the **Control Panel**, click on **Network Connections**. Right-click **Open**.

    b. From the **Advanced** menu, select **Advanced Settings**.

    c. Ensure that the NIC connected to the public network is in the topmost position of the **Connections** list. This network card should be associated with the highest priority comm path, which will be used by the LifeKeeper GUI.

    d. Do not register private network connections with DNS. Uncheck the **Register this connection's address with DNS** checkbox for the private network adapter as follows:

    Internet Protocol (TCP/IP) Properties-> Advanced -> DNS Tab

    Since no DNS servers are needed for the private network connection, none should be listed.

    e. If you are running LifeKeeper on a Windows domain controller, after saving and closing the Control Panel, execute the following command:

    ```
    net config Server /hidden:yes
    ```

    This prevents the browser from occasionally getting confused when switching over LAN Manager computer names.

4. From each server, ping the local server, and ping the other server(s) in the cluster. If the ping fails, then do the necessary troubleshooting and perform corrective actions before continuing.
5. To ensure that the LifeKeeper GUI server and client components can effectively communicate ensure that *localhost* is resolvable by each server in the cluster.

    - If DNS is not implemented, edit the ***%windir%\system32\etc\drivers\hosts*** file and add an entry for the localhost name. This entry can list either the IP address for the local server, or it can list the default entry (127.0.0.1). If *localhost* is not resolvable, the LifeKeeper GUI may not work.
    - If DNS is implemented, verify the configuration to ensure the servers in your LifeKeeper cluster can be resolved using DNS.

6. Ensure each server’s hostname and networking addressing information is correct and will not change after LifeKeeper is installed. If changing the hostname after LifeKeeper is in operation, you must run the **lk_chg_value** utility to modify the computer name in the LifeKeeper configuration files. If changing the networking configuration after LifeKeeper is in operation, you must run the **lk_chg_value** utility to modify existing LifeKeeper comm paths and resource hierarchies after re-configuring your network information.

Note: If you are using SteelEye DataKeeper for Windows, refer to the SteelEye DataKeeper section of the documentation for additional information on specifying the network cards to be used for replication and comm path considerations.

Switchable IP Address

Most LifeKeeper configurations use the IP Recovery Kit, which defines a switchable IP address. A switchable IP address is a “virtual” IP address that can be switched between servers. It is separate from the IP address associated with the network interface card of each server. Applications under LifeKeeper protection are associated with the switchable IP address. Then, if there is a failure on the primary server, the switchable IP address “switches” to the backup server.

If you plan to configure resource hierarchies for switchable IP addresses, you must do the following on each server in the cluster:

- Verify that the computer name is correct and will not be changed.
- Verify that the switchable IP addresses are unique using the `ping` command.
- If you wish to assign a hostname to the switchable IP address, you must edit the ***%windir%/system32/etc/drivers/hosts*** file on each server to add an entry for each switchable IP address and associated hostname.

Note: If using teaming software or if network cards are changed after creating a switchable IP resource, the switchable IP resource should be deleted and recreated as the associated index number for the card can change.

Note: By default, network broadcast pings are used to verify network presence before failing over an IP resource. If no broadcast pingable equipment is on your network or if you prefer not to test for network presence this way you may disable this test by setting the following registry value to 0 (disabled). Reinstalling LifeKeeper will restore the default value of 1(enabled).

```
HKEY_LOCAL_MACHINE\SOFTWARE\Steeleye\LifeKeeper\RK\IP\BroadcastPing
```

Switchable IP Addresses, DNS and LifeKeeper GUI Considerations

Special network considerations must be made when a "virtual" IP address is used on the server's main NIC and DNS registration is enabled on the NIC. When a "virtual" IP address is created by LifeKeeper on a registered NIC, DNS will add this additional IP address for the server and start using it for host name resolution on the network. However, LifeKeeper protected "virtual" IP addresses are switchable across cluster nodes. Therefore, precautions must be taken to prevent the LifeKeeper GUI from also using DNS registered "virtual" IP addresses to get updates from local and remote cluster nodes.

To keep LifeKeeper GUI connections to local and remote systems stable when using "virtual" IP addresses, there are two options:

1. Use a network *hosts* file on each LifeKeeper node.

    - In the *hosts* file, identify the permanent IP address for every other remote cluster node.
    - Do this on every LifeKeeper system in the cluster.

    As explained above, these addresses must be on the highest priorty network used for LifeKeeper GUI binding.

2. Use an alternate network and associated alternate NIC for LifeKeeper GUI connections to all other nodes in the cluster. This option differs from the simpler recommendations explained above.

    - Enable DNS registration on the alternate network and NIC.
    - Make the alternate network the highest priority in the *Network Connections -> Advanced -> Advanced Settings* selection in the **Adapters and Bindings** tab.
The LifeKeeper GUI will use this highest binding network.
    - The highest priority LifeKeeper comm path should also use this network.
    - Do this on every LifeKeeper system in the cluster.

    The LifeKeeper GUI will use this alternate network for connections to all cluster nodes. With no virtual IPs assigned to this alternate network, the address registration will be stable. DNS registration may also be used for the main/public NIC on the server as needed.

Note: After making network configuration changes, the "`ipconfig /flushdns`" command may be used to remove obsolete cached DNS information.

IP Local Recovery Configuration

LifeKeeper provides the ability to monitor local switchable IP addresses and move them to another network adapter in the same system when a failure is detected. This functionality, called IP Local Recovery, imposes additional requirements and limitations on the system configuration.

The backup adapter, also known as the Local Recovery Adapter where the switchable address will become active after a failure of the primary adapter,must be configured as follows:

- Both adapters must be connected to the same physical subnet.
- For routing purposes, all addresses on the Local Recovery Adapter must be on a different logical subnet than any permanent addresses on the Primary adapter. They must also be on a different logical subnet than any LifeKeeper-protected switchable addresses that are configured on the Primary adapter.
- Cabling and network routing must be configured to permit a ping command issued from either logical subnet to reach the protected IP address and its associated subnet when it is placed on either the primary network card or the local recovery network card. This can be verified by manually issuing a ping command from other systems on each logical subnet. A failed ping

command indicates a network routing problem.

- IP Local Recovery can only be enabled at the time the IP resource is created. Local Recovery cannot be added to an IP resource by modifying its resource attributes after the resource has been created.
- IP Local Recovery may be disabled for an IP resource by using the "`ins_setlocalrecovery`" command line utility. This utility is located in the LifeKeeper ***bin*** directory (***C:\LK\bin*** by default). From a command prompt, type "`ins_setlocalrecovery`" for the usage and switch options.

How IP Local Recovery Works

When IP Local Recovery is enabled, and the IP resource fails its deepcheck (a periodic extensive check of the IP resource) then LifeKeeper will do the following:

- First, LifeKeeper will attempt to bring the IP address back in-service on the current network adapter.
- If that fails, LifeKeeper will check the resource instance to determine if there is a backup (Local Recovery Adapter) available. If so, it will then attempt to move the IP address to the backup interface.
- If all local recovery attempts fail, LifeKeeper will perform a failover of the IP address and all dependent resources to a backup server.

Even if you do not have a backup adapter, you can enable Local Recovery so that LifeKeeper will retry the primary network interface before initiating failover to a backup server.

DNS Resource Requirements

The DNS Recovery Kit included with the LifeKeeper for Windows Core product provides a mechanism to update DNS A and PTR records of the primary server or an alias name on the DNS servers in your configuration. The DNS resource allows the user to select the A record of the primary server or an alias name in DNS which will be modified along with the PTR record (if exists) with the IP address of a backup server when failover or switchover occurs. Using a DNS resource allows clients to connect to the servers in a WAN environment when a failover or switchover occurs. When LifeKeeper servers are in different network subnets, it is not possible to use a switchable IP address. In this type configuration, a DNS resource should be used to provide client connectivity. For details on creating DNS resources, refer to Creating a DNS Resource Hierarchy.

Restriction: LifeKeeper servers should not be configured as Domain Controllers or DNS Servers. Creating a DNS resource that points to a DNS server on the same system will fail with the following error message: “`User credentials cannot be used for local connections.`”

TTL of DNS Records

When the LifeKeeper for Windows DNS Recovery Kit updates the A record of the primary server or alias name in DNS, the A record on the caching DNS servers' cache is not updated. These caching DNS servers are those who do not hold the zone that the LifeKeeper protected A record belongs to. The A record in the cache stays until the TTL is expired or the cache is cleared manually. Therefore, the clients of those caching DNS servers will not get the updated value of the A record in timely fashion. For LifeKeeper protected DNS resources, it is recommended that the TTL value of the A record of the primary server or alias name should be set to a lower value.

If LifeKeeper creates the A and PTR records for a DNS resource, the TTL of those records is set to 5 minutes. This value can be changed using the Microsoft DNS management console (dnsmgmt.msc). However, changing the value to a higher value will make the A record live in the cache longer on caching DNS servers.

For DNS A and PTR records created prior to creating the LifeKeeper DNS resource hierarchy, it is recommended that the TTL value be set to a lower value like 5 minutes.

Installing and Setting Up Database Applications

If your environment includes a protected database application such as SQL Server or Oracle, you should install the application using the documentation provided with the database. Ensure that the database is on a shared or replicated file system and that the configuration files are on a shared or replicated file system. The executables may either be on each local or a shared file system. Refer to the specific LifeKeeper Database Recovery Kit for additional installation and setup considerations.

Although it is possible to install your application after LifeKeeper is installed, you should test the application to ensure it is configured and operating properly before placing it under LifeKeeper protection.

Installing LifeKeeper

If you have completed planning and setting up your LifeKeeper environment, you should be ready to install the LifeKeeper software on each server in your cluster.

LifeKeeper Core Software

The LifeKeeper Core software is available on the LifeKeeper for Windows CD or via ftp download. The LifeKeeper Core is comprised of:

- The basic LifeKeeper software, including:
    - Perl (CPAN v5.8.8)
    - Cygwin
    - International version of Java Runtime Environment (JRE) v1.5.0 Update 6
    - SUperior SU 2.0.0.6 and SUperior SU Patch 2.0.0.18
    - LifeKeeper GUI (both server and client)
    - Microsoft Visual C++ 2008 Redistributable package (v 8.0.56336
- Core recovery kits:
    - Volume
    - IP

- DNS
- LAN Manager
- File Share
- Generic Application
- Internet Information Services (IIS)

Installing the LifeKeeper Core Software

LifeKeeper uses the Flexera InstallShield product to provide a standard installation interface. A license must be obtained and installed for each server in the cluster. We recommend that you read the *LifeKeeper Release Notes* before installing and configuring LifeKeeper.

To install LifeKeeper Core, run the setup program delivered with the LifeKeeper for Windows product. Follow the Setup instructions on each screen. Some explanatory notes are included below.

Installation Notes

- You must have administrative privileges to install the LifeKeeper software. While non-administrative users will not be prohibited from running the setup program, the installation will exit immediately due to lack of special permissions required during setup.
- Installing LifeKeeper on your shared storage is **not** supported. Each server should have its own copy installed on its local disk.
- The SUperior SU installation is called from the LifeKeeper installation program.
- The default LifeKeeper installation path is *C:\LK*. You may change this path, but due to some scripting issues, **be sure to choose a path with NO EMBEDDED SPACES and containing eight characters or less**. For instance, *C:\Program Files\LK* and *C:\LifeKeeper* are invalid choices that will result in application errors.
- Two Windows registry changes are made during the installation of LifeKeeper: *DisableStrictNameChecking* and *DisableLoopbackCheck*. Both of these changes are required to allow access to servers using an alias name.

Setup Type

Choose one of the following:

- **Typical** installs the LifeKeeper Core and all Core recovery kits(recommended). **Note**: DHCP Media Sense for TCP/IP will be disabled by default.
- **Compact** installs the LifeKeeper Core only (which includes the Volume Recovery Kit).
- **Custom** allows you to select from the list of LifeKeeper components to install: Core files (always required), IP Recovery Kit, DNS Recovery Kit, LAN Manager Recovery Kit, File Share Recovery Kit, Generic Application Recovery Kit and IIS Recovery Kit. The Custom option will ask the following questions:

- “Disable DHCP Media Sense for TCP/IP?”
- “Do you wish to start the LifeKeeper Services?” See Starting LifeKeeper Services below for details.

Firewall Change Prompt (Windows 2008 Systems)

LifeKeeper cannot function properly if the firewall settings for the source and target machines are not configured correctly. During installation of LifeKeeper, you will be prompted to allow the installer to configure your firewall rules needed by LifeKeeper, as well as to configure other system settings that are required by LifeKeeper. If you choose to allow the installer to make these changes, you will not need to configure your firewall manually. Please refer to Troubleshooting for more information.

LifeKeeper requires the following ports / protocols / processes to be open or enabled:

TCP Ports: 81, 82, 1500, 3278, 3279

Processes: %LKROOT%\jre1.5\bin\java.exe

Protocols: ICMP Echo

Starting LifeKeeper Services

If you choose the **Custom** installation option, you will be asked, “Do you wish to start the LifeKeeper Services?” In most cases you should answer **Yes** so that LifeKeeper will be started automatically when the system is booted. Answering **No** will cause LifeKeeper not to be started after installation, and it will set the **Startup Type** for the LifeKeeper services to **Manual**.

If you select **No**, and you later wish to start the LifeKeeper services, you should do so using the **Services** tool in the **Windows Control Panel**. (You should start both LifeKeeper and LifeKeeper External Interfaces). In addition, you can set the **Startup Type** to **Automatic** by right-clicking on each service and selecting **Properties**, then changing the **Startup Type** option to **Automatic**. This will tell LifeKeeper to always start at system boot time.

Question: In what situation would it make sense to answer **No** to starting the LifeKeeper services?

Answer: Choosing not to start the LifeKeeper services may be useful in a staging environment where you are not ready to configure your network addresses but you wish to install LifeKeeper and replicate it across a number of systems prior to final installation of the cluster.

Explanation: When LifeKeeper is started the FIRST time, the system’s network configuration information is written into the **LifeKeeper Configuration Database** (LCD). Changing your network configuration AFTER LifeKeeper is started requires deleting and re-creating your comm paths and resource hierarchies. Therefore, by choosing NOT to start the LifeKeeper services at install time, you can install LifeKeeper and associated recovery kits, then configure your network later.

SUperior SU installed with LifeKeeper Core

The LifeKeeper for Windows core product installs the SUperior SU 2.0.0.6 and Patch 2.0.0.18 software by Stephan Kuhr. The SUperior SU software provides a robust switch user utility currently available at no charge from Stephan Kuhr on the Internet at http://www.stefan-kuhr.de/cms/index.php?option=com_content&view=article&id=62&Itemid=73. The SUperior SU service is disabled during the install of the software since the service is not currently needed by the LifeKeeper core or its recovery kits.

LifeKeeper recovery kit scripts are executed using the Windows “Local System” account, which has no standard ID or password associated with it and by default has no desktop privileges either. Some applications protected by LifeKeeper require queries and other operations to monitor and manage them. To perform these operations as needed without user intervention, some LifeKeeper recovery kits must assume the role of a valid user during the restore and monitoring processes. The SUperior SU software provides a programmatic “Switch User” or “Run As User” utility program that allows the recovery kit to perform these operations without user intervention. User accounts to be used for monitoring purposes by LifeKeeper must have login privileges that are valid on every system where the protected application may be placed in service.

Note: Removal of LifeKeeper software does NOT uninstall SUperior SU. SUperior SU can be removed separately. The SUperior SU patch should be removed prior to uninstalling the SUperior SU software. You can contact SIOS Technology Corp. support for assistance in uninstalling the Superior SU program on a Windows 2008 environment. However, you can use the **Add/Remove Programs** feature to remove SUperior SU in a Windows 2003 environment.

Silent Installation of LifeKeeper for Windows

You can install LifeKeeper for Windows silently through the use of the `-silent` command line option. This option suppresses both the wizard and launcher user interfaces (UIs) resulting in what is considered a "silent installation." This is how an installation is performed without any information displaying to or requiring any interaction with the end user. Response files, also known as "*options*" files, are used to pass command-line options at installation. This is done as you would normally specify them on the command line to represent the responses to dialogs and/or to set the value of a property or variable. The options specified in the response/options file are executed after the execution of the options that were entered directly on the command line.

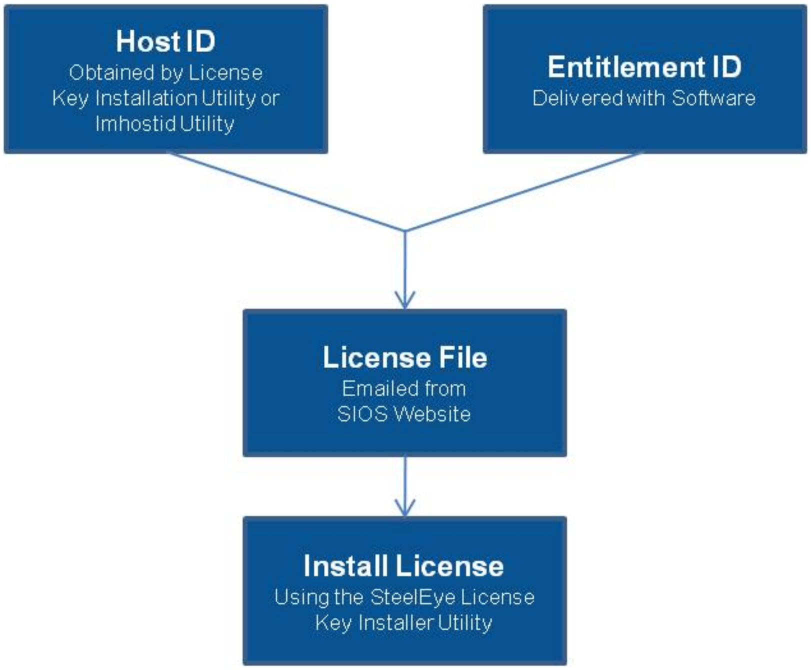
To create a response file, open a command window and run the LifeKeeper setup program using the command **`LK-{version}-Setup.exe –r /f1C:\setup.iss`**. The responses entered to the dialogs will be recorded into the file ***setup.iss***.

To perform a silent install using the created response file, open a command window and run the LifeKeeper setup program using the command **`LK-{version}-Setup.exe –s /f1C:\setup.iss /f2C:\setup.log`**. Results from the silent install are stored in the file***setup.log***. “Result Code=0” indicates a successful install.

Obtaining and Installing the License

SteelEye LifeKeeper requires a unique license for each server. The license is a run-time license which means that you can *install* LifeKeeper without the license, but the license must be installed before you can successfully *start* and *run* LifeKeeper.

The final screen of the InstallShield installation utility displays the Host ID of your server. The **Host ID**, along with the **Entitlement ID** (Authorization Code) that was provided with your SteelEye LifeKeeper software, is used to obtain the license required to run SteelEye LifeKeeper. The process is illustrated below.


Host ID
Obtained by License Key Installation Utility or lmhostid Utility
Entitlement ID
Delivered with Software
License File
Emailed from SIOS Website
Install License
Using the SteelEye License Key Installer Utility

License Key Manager

In addition to installing LifeKeeper product licenses, the **License Key Manager** allows you to perform the following functions:

- View all licenses currently installed on your system.
- View all expiration notifications (days remaining) for each time-expiring license.
- Identify invalid licenses that are currently installed.
- Delete any installed licenses (right-click on the license and select **Delete**).
- Delete all expired licenses as a group (press the **Delete Expired License** button).
- **Refresh** the Installed License list when installing software or upgrades.

Perform the following steps to obtain and install your licenses for each server in the LifeKeeper cluster:

1. Get your **Host ID**. At the end of the LifeKeeper installation, make note of the **Host ID** displayed by the **License Key Installer** utility as shown below. The Host ID may also be obtained by running `%LKROOT%\bin\lmhostid` (where *%LKROOT%* is the LifeKeeper installation path, by default *C:\LK*) on the system(s) that you are obtaining licenses for. (If you need to obtain your Host ID again at a later time, you may do so by running the **License Key Installer** utility from the **Start-Programs** menu **Start-All Programs-SteelEye-LifeKeeper-License Key Installer**.)

2. Write the **Host IDs** in a notebook or save them in a file. If saved in a file, copy that file to a system with internet access. Otherwise, take your notebook with you to the system with internet access.

3. Ensure you have your LifeKeeper **Entitlement ID** (Authorization Code). You should have received an email with your software containing the Entitlement ID needed to obtain the license.

4. Obtain your licenses from the SIOS Technology Corp. Licensing Operations Portal.

   a. Using the system that has internet access, navigate to the SIOS Technology Corp. Licensing Operations Portal and log in entering your **User Name** and **Password**.

   b. Select **Manage Entitlements**.

      Note: If changing password, use the **Profile** button in the upper right corner of the display.

   c. Find your **Entitlement ID** and select each **Activation ID** associated with that Entitlement ID by checking the box to the left of the line item.

   d. Select the **Activate** tab.

   e. Define the required fields and select **Next**.

   f. Click on **Select Existing Host** to choose an already defined host or create a new host by selecting **Add New Host**.

   g. Enter the **Host ID** and click **Okay**.

   h. Check the box to the left of the **Host ID** and select **Generate**. The **Fulfillment ID** will display on the **License Summary** screen.

   i. Check the box to the left of the **Fulfillment ID** and select the **Email License** tab.

   j. Enter a valid email address to send the license to and select **Send**.

   k. Select **Complete**.

   l. Retrieve the email(s).

   m. Copy the file(s) to the appropriate system(s).

5. Install your license(s). To install your license(s), choose one of the following options. **Note**: If you received your license after July 23, 2010, use **Option B**.

   a. Install via the **License Key Installer**.

- On each system, run the **License Key Installer** from the **Start-Programs** menu (**Start-All Programs-SteelEye-LifeKeeper-License Key Installer**).
- Press the **Install License File…** button on the main screen of the **License Key Installer**.
- Browse to the location of the license file that you saved in **Step 4** above.
- Click on the license file name. It will become highlighted.
- Press the **Install License File…** button that appears in that dialog box below the file names. A license detection confirmation popup will be displayed.

or

b. Copy the license file(s) to the appropriate directory manually.

On each system, copy the license file(s) to *%windir%\system32\LKLicense* on x86, or *%windir%\SysWOW64\LKLicense* on x64 (where *%windir%* is the Windows installation path, by default *C:\Windows*). If the *LKLicense* directory does not already exist, it will need to be created prior to copying the files. **Note**: It is recommended that this file(s) be renamed to *YYYYMMDD.lic* format to distinguish the day the license was activated.

6. Repeat on all additional servers. You must install a license on the other LifeKeeper server(s) using the unique Host ID for each server.
7. Reboot your system.

Primary Network Interface Change May Require a License Rehost

The Host ID used by the License Key Installer utility is obtained from the LifeKeeper server's primary network interface card (NIC). LifeKeeper will check for a valid license each time it starts. If your LifeKeeper server should require a NIC replacement in the future that would cause the Host ID to change, then the next time LifeKeeper is stopped, a License Rehost must be performed before starting LifeKeeper again. Log in to the SIOS Technology Corp. Licensing Operations Portal and select **Support Actions/Rehost** from the **Manage Licenses** screen to perform this rehost. (**Note**: A rehost can be performed one time per six-month period without contacting support.)

Subscription Licensing

A subscription license is a time-limited license with renewal capability. Similar to an evaluation license, it will expire after a set amount of time unless renewed. This renewal process can be set up to renew automatically by following the procedure below:

1. Install the subscription license program: `%LKROOT%\bin\lmSubscribe.exe`
2. Enter **User ID** and **Password** (from **SIOS Technology Corp. Customer Registration**). These credentials are saved in an encrypted file.
3. Select **OK**.

If the previous steps run successfully, the subscription renewal service will now run, in the background, periodically checking renewal status. If licenses are found that will be expiring in a certain number of days (90, 60,30, 20, 10, 5, 4, 3, 2, 1), a warning notification will be sent to the **Windows Event Viewer**, and an attempt will be made to renew the license. If a new license activation is available (a new activation has been purchased for this system's Entitlement), it will be automatically fulfilled and the new licenses will be installed on the system replacing the old licenses. As long as licenses for this system are renewed (activations purchased), the service will ensure that the licenses are upgraded on the system without user intervention.

Troubleshooting

If errors are encountered, please try the following before contacting support:

- Review the error messages in the **Windows Event Viewer**.
- Verify credentials by logging in to the SIOS Technology Corp. Licensing Operations Portal. Enter **User ID** and **Password**. Run `%LKROOT%\bin\lmSubscribe.exe` again using the correct **User ID** and **Password**.
- To force a manual check for a license renewal, stop and restart the service. (**Note**: To find the service, bring up the view for all of the Windows services and search for “**SteelEye Subscription Licensing**”.)
- If ownership of the license certificate has changed, please contact SIOS Technology Corp. support personnel to have the certificate moved to the new owner. Once ownership has been moved, the automatic license renewal service will need to be updated with these new credentials by running the above command again using the new **User ID** and **Password**.

Installing LifeKeeper for Windows Localized Language Supplement

LifeKeeper for Windows Localized Language Supplements are available to support LifeKeeper running in localized environments. Each Localized Language Supplement contains translated LifeKeeper GUI text strings and context-sensitive help in the localized language. For LifeKeeper v4.2, a Japanese Language Supplement was available. For LifeKeeper v7.2.1, language supplements are available for Chinese and Japanese languages. The international version of Java Runtime Environment (JRE) is required to support running the LifeKeeper GUI in localized environments. The LifeKeeper for Windows Core installation program installs the required version of JRE.

For LifeKeeper v7.2.1, the Chinese Localized Language Supplement includes language content for the SteelEye Protection Suite including LifeKeeper and DataKeeper v7.2.1 products. The administrator can select which product to update. However, the Japanese Language Supplement will update only LifeKeeper v7.2.1. A standalone DataKeeper v7.2.1 product will be available in Japanese. Also, the DataKeeper mmc-based GUI requires the Windows language pack be installed unless the complete localized OS is already installed.

The LifeKeeper for Windows Localized Language Supplement, like the LifeKeeper Core, is installed via InstallShield. The installation requires no selection for Typical/Compact/Custom options. To install the LifeKeeper for Windows Localized Language Supplement, run the setup program shipped with the Localized Language Supplement product.

To repair an existing installation of LifeKeeper for Windows Localized Language Supplement, run the setup program and choose **Repair** from the list of InstallShield options.

To remove LifeKeeper for Windows Localized Language Supplement, run **Add/Remove Programs** from the **Windows Control Panel**. The Localized Language Supplement must be removed before removing the LifeKeeper core product.

Installing LifeKeeper Recovery Kits

Recovery kits are generally installed after the application is installed and configured and after the LifeKeeper Core software is installed. For a complete list of currently available recovery kits, see the LifeKeeper for Windows Release Notes or visit http://us.sios.com/.

Recovery kits, like the LifeKeeper Core, are installed via InstallShield. The installation requires no selection for Typical/Compact/Custom options.

Note: All separately packaged (optional) LifeKeeper recovery kits require a software license key in order to function with LifeKeeper v4.3 and higher. You can install the license key by running the LifeKeeper **License Key Utility** from the **Start-Programs** menu.

Reinstalling LifeKeeper

In the event that you need to reinstall the LifeKeeper Core or recovery kits, you should do the following for each server:

1. Bring the active resource hierarchies In Service on a backup server.
2. Exit the LifeKeeper GUI.
3. Stop all LifeKeeper services by opening a command window and entering `$LKROOT\bin\lkstop` (where *$LKROOT* is the LifeKeeper installation path, by default *C:\LK*). Wait until you see “LIFEKEEPER NOW STOPPED” before continuing.
4. Run the setup program shipped with the LifeKeeper Core product.
5. Choose **Repair** from the list of InstallShield options.

The remaining process will resemble a fresh installation.

After installation is complete, bring the resources back In Service on this server.

Upgrading from Previous Versions of LifeKeeper and SteelEye DataKeeper

You may upgrade from previous versions of LifeKeeper for Windows and SteelEye DataKeeper for Windows while preserving your resource hierarchies by using the procedure below.

Upgrade Procedure

The following scenario illustrates the upgrade process when upgrading both LifeKeeper and SteelEye DataKeeper. You should first upgrade LifeKeeper before upgrading SteelEye DataKeeper. The LifeKeeper Services and SteelEye DataKeeper Service will be stopped during the upgrade process. A system reboot is required after upgrading both LifeKeeper and SteelEye DataKeeper.

Given two systems (Sys1 and Sys2), with Sys1 being the primary (active)server, perform the following steps to upgrade to LifeKeeper v7 and SteelEye DataKeeper v7:

Upgrading the Backup Server

1. Exit the LifeKeeper GUI and SteelEye DataKeeper GUI on backup server *Sys2*.
2. Open a command window and enter `$LKROOT\bin\lkstop` (where *$LKROOT* is the LifeKeeper installation path, by default *C:\LK*) to stop all the LifeKeeper services. Wait until you see “LIFEKEEPER NOW STOPPED” before continuing.
3. To upgrade LifeKeeper for Windows on the backup server *Sys2*: Run the setup program to upgrade LifeKeeper for Windows. Click **Yes** to continue upgrading LifeKeeper.
4. The existing LifeKeeper files will be overwritten by the LifeKeeper installation. You should install your new LifeKeeper license using the **License Manager** utility – pre-7.0 LifeKeeper licenses will not work with LifeKeeper7.0. Do not reboot the backup server until SteelEye Data Replication is upgraded to SteelEye DataKeeper.
5. To upgrade SteelEye DataKeeper for Windows on the backup server *Sys2*: Run the SteelEye DataKeeper for Windows setup program. Click Yes to continue upgrading SteelEye DataKeeper. You should install your new DataKeeper license using the **License Manager** utility – SteelEye Data Replication licenses will not work with SteelEye DataKeeper.
6. Upgrade the language supplement package (if required) and any optional recovery kits at this time by running the appropriate installation program.
7. Reboot the backup server *Sys2*.

For additional backup servers in your cluster, follow these steps on each server.

Upgrading the Primary Server

8. Exit the LifeKeeper GUI and SteelEye DataKeeper GUI on primary server *Sys1*.
9. Open a command window and enter `$LKROOT\bin\lkstop` (where *$LKROOT* is the LifeKeeper installation path, by default *C:\LK*) to stop all the LifeKeeper services. Wait until you see “LIFEKEEPER NOW STOPPED” before continuing.
10. To upgrade LifeKeeper for Windows on the primary server *Sys1*: Run the setup program to upgrade LifeKeeper for Windows. Click **Yes** to continue upgrading LifeKeeper.
11. The existing LifeKeeper files will be overwritten by the LifeKeeper installation. You should install your new LifeKeeper license using the **License Manager** utility – pre-7.0 LifeKeeper

licenses will not work with LifeKeeper7.0. Do not reboot the server until SteelEye DataKeeper is upgraded.

12. To upgrade SteelEye Data Replication for Windows on the primary server *Sys1*: Run the SteelEye DataKeeper for Windows setup program. Click **Yes** to continue upgrading SteelEye Data Replication. You should install your new DataKeeper license using the **License Manager** utility – SteelEye Data Replication licenses will not work with SteelEye DataKeeper.

13. Upgrade the language supplement package (if required) and any optional recovery kits at this time by running the appropriate installation program.

Reboot the primary server *Sys1*.

14. Start the LifeKeeper GUI on *Sys1* by clicking **Start**, and then point to **Programs**, then **LifeKeeper**, then **LifeKeeper GUI**, and login to *Sys1*.

Starting LifeKeeper

With a typical installation, LifeKeeper is started automatically when the server is booted. Your applications are brought up in a protected state.

When LifeKeeper starts, it also starts the LifeKeeper GUI Server. The LifeKeeper GUI client is launched from a web browser or from the ***Start->All Programs->SteelEye->LifeKeeper->LifeKeeper (Admin Only)***, and is described in detail in the LifeKeeper GUI section.

Starting and Stopping LifeKeeper Processes

Because LifeKeeper is started automatically when the server is booted, you should not normally need to start/stop LifeKeeper. In the rare event that you need to start or stop LifeKeeper manually, you can do so in one of two ways:

Services MMC Snap-In

You can stop and start LifeKeeper services using the **Services MMC** snap-in under **Administrative Tasks**.

LifeKeeper consists of two services:

- LifeKeeper
- LifeKeeper External Interfaces

Generally, these two services should be stopped and started together. However, since LifeKeeper External Interfaces is a dependency of the LifeKeeper service, stopping it will also stop the LifeKeeper service. Likewise, it must be started before the LifeKeeper service can be started.

Command Line

When stopping LifeKeeper, there are a number of related services that must be stopped. This process can take several seconds, although the Services tool does not reflect exactly when all the services are stopped. Using the command line to enter `$LKROOT\bin\lkstop` will show the services as they are being stopped, and when completed, the message “LIFEKEEPER NOW STOPPED” will display as confirmation.

Caution: Stopping LifeKeeper takes all protected hierarchies out of service. This means that any protected applications will not be accessible.

Chapter 3: Configuration

Configuration Steps

If the LifeKeeper environment has been installed, the LifeKeeper software can be configured on each server in the cluster. Follow the steps in the topic below which contain links to topics with additional details.

LifeKeeper Configuration Steps

Follow the steps below which contain links to topics with additional details. Perform these tasks on *each server* in the cluster.

1. Ensure that the LifeKeeper services are running by checking the **Services** in the **Administrative Tools** on the **Control Panel**. You should see both *LifeKeeper* and *LifeKeeper External Interfaces* services. If they are not both running, start them now.

   For additional information, see Starting and Stopping LifeKeeper.

2. Users with administrator privileges on a LifeKeeper server can run the application client from that server. Click **Start**, then point to **All Programs**, then **SteelEye->LifeKeeper->LifeKeeper (Admin Only)**.

   After the application is loaded, the **LifeKeeper GUI** appears and the **Cluster Connect** dialog is displayed. Enter the **Server Name** you wish to connect to, followed by the **login** and **password**.

3. Create Communication Paths. Before you can activate LifeKeeper protection, you must create the communications path (heartbeat) definitions within LifeKeeper.

4. Set your Server Shutdown Strategy. This tells LifeKeeper whether to switch over resources when you initiate an orderly shutdown.

5. LifeKeeper is now ready to protect your applications. The next step depends on which LifeKeeper Recovery Kit(s) you will be using:

- If you are using a Core Recovery Kit, then refer to the topics for creating Volume, DNS, IP , File Share, LAN Manager, or Generic Application hierarchies. See the LifeKeeper Microsoft IIS Recovery KitLifeKeeper Microsoft IIS Recovery Kit documentation for creating and extending IIS resources.
- If you are using an optional Recovery Kit, refer to the Recovery Kit section for instructions on creating and extending your resource hierarchies.

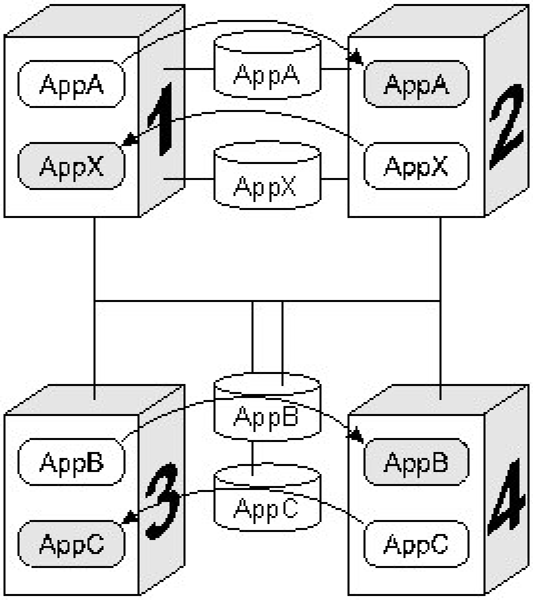
Active-Active Grouping

In an active/active group, all servers are active processors; they also serve as the backup server for resource hierarchies on other servers.

For example, the configuration example below shows two active/active pairs of servers. *Server 1* is processing *AppA*, but also serves as the backup server for *AppX* running on *Server 2*. The reverse is also true. *Server 2* is processing *AppX*, but also serves as the backup server for *AppA* running on *Server 1*. *Servers 3* and *4* have the same type of active/active relationships.

Although the configurations on *Servers 1* and *2* and the configurations on *Servers 3* and *4* are similar, there is a critical difference. For the *AppA* and *AppX* applications, *Servers 1* and *2* are the only servers available for grouping. They are the only servers that have access to the shared resources.

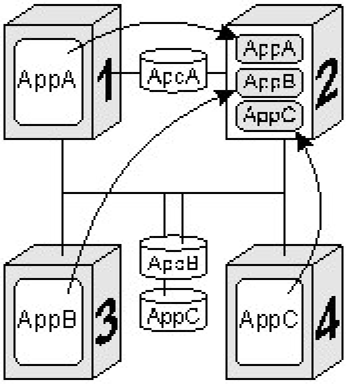
AppB and *AppC*, however, have several grouping options because all four servers have access to the *AppB* and *AppC* shared resources. *AppB* and *AppC* could also be configured to failover to *Server1* and/or *Server2* as a third or even fourth backup system.



Note: Because LifeKeeper applies locks at the volume level, only one of the four systems connected to the *AppB* and *AppC* disk resources can have access to them at any time. Therefore, when Server 3 is actively processing *AppB*, those disk resources are no longer available to *Servers 1*, *2*, and *4*, even though they have physical connections.

Active-Standby Grouping

In an active/standby group, the primary server is processing, and the back-up servers are standing by in case of a failure on the primary server. The standby systems can be smaller, lower-performance systems, but they must have the processing capability to assure resource availability should the primary server fail.



A standby server can provide backup for more than one active server. For example in the figure above, *Server2* is the standby server in three active/standby resource pairs. The LifeKeeper resource definitions specify the following active/standby paired relationships:

- *AppA* on *Server1* fails over to *Server2*.
- *AppB* on *Server3* fails over to *Server2*.
- *AppC* on *Server4* fails over to *Server2*.

Be aware of these three critical configuration concepts when you are considering configurations with multiple active/standby groups:

- **Disk ownership**. Different active applications cannot use disk slices on the same volume. LifeKeeper applies locks at the volume level. When the SCSI locks are applied, only one system on the shared SCSI bus can access volumes on the disk device. In the example, *Server3* has ownership of the *AppB* disk resources and *Server4* owns the *AppC* resources.
- **Processing capacity**. Although it is unlikely that *Servers 1*, *3*, and *4* would fail at the same time, you must take care when designating a standby server to support multiple resource relationships so that the standby server can handle all critical processing should multiple faults occur.
- **LifeKeeper administration**. In the example, *Server2* provides backup for three other servers. In general it is not desirable to administer the LifeKeeper database on the different logical groups simultaneously. You should first create the resources between the spare and one active system, then between the spare and another active system, and so on.

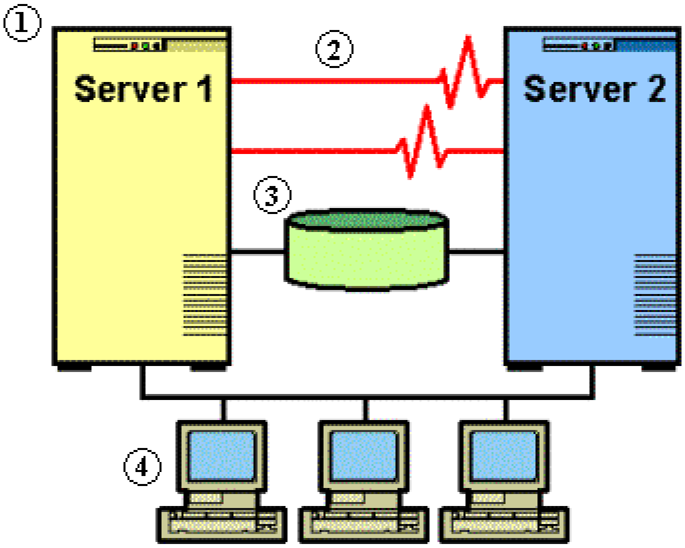
Common Hardware Components

All LifeKeeper configurations share these common components as illustrated in the diagram below:

1. **Server Groups**. The basis for the fault resilience provided by LifeKeeper is clustered Windows 2003 servers. The servers, also referred to as LifeKeeper nodes, do not have to be the same hardware platform.
2. **Communication paths for heartbeat**. It is strongly recommended that each pair of servers in

the group share at least two communication paths (comm paths), although only one is required. To avoid unnecessary failover due to communication failure, you should configure your redundant comm paths using different protocols and communication media, for example TCP/IP (or socket). LifeKeeper uses the comm paths to coordinate resource availability for the fault-detection heartbeat, a periodic message between nodes and for switchover of resources. (See Overview of Communication Paths.)

3. **Shared data resources**. LifeKeeper can recover and restore shared or mirrored data, applications, and communication resources. LifeKeeper controls access at the volume (drive letter) level. In case of a server failure, LifeKeeper automatically switches availability of protected resources to an active server. Peripheral devices that are to be shared between systems must be packaged in external peripheral cabinets. See the Configuring Your Storage topic for information to help you configure your shared storage.
4. **Shared communication for user connections**. LifeKeeper can also automatically manage the switchover of user communication resources, such as IP addresses, computer alias names, and file share lists. Switchover of communication resources allows users to connect using their normal paths.



Intelligent Versus Automatic Switchback

By default, the switchback setting of a resource is *intelligent*. This means that once the failover occurs for that resource from *Server A* to *Server B*, the resource remains on *Server B* until another failure or until an administrator *intelligently* switches the resource to another server. Thus the resource continues to run on *Server B* even after *Server A* returns to service. *Server A* now serves as a backup for the resource.

In some situations, it may be desirable for a resource to switch back automatically to the original failed server when that server recovers. LifeKeeper offers an *automatic switchback* option as an alternative to the normal *intelligent switchback* behavior described above. This option can be selected for individual resource hierarchies on individual servers. If *automatic switchback* is selected for a resource hierarchy in the In-Service-Protected (ISP) state running on a given server and that server fails, the resource hierarchy is failed over to a backup system; when the failed server recovers, the hierarchy is automatically switched back to the original server.

Notes:

- If using data replication (DataKeeper), you must choose ***intelligent switchback***. *Automatic switchback* is not supported.
- Checks for *switchback* are only made either when LifeKeeper starts or when a new server is added to the cluster; they are not performed during normal cluster operation.
- LifeKeeper never performs an *automatic switchbac*k from a higher priority server to a lower priority server.
- If there is a dependency between two resources with different *switchback* strategies, the *switchback* strategy of the parent resource takes precedence.

LifeKeeper Configurations

LifeKeeper works on the basis of resource hierarchies you define for groups of two or more servers. The following three topics introduce the LifeKeeper failover configuration concepts.

- Common Hardware Components
- System Grouping Arrangements
- Resource Hierarchies

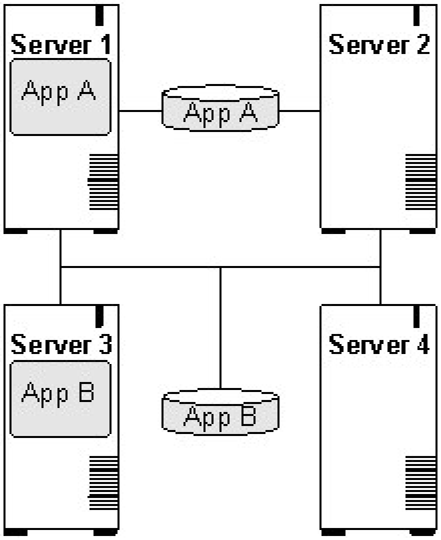
System Grouping Arrangements

A resource hierarchy is defined on a cluster of LifeKeeper servers. For a given hierarchy, each server is assigned a priority, with one *(1)* being the highest possible priority. The primary, or highest priority, server is the computer you want to use for the normal operation of those resources. The server having the second highest priority is the backup server to which you want LifeKeeper to switch those resources should the primary server fail.

In an active/active group, all servers are active processors, but they also serve as the backup server for resource hierarchies on other servers. In an active/standby group, the primary server is processing and any one of the backup servers can be configured to stand by in case of a failure on the primary server. The standby systems can be smaller, lower-performance systems, but they must have the processing capability to assure resource availability should the primary server fail.

Your physical connections and access to the shared resources determine your grouping options. To be grouped, servers must have communications and heartbeat paths installed and operational, and all servers must have access to the disk resources through a shared SCSI or Fibre Channel interface. For example in the following diagram, there is only one grouping option for the resource *AppA* on *Server 1*. *Server 2* is the only other server in the configuration that has shared access to the *AppA* database.

The resource *AppB* on *Server 3*, however, could be configured for a group including any one of the other three servers, because the shared SCSI bus in this example provides all four servers in the configuration access to the *AppB* database.



Chapter 4: Administration

LifeKeeper Administration

LifeKeeper does not require administration during operation. LifeKeeper works independently to monitor protected resources and to perform the specified recovery actions if a fault should occur.

LifeKeeper Administration Overview

LifeKeeper provides two administration interface options:

- LifeKeeper GUI
- LifeKeeper command line interface

The LifeKeeper GUI is used for the following tasks which are listed in the typical sequence for configuring LifeKeeper.

- **Communications path definition**. You must define the communications paths you want to use before you define any other resource instances or hierarchies in LifeKeeper. This can be done using the Edit menu or the Create Comm Path icon on the GUI toolbar.
- **Resource definition**. As you install recovery kits, the resource types supported by those kits appear in the Create Resource Hierarchy dialog box.For most recovery kits, the necessary dependencies will be created automatically.
- **Monitoring**. The LifeKeeper GUI's status display provides a visual status of resources protected by LifeKeeper on the connected servers. In addition, LifeKeeper maintains log files which you can view through the GUI.
- **Manual intervention**. You may need to stop servers or specific resources for maintenance or other administrative actions. The LifeKeeper GUI provides menu functions that allow you to bring specific resources in and out of service. Once applications have been placed under LifeKeeper protection, they should be started and stopped only through LifeKeeper.

For initial step-by-step configuration instructions, see LifeKeeper Configuration Steps.

See the GUI Tasks and Maintenance Tasks topics for detailed instructions on performing LifeKeeper administration, configuration and maintenance operations using the GUI.

Note: LifeKeeper is set up so that the LifeKeeper services are run by the local system account on each server. LifeKeeper should not be changed to run as any other user account.

Administrator GUI Tasks

Editing Server Properties

1. To edit the properties of a server, begin just as you would for viewing server properties.
2. If you are logged into that server with the appropriate permissions, the following items will be editable.
    - Shutdown Strategy
    - Automatic Failover Configuration
    - Server Configuration (only for servers with specialized configuration settings)
3. Once you have made changes, the **Apply** button will be enabled. Clicking this button will apply your changes without closing the window.
4. When you are finished, click **OK** to save any changes and close the window, or **Cancel** to close the window without applying changes.

Set Server Shutdown Strategy

The Shutdown Strategy is a configuration option that governs whether or not resources are switched over to a backup server when a server is shut down. The options are:

Do Not Switchover Resources (default)	LifeKeeper will not switchover resource hierarchies during an orderly shutdown.
Switchover Resources	LifeKeeper will switchover all resource hierarchies during an orderly shutdown.

Restriction: The Switchover on Shutdown setting is not supported with SteelEye DataKeeper resources.

The Shutdown Strategy is set by default to ”***Do Not Switchover Resources***.” You should decide which strategy you want to use on each server, and if you wish, change the Shutdown Strategy to "***Switchover Resources***".

For each server in the cluster:

1. Bring up the **Server Properties** dialog just as you would for viewing server properties.
2. On the General Tab of the Server Properties dialog, select the **Shutdown Strategy**.

Note: The LifeKeeper process must be running during an orderly shutdown for the Shutdown Strategy to have an effect. If LifeKeeper is not running or the resources are not currently in service, the resources will not switch over.

Server Properties

The Server Properties dialog is available from the Edit Menu or from a server popup menu. This dialog displays the properties for a particular server. When accessed from the **Edit** menu, you can select the server. The **Server Properties** dialog updates itself when the selected server changes.

The **OK** button applies any changes that have been made and then closes the window. The **Apply** button applies any changes that have been made. The **Cancel** button closes the window without saving any changes made since **Apply** was last clicked.

General Tab

- **Name.** Name of the selected server.
- **Permission.**The permission level of the user currently logged into that server. These are the possible permission values:
    - **Administrator** - the user can perform any LifeKeeper task.
    - **Operator**- the user can monitor LifeKeeper resource and server status, and can bring resources in service and take them out of service.
    - Guest- the user can monitor LifeKeeper resource and server status.
- **State.** Current state of the server. These are the possible server state values:
    - **ALIVE**- server is available.
    - **DEAD**- server is unavailable.
    - **UNKNOWN** - state could not be determined. The GUI server may not be available.
- **ShutdownStrategy.** (editable) The setting that governs whether or not resources which are in service are switched over to a backup server in the cluster when a server is shutdown. The setting "Switchover Resources" indicates that resources will be brought in service on a backup server in the cluster. The setting "Do not Switchover Resources" indicates that resources will not be brought in service on another server in the cluster.
- **Server Name.** Automatic failover capabilities from the local server to other servers in the cluster may be configured here. All servers in the cluster should be operational (i.e. at least one LifeKeeper comm path must be active) as inactive servers are not listed. The name of each active server in the cluster is listed, excluding the local server. For each server, two types of failover capability are configurable. By default, all failover capabilities are enabled.
    - **Disable Resource Failover**- Select the remote server(s) to be disqualified as a backup server for any failed resource hierarchy on the local server. When disabled, the designated server is disqualified as a failover site if a local resource fails. Unselect to re-enable automatic failover capabilities.
    - **Disable System Failover**- Select the remote server(s) to be disqualified as a backup server for a complete failure of the local server. When disabled, the designated server

is disqualified as a failover site if the local server completely fails. Unselect to re-enable automatic failover capabilities.

Note: If all remote servers are disabled for resource failovers, then the failed resource will be marked as "Failed" and no additional quick check or deep check monitoring will be performed for the failed resource. However, the failed resource as well as other dependent resources in the hierarchy will not be removed from service and no failover will be attempted.

CommPaths Tab

- **Server.** The server name of the other server to which the communication path is connected in the LifeKeeper cluster.
- **Type.** The type of comm path between the server in the list and the server specified in the **Server** field (TCP/IP or Shared Disk).
- **State.** State of the comm path in the LifeKeeper Configuration Database (LCD). These are the possible comm path state values:
  - **ALIVE** - functioning normally
  - **DEAD** - no longer functioning normally
  - **UNKNOWN** - state could not be determined. The GUI server may not be available.
- **Address/Device.** The IP address or device name that this comm path uses.
- **Comm Path Status.** Summary comm path status determined by the GUI based on the state of the comm paths in the LifeKeeper Configuration Database (LCD). These are the possible comm path status values displayed below the detailed text in the lower panel:
  - **NORMAL** - all comm paths functioning normally
  - **FAILED**- all comm paths to a given server are dead
  - **UNKNOWN**- comm path status could not be determined. The GUI server may not be available.
  - **WARNING**- one or more comm paths to a given server are dead, or only one comm path exists.
  - **DEGRADED**- one or more redundant comm paths to a given server are dead
  - **NONE DEFINED** - no comm paths defined

Resources Tab

- **Name.** The tag name of a resource instance on the selected server.
- **Application.** The application name of a resource type (gen, scsi, ...)
- **Resource Type.** The resource type, a class of hardware, software, or system entities providing a service (for example, volume, TCP/IP, SQL...)

- **State.** The current state of a resource instance.
    - **ISP** - In-service locally and protected.
    - **ISU** - In-service locally, but local recovery will not be attempted.
    - **OSF** - Out-of-service, failed.
    - **OSU** - Out-of-service, unimpaired.
    - **ILLSTATE** - Resource state has not been initialized properly by the resource initialization process which is run as part of the LifeKeeper startup sequence. Resources in this state are not under LifeKeeper protection.
    - **UNKNOWN** - Resource state could not be determined. The GUI server may not be available.

Disabling Automatic Failover

In the event that the primary server has attempted and failed local recovery, or failed completely, most server administrators will want LifeKeeper to automatically restore the protected resource(s) to a backup server. This is the default LifeKeeper behavior. However, some administrators may not want the protected resource(s) to automatically go in service at a recovery site. For example, if LifeKeeper is installed in a WAN environment where the network connection between the servers may not be reliable in a disaster recovery situation.

Automatic failover is enabled by default for all protected resources. To disable automatic failover for protected resources or to prevent automatic failover to a backup server, use the **Failover** section located on the **General** tab of Server Properties to configure as follows:

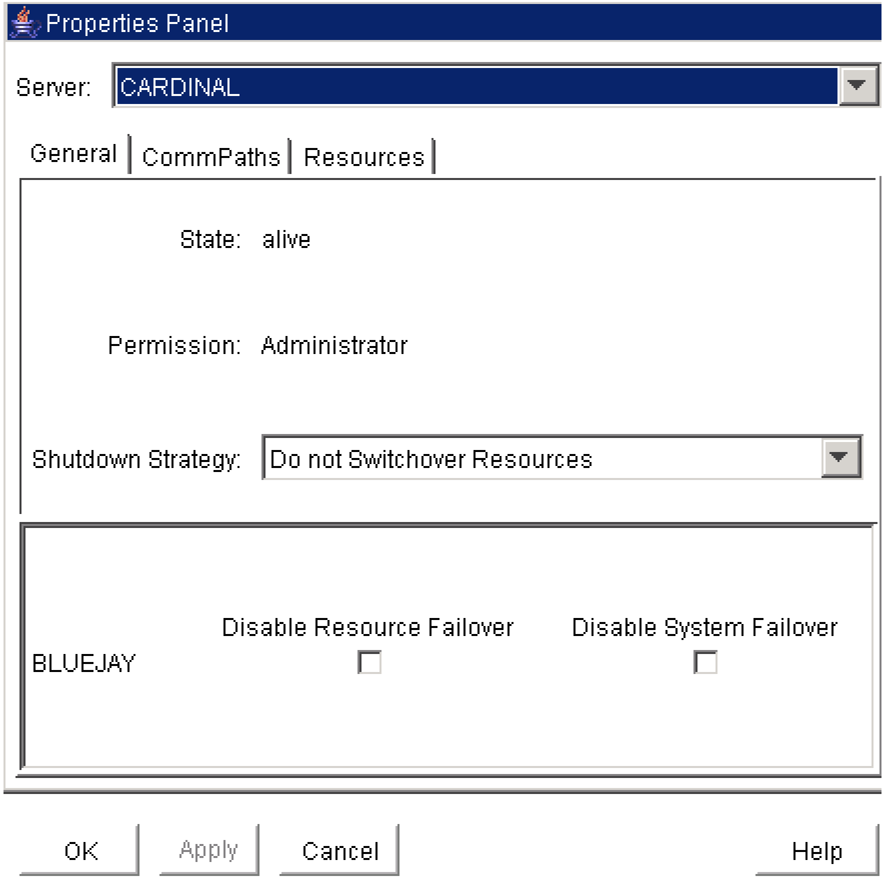
For each server in the cluster:

1. Bring up the **Server Properties** dialog just as you would for viewing server properties.
2. Select the General tab. In the **Failover** section of the **Server Properties** dialog, check the server to disable system and resource failover capabilities. By default, all failover capabilities of LifeKeeper are enabled.

In the **Disable System Failover** column, select the server to be disqualified as a backup server for a complete failure of the local server.

In the **Disable Resource Failover** column, select the server to be disqualified as a backup server for any failed resource hierarchy on this local server. Resource failovers cannot be disabled without first disabling system failover capabilities.

To commit your selections, press the **Apply** button.



Server: CARDINAL

State: alive

Permission: Administrator

Shutdown Strategy: Do not Switchover Resources

	Disable Resource Failover	Disable System Failover
BLUEJAY	□	□

Creating a Communication Path

Before configuring a LifeKeeper communication path between servers, verify the hardware and software setup. See the Configuration section for requirements.

Configuration Notes

- You should configure **no more than one shared disk comm path and one TTY comm path** between servers.
- TTY and Shared Disk comm paths are supported for two-server clusters only.
- For greater than two-server clusters, use multiple TCP/IP comm paths for heartbeat redundancy. A priority value is used to tell LifeKeeper the order in which TCP/IP paths to a given remote server should be used.
- **IMPORTANT**: Supported configurations require that you define redundant comm paths so that the failure of a single communication line will not cause an unnecessary failover. If a single comm path is used and the comm path fails, LifeKeeper hierarchies may come in-service on multiple servers simultaneously. This is known as "split-brain". Additionally, heavy

network traffic on a TCP/IP comm path can result in unexpected behavior, including false failovers and LifeKeeper initialization problems.

Creating a Comm Path

1. Select one of the servers, and then select **Create Comm Path** from the server context menu or server context toolbar.
2. Select one or more **Remote Servers** in the list box. If a remote server is not listed in the list box (i.e. it is not yet connected to the cluster), you may enter it using **Add Server**. You must make sure that the network addresses for both the local and remote servers are resolvable (for example, with DNS or added to the */etc/hosts* file). Click **Next**.
3. Select either *TCP*, *TTY* or *DISK* for **Device Type** and click **Next**.
4. Provide all required information for the **Device Type** that you selected, and click **Next** after each step. Refer to the table below for additional information on each configuration field.

Field	Tips
For TCP/IP Comm Path...	
Heartbeat Interval	Enter a value between 4 and 15 for the heartbeat interval, which is the number of seconds between heartbeat signals that verifies the servers are alive). The default = 6.
Maximum Heartbeat Misses	Enter a value between 3 and 99. This is the number of consecutive heartbeat signals that can be missed before the comm path is marked as dead. The default = 5.
Local IP Address	Enter the IP address to be used by the local server for this comm path.
Priority	Enter the priority for the comm path on the local server. The priority will be used to determine the order that the comm paths between the two servers will be used. Priority 1 is the highest, 99 is the lowest.
Remote IP Address	Enter the IP address to be used by the remote server for this comm path.
Port Number	Enter a unique port number to be used by the TCP/IP service. This number must be between 1500 and 10000. LifeKeeper offers a default which you can change.
For TTY Comm Path...	
Heartbeat Interval	Enter a value between 4 and 15 for the heartbeat interval, which is the number of seconds between heartbeat signals that verifies the servers are alive). The default = 6.
Maximum Heartbeat Misses	Enter a value between 3 and 99. This is the number of consecutive heartbeat signals that can be missed before the comm path is marked as dead. The default = 5.

Field	Tips
Priority	Enter the priority for the comm path on the local server. The priority will be used to determine the order that the comm paths between the two servers will be used. Priority 1 is the highest, 99 is the lowest.
Local TTY Com Port	Select a TTY port number on the local server to be used to communicate with the remote server.
Remote TTY Com Port	Select a TTY port number on the remote server to be used to communication with the local server.
Baud Rate	Select the baud rate for the TTY COM ports on the local and remote server.
For Shared Disk Comm Path...	
Heartbeat Interval	Enter a value between 4 and 15 for the heartbeat interval, which is the number of seconds between heartbeat signals that verifies the servers are alive). The default = 6.
Maximum Heartbeat Misses	Enter a value between 3 and 99. This is the number of consecutive heartbeat signals that can be missed before the comm path is marked as dead. The default= 5.
Priority	Enter the priority for the comm path on the local server. The priority will be used to determine the order that the comm paths between the two servers will be used. Priority 1 is the highest, 99 is the lowest.
Drive Letter	The drive letter associated with the shared volume to be used for the shared disk comm path. This must be the same letter on both servers.

5. Click **Create**. The dialog should display a message indicating the network connection is successfully created. If the output panel is enabled, the message will be displayed there as well. Click **Next**.

6. If you selected multiple Local IP Addresses or multiple Remote Servers and the **Device Type** was set to **TCP**, then you will be taken back to Step 4 to continue with the next Comm Path. If you selected multiple Remote Servers and the **Device Type** was set to **TTY** or **DISK**, then you will be taken back to Step 3 to continue with the next Comm Path.

7. Click **Done** when presented with the concluding message.

Verifying the Comm Path

You can verify the comm path by viewing the Server Properties dialog. You should see an **Alive** status.

In addition, check the server icon in the right pane of the GUI. If this is the first comm path that has been created, the server icon shows a yellow heartbeat indicating that one comm path is ALIVE, but there is no redundant comm path.	⚠
The server icon will display a green heartbeat when there are at least two comm paths ALIVE.	✓

If the comm path does not activate after a few minutes, verify that the paired server's computer name is correct. If using TTY comm paths, verify that the cable connection between the two servers is correct and secure.

Deleting a Communication Path

1. Select one of the servers, and then select **Delete Comm Path** from the server context menu or server context toolbar.

2. Select the communications path(s) that you want to delete and click **Delete Comm Path(s)**.

3. If the output panel is enabled, the dialog closes, and the results of the commands to delete the communications path(s) are shown in the output panel. If not, the dialog remains up to show these results, and you click **Done** to finish when all results have been displayed.

Working With Resource Hierarchies

The following topics describe the tasks that are common across any type of resource hierarchy. These tasks function very much the same regardless of whether you are working with a core Recovery Kit or an optional Recovery Kit:

Creating Resource Hierarchies

Extending Resource Hierarchies

Unextending a Hierarchy

Bringing a Resource In Service

Taking a Resource Out of Service

Adding a Resource Dependency

Removing a Resource Dependency

Deleting a Hierarchy from All Servers

The optional LifeKeeper Recovery Kit documentation is available on the LifeKeeper for Windows Technical Documentation under Recovery Kits.

Creating Resource Hierarchies

1. Select the server, and then select **Create Resource Hierarchy** from the server context menu or server context toolbar.

2. A dialog entitled **Create Protected Application** will appear with a list of all recognized recovery kits installed within the cluster. Select the **Recovery Kit** that builds resource hierarchies to protect your application and click **Next**.
3. Continue through the succeeding dialogs, entering whatever data is needed for the type of resource hierarchy that you are creating.
4. After receiving the message that the resource hierarchy has been created successfully, click **Next** to continue. If LifeKeeper has detected a problem, an ERROR will appear in the information box, the partially created resource hierarchy will be removed and the **Next** button will be disabled. In that case, click **Cancel** to exit the Wizard.

LifeKeeper Application Resource Hierarchies

If you install LifeKeeper without any optional recovery kits, the **Application to Protect** list includes options **DNS**, **File Share List**, **Generic Application**, **IIS**, **IP Address**, **LAN Manager**, and **Volume** by default. The **Generic Application** option may be used for applications that have no associated recovery kits.

See the following topics describing these available options:

- Creating a DNS Resource Hierarchy
- Creating a File Share Resource Hierarchy
- Creating a Generic Application Resource Hierarchy
- Creating an IP Address Resource Hierarchy
- Creating a LAN Manager Resource Hierarchy
- Creating a Volume Resource Hierarchy

Additional Recovery Kits

Each optional recovery kit that you install adds entries to the **Application to Protect** list; for example, you may see **Oracle**, **MS SQL Server** or **MS Exchange Server** (2003 or 2007) Recovery Kits. Refer to LifeKeeper Recovery Kits for instructions on creating the required resource hierarchies.

Creating a DNS Resource Hierarchy

The DNS Recovery Kit provides a mechanism to update the DNS A record and associated PTR record (if exists) of the primary server or an alias name on all DNS servers in your configuration. The kit allows selection of the name for the primary server or an alias name in DNS which will be modified with the IP address of a backup server when failover or switchover occurs.

The example below shows the changes that occur on the DNS Server for a protected DNS resource after failover or switchover.

```
Primary Server: ExchSrvr1 (172.17.10.24/255.255.255.0)

Backup Server: ExchSrvr2 (172.16.10.25/255.255.255.0)
```

DNS Server:

```
Zone: mydomain.com
```

Before failover:

A Record	ExchSrvr1	172.17.10.24
	ExchSrvr2	172.16.10.25
PTR Record	24.10.17.172.in-addr.arpa	ExchSrvr1.mydomain.com
	25.10.16.172.in-addr.arpa	ExchSrvr2.mydomain.com

After Failover:

A Record	ExchSrvr1	172.16.10.25
	ExchSrvr2	172.16.10.25
PTR Record	25.10.16.172.in-addr.arpa	ExchSrvr1.mydomain.com
	25.10.16.172.in-addr.arpa	ExchSrvr2.mydomain.com

During create of a DNS resource, enter the primary server name or an alias name which will be modified on the primary DNS server upon failover or switchover. If the server is multi-homed, select the IP address of the *A record* to be updated. The *A* and *PTR records*, based on the selection made during the create, will be created if they do not exist on the DNS server. The records are created on the primary DNS server specified during create and on all of the NS (Name server) servers who are the primary zone servers for the *A record*.

The deep check script, which monitors the DNS resource, will check for the existence of the *A record* of the protected server (either primary server name or alias name) on all of the DNS servers. If the *A record* mapping to the correct IP address is not found on at least one of the DNS servers, the deep check script will fail which will trigger local recovery (if enabled) and the *A* and *PTR records* will be recreated on all of the primary DNS servers.

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

1. Select the server, and then select **Create Resource Hierarchy** from the server context menu or server context toolbar.
2. The **Create Protected Application** window appears displaying the **Primary** and **Backup** Servers in your cluster. Select the correct systems for this configuration.
3. A dialog appears with a list of all recognized recovery kits installed within the cluster. Select **DNS** and click **Next**.
4. The **Create Protected Application** will prompt you to enter 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 especially helpful should you encounter an error that might require you to correct previously entered information. If you click **Cancel** at any time during the sequence of creating your hierarchy, LifeKeeper will cancel the entire creation process.

Field	Tips
Resource DNS Tag	Select or enter a unique tag for the DNS resource instance you are creating. This field provides a default tag name which you can change if desired.
A Record Owner Name	Enter the name of the server whose A and PTR records will be updated in DNS. If you want to update DNS records that belong to an alias of the server, enter the alias name here. LifeKeeper will protect DNS records for the alias name in DNS mapping to the IP address you will select later.
IP Address	Enter the IP address of the server or alias name whose A record will be updated. The A record mapping to this IP address will be updated upon failover switchover.
DNS Server Name (Fully Qualified)	Enter the fully qualified name of a DNS server, in the form of *<DNS Server Name>* *<mydomain>.com*, where the Resource Records can be modified. The DNS server should be accessible from the primary server, preferably in the same site. Upon failover or switchover, records on the NS (Name Server) in the environment will also be updated.
DNS Administrative User Name	Enter the user name of the Windows DNS/Domain administrator. This user account should have privileges to make changes in the DNS configuration and should be a member of the "Domain Admins" group in the same domain as the DNS server. Enter the user ID in *<DomainName>\<UserID>* format where *<DomainName>* is the NetBIOS name of the domain.
DNS Administrator Password	Enter the password associated with the Windows DNS/Domain administrator account.

5. After receiving the message that the resource hierarchy has been created successfully, click **Next** to continue. If LifeKeeper has detected a problem, an ERROR will appear in the information box, the partially created resource hierarchy will be removed, and the **Next** button will be disabled. In that case, click **Cancel** to exit the **Wizard**.

Creating a File Share Resource Hierarchy

The Windows File Manager function allows you to define file shares. If you install the LifeKeeper LAN Manager recovery kit, you can create a file share list resource that includes one or more of those file shares.

Criteria for File Share Resources

Not all file shares are available to be shared. The following statements will help you to determine which files shares are available.

- The share name must reside on a volume that is shared between the machines.
- The shared volume can already be protected between the two machines where the file share resource is being created; however, it should not exist on a third machine until you extend the

file share hierarchy to that machine.

- If the share name already exists on the second machine then both share names must point to the exact same directory.
- If the share name is already protected on either machine, it is not eligible.
- It is the responsibility of the administrator to ensure that any share names created actually point to directories. It is possible to create a share name for a directory and then delete the directory. If this is the case then the administrator should ensure that the share name is deleted as well.

Note: After a file share has been brought in-service on a backup server it becomes a share on that machine. The share remains even after the hierarchy is deleted.

File Share Resource Creation

To create a file share resource hierarchy, follow the steps below.

1. Select the server, and then select **Create Resource Hierarchy** from the server context menu or server context toolbar.
2. The **Create Protected Application** window appears displaying the **Primary** and **Backup** Servers in your cluster. If not already selected, choose the appropriate systems to configure.
3. A dialog appears with a list of all recognized recovery kits installed within the cluster. Select **File Share List** and click **Next**.
4. The **Configuration Wizard** will prompt you to enter 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 especially helpful should you encounter an error that might require you to correct previously entered information. If you click **Cancel** at any time during the sequence of creating your hierarchy, LifeKeeper will cancel the entire creation process.

Field	Tips
File Share and Path Name	Select one or more file shares to be protected. If "none found" is displayed, verify that the volume where the file share exists is under LifeKeeper protection.
File Share Resource Tag	Select or enter a unique tag for the File Share resource instance you are creating. This field provides a default tag name FSList.x (where x is a number assigned by LifeKeeper, starting with) which you can change if desired.

5. After all of the data is entered, the **Next** button will appear. When you click **Next,** LifeKeeper will create and validate your resource hierarchy.
6. After receiving the message that the resource hierarchy has been created successfully, click **Next** to continue. If LifeKeeper has detected a problem, an ERROR will appear in the

information box, the partially created resource hierarchy will be removed, and the **Next** button will be disabled. In that case, click **Cancel** to exit the **Wizard**.

Creating a Generic Application Resource Hierarchy

Use this option to protect an application that has no associated recovery kit.

Before Creating a Resource Hierarchy

1. The first task is to create scripts for the five basic LifeKeeper action functions:

    - Restore
    - Remove
    - Quick Check
    - Deep Check
    - Local Recovery

    Perl and VB Script templates are provided for these scripts in *$LKROOT\admin\kit\app\templates*. Be sure to copy these templates to another directory on the same volume as *$LKROOT* before customizing and testing them for the application that you wish to protect.

    Note: If you want to use optional **Create**, **Extend** and **Delete** scripts, also include them in the folder with your other scripts. The script selection wizard will search for them by these names (and extension) and automatically enter them for you.

2. For applications depending upon other resources such as a volume or IP address, create each of these resources separately before creating your Generic Application resource hierarchy. You can create the appropriate dependencies later using Add Dependency.

Creating Your Resource Hierarchy

Now you are ready to create the Generic Application resource hierarchy using the modified scripts.

1. Select the server, and then select **Create Resource Hierarchy** from the server context menu or server context toolbar.
2. The **Create Protect Application** window appears displaying the **Primary** and **Backup** Servers in your cluster. If not already selected, choose the appropriate systems to configure. Click **Next**.
3. A dialog appears with a list of all recognized recovery kits installed within the cluster. Select **Generic Application** and click **Next**.
4. The **Configuration Wizard** will prompt you to enter the following information. **Note**: When the **Back** button is active in any of the dialog boxes, you can go back to the previous dialog box. This is especially helpful should you encounter an error that might require you to correct

previously entered information. If you click **Cancel** at any time during the sequence of creating your hierarchy, LifeKeeper will cancel the entire creation process.

Field	Tips
Restore Script	Enter the path and filename for the **Restore Script** for the application. This is the command that starts the application. A template restore script is provided in the templates directory. The restore script must not impact applications that are already started.
Remove Script	Enter the path and filename for the **Remove Script** for the application. This is the command that stops the application. A template remove script is provided in the templates directory.
Quick Check Script [optional]	Enter the path to the **Quick Check Script** for the application. This is the command that monitors the application. A template quickchk script is provided in the templates directory.
Deep Check Script [optional]	Enter the path to the **Deep Check Script** for the Application. This command monitors the protected application in more detail than the Quick Check Script. A template deepchk script is provided in the templates directory.
Local Recovery Script [optional]	Enter the path to the **Local Recovery Script** for the application. This is the command that attempts to restore a failed application on the local server. A template recover script is provided in the templates directory.
Application Information [optional]	Enter any **Application Information** next. This is optional information about the application that may be needed by the restore, remove, recover, and quickCheck scripts.
Resource Tag Name	This field provides a default tag name *App.x* (where x is a number assigned by LifeKeeper, starting with *0*) which you can change if desired.

5. After all of the data is entered, the **Create Instance** button will appear. When you click **Create Instance**, LifeKeeper will create and validate your resource hierarchy.
6. After receiving the message that the resource hierarchy has been created successfully, click **Next** to continue. If LifeKeeper has detected a problem, an ERROR will appear in the information box, the partially created resource hierarchy will be removed, and the **Next** button will be disabled. In that case, click **Cancel** to exit the **Wizard**.

Creating a LAN Manager Resource Hierarchy

The LAN Manager Recovery Kit provides a way to create a computer alias name with associated file shares. The computer alias name acts as a"switchable" computer name, and its associated file shares become available on the system that has the LifeKeeper LAN Manager hierarchy in service. In addition, an IP address can be associated with the computer alias name as part of the hierarchy.

1. Select the server, and then select **Create Resource Hierarchy** from the Server Context Menu or Server Context Toolbar.

2. The **Create Protect Application** window appears displaying the **Primary** and **Backup** Servers in your cluster. If not already selected, choose the appropriate systems to configure. Click **Next**.
3. A dialog appears with a list of all recognized recovery kits installed within the cluster. Select **LAN Manager** and click **Next**.
4. The **Configuration Wizard** will prompt you to enter 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 especially helpful should you encounter an error that might require you to correct previously entered information. If you click **Cancel** at any time during the sequence of creating your hierarchy, LifeKeeper will cancel the entire creation process.

Field	Tips
Computer Alias Name	Enter a name to be used for the computer alias, or you can accept the default name offered by LifeKeeper.
LAN Manager Resource Tag	Select or enter a unique tag for the LAN Manager resource instance you are creating. This field provides a default tag name (the same as the computer alias name entered in the previous step) which you can change if desired.

5. After all of the data is entered, the **Next** button will appear. When you click **Next**, LifeKeeper will create and validate your resource hierarchy.
6. After receiving the message that the resource hierarchy has been created successfully, click **Next** to continue. If LifeKeeper has detected a problem, an ERROR will appear in the information box, the partially created resource hierarchy will be removed, and the **Next** button will be disabled. In that case, click **Cancel** to exit the **Wizard**.

Creating a Volume Resource Hierarchy

When you want to protect resources on shared SCSI disks, you partition the shared disk into logical volumes using the Windows Disk Management tool. LifeKeeper can protect shared volumes by defining a volume resource instance. Each instance is assigned a drive letter (for example, G:).

LifeKeeper brings the volume resource instance into service on the primary server and provides software locks so that a backup server cannot access the volume while it is active on the primary server. In case of a failure of the primary server, LifeKeeper automatically brings the volume resource into service on the backup server and locks the primary server from accessing the volume resource when it is repaired.

LifeKeeper also automatically changes the primary and designations so that the failed server is now locked from access to the volume resource. In this way, the resource is protected from inappropriate access while you repair the failed server.

This dynamic redefinition of primary and backup servers is LifeKeeper's intelligent switchback feature that allows you to select the appropriate time to bring the resource back into service on the repaired system.

To create a volume resource, follow the steps below. Since LifeKeeper maintains the volume locks, do not stop LifeKeeper, after creating the resource, as this would disable the locks.

Note: Before creating and extending a mirrored volume resource, be sure to exit from any DataKeeper GUI processes that are connected to any of the LifeKeeper cluster systems.

1. Select the server, and then select **Create Resource Hierarchy** from the server context menu or server context toolbar.
2. The **Create Protected Application** window appears displaying the **Primary** and **Backup** Servers in your cluster. If not already selected, choose the appropriate systems to configure.
3. A dialog appears with a list of all recognized recovery kits installed within the cluster. When the **Back** button is active in any of the dialog boxes, you can go back to the previous dialog box. This is especially helpful should you encounter an error that might require you to correct previously entered information. If you click **Cancel** at any time during the sequence of creating your hierarchy, LifeKeeper will cancel the entire creation process.

Field	Tips
Volume	Select the volume to be protected. If "none found" is displayed, verify that the volume is under LifeKeeper protection.
Volume Tag	The Volume tag is a resource identifier. LifeKeeper provides a default volume tag name in the form: Volume.X, where X is the drive letter. You can change the tag name, but it must be unique.

4. After the data is entered, the **Next** button will appear. When you click **Next**, LifeKeeper will create and validate your resource hierarchy.
5. After receiving the message that the resource hierarchy has been created successfully, click **Next** to continue. The **Extend Volume Resource** window displays. Refer to the help topic, Extending a Volume Resource Hierarchy for additional information while completing this procedure.
6. If LifeKeeper has detected a problem, an ERROR will appear in the information box, the partially created resource hierarchy will be removed and the **Next** button will be disabled. In that case, click **Cancel** to exit the **Wizard**.

After the Volume Resource is Created

After a LifeKeeper volume is created or deleted, the following command is executed for the LifeKeeper protected volume:

```
chkntfs /x <vol_1> <vol_2> ... <vol_n>
```

This Windows command excludes the volumes listed from being checked by chkdsk at system startup. This is required for LifeKeeper protected volumes so that they will not be accessed - particularly on backup systems - before LifeKeeper has a chance to start. If no LifeKeeper volumes remain, `chkntfs /d` is executed to restore the Windows default settings.

Caution: The `chkntfs /x` command does not remember previous volumes it was applied to, so if a user executes this command, it could disable the LifeKeeper settings, (and likewise, LifeKeeper could subsequently override the user's settings). If you wish to exclude a non-LifeKeeper volume from checking at startup, you should also include all the LifeKeeper volumes in the `chkntfs /x` command.

Creating an IP Address Resource Hierarchy

LifeKeeper provides the ability to monitor local switchable IP addresses and moves them to another network adapter in the same system when a failure is detected. This can avoid an entire resource hierarchy failing over to a backup server.

IP Local Recovery imposes requirements and limitations on the system configuration.

Requirements for IP Local Recovery

IP local recovery allows you to specify a single backup network interface for each LifeKeeper-protected IP address on a server. In order for the backup interface to work properly, it must be attached to the same physical network as the primary interface. The system administrator is expected to ensure that a valid interface is being chosen. Note that it is reasonable and valid to specify a backup interface on one server but not on another within the cluster (i.e., the chosen backup interface on one server has no impact on the choice of a backup on any other server).

The backup adapter, also known as the Local Recovery Adapter where the switchable address will become active after a failure of the primary adapter,must be configured in the following way:

- Both adapters must be connected to the same physical subnet.
- For routing purposes, all addresses on the Local Recovery Adapter must be on a different logical subnet than any permanent addresses on the Primary adapter. They must also be on a different logical subnet than any LifeKeeper-protected switchable addresses that are configured on the Primary adapter.
- The IP Local Recovery feature requires that a network gateway exist on the network. Specifically, the default gateway field in the TCP/IP configuration for the system must contain the address of a network gateway. In addition, the local recovery adapter must also be configured with the same network gateway.
- Cabling and network routing must be configured to permit a ping command issued from either logical subnet to reach the protected IP address and its associated subnet when it is placed on either the primary network card or the local recovery network card. This can be verified by manually issuing a ping command from other systems on each logical subnet. A failed ping command indicates a network routing problem.
- IP Local Recovery can only be enabled at the time the IP resource is created. Local Recovery can not be added to an IP resource by modifying its resource attributes after the resource has been created.
- IP Local Recovery may be disabled for an IP resource by using the "ins_setlocalrecovery" command line utility. This utility is located in the LifeKeeper \bin directory (C:\LK\bin by

default). From a command prompt, type "ins_setlocalrecovery" for the usage and switch options.

Before you create and use IP Address resources in LifeKeeper hierarchies, your network should be configured and tested as described in the Verifying Network Configuration topic.

Also verify that the switchable IP address you plan to use is unique using the ping command. The switchable IP address does not need to be created as a prerequisite; it is created when you create the IP address hierarchy.

To create an IP address resource hierarchy from the primary server, you should complete the following steps:

1. Select the server, and then select **Create Resource Hierarchy** from the server context menu or server context toolbar.
2. A dialog entitled **Application to Protect** will appear with a list of all recognized recovery kits installed within the cluster. Select **IP Address** and click **Next**.
3. The **Wizard** will prompt you to enter 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 especially helpful should you encounter an error that might require you to correct previously entered information. If you click **Cancel** at any time during the sequence of creating your hierarchy, LifeKeeper will cancel the entire creation process.

Field	Tips
IP Address	This is the switchable IP address or hostname that LifeKeeper will use for this resource. This is used by client applications to login to the parent application over a specific network interface. If you use a hostname, it must exist in the local *%windir%\system32\etc\drivers\hosts* file or be accessible via a Domain Name Service (DNS). Alias names and domain names are acceptable as long as they meet the criteria listed above. No defaults are provided for this information field. **Note**: If you choose to use a hostname, be advised that when you extend this resource, the actual IP address will appear in one of the dialog boxes as the TCP/IP resource designation.
Subnet Mask	The IP subnet mask which your TCP/IP resource will use on the target server. Any standard netmask for the class of the specific TCP/IP resource address is valid. **Note**: The subnet mask you choose, combined with the IP address, determines the subnet that will be used by the TCP/IP resource and should be consistent with the network configuration.
IP Resource Tag	Select or enter a unique IP Resource Tag name for the IP resource instance you are creating. This field is populated automatically with a default tag name that matches the resource name or IP address. You can change this tag if you want to.
Network Connection	This is the physical Ethernet card that the IP address is interfacing with. Valid choices will depend on the existing network configuration and values chosen for the TCP/IP resource address and netmask. The default value is the interface within the set of valid choices which most closely matches the address and netmask values you have selected in previous dialogs.
Local Recovery Network Connection	If you answered "**Yes**" to Local Recovery, you must select a network connection to use as the backup interface. Specify the backup NIC if one exists; otherwise, specify the primary NIC.

4. After all of the data is entered, click **Next** and LifeKeeper will create and validate your resource hierarchy.

5. After receiving the message that the resource hierarchy has been created successfully, click **Next** to continue. If LifeKeeper has detected a problem, an ERROR will appear in the information box, the partially created resource hierarchy will be removed, and the **Next** button will be disabled. In that case, click **Cancel** to exit the **Wizard**.

Note: If using teaming software or if network cards are changed after creating a switchable IP resource, the switchable IP resource should be deleted and recreated as the associated index number for the card can change.

IP Local Recovery Scenario

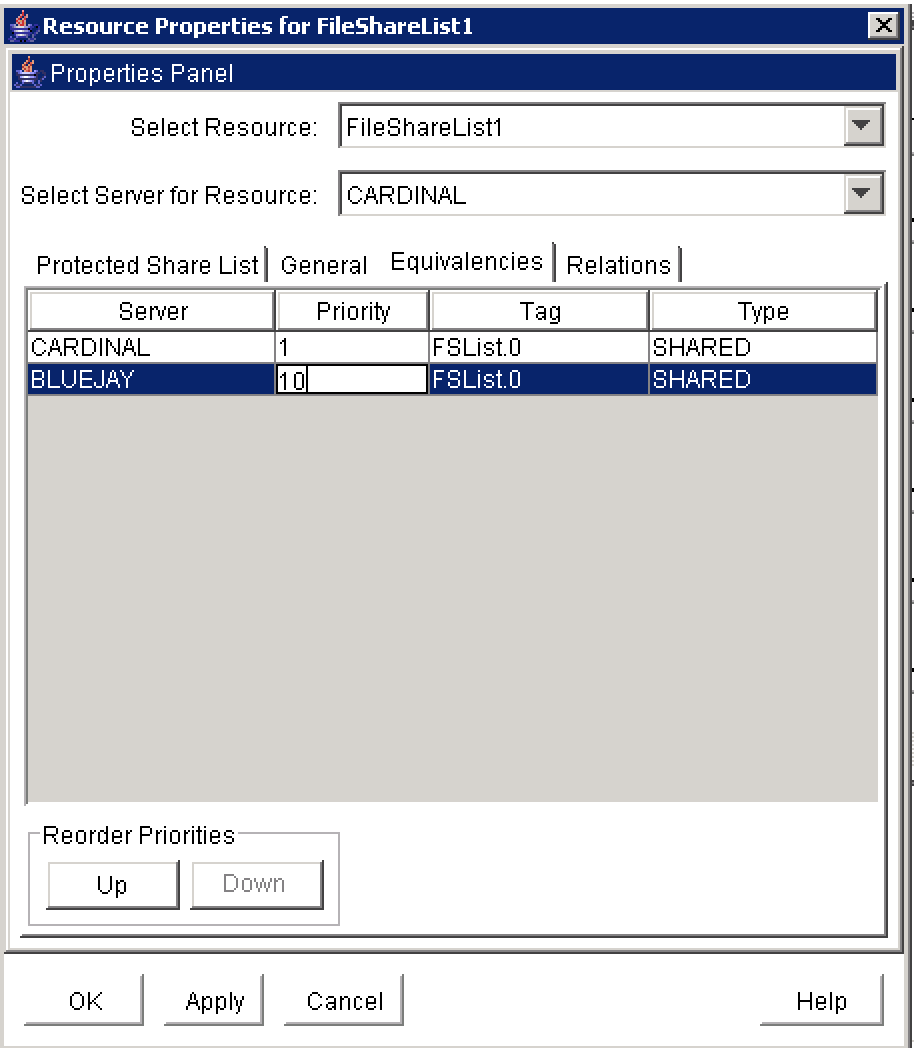
When IP Local Recovery is enabled and the IPresource fails its deepcheck (a periodic extensive check of the IPresource) then LifeKeeper will do the following:

- First, LifeKeeper will attempt to bring the IP address back in-service on the current network interface.
- If that fails, LifeKeeper will check the resource instance to determine if there is a backup interface available. If so, it will then attempt to move the IP address to the backup interface.
- If all local recovery attempts fail, LifeKeeper will perform a failover of the IP address and all dependent resources to a backup server.

Even if you do not have a backup adapter, you can enable Local Recovery so that LifeKeeper will retry the primary network interface before initiating failover to a backup server.

Editing Resource Priorities

You can edit or reorder the priorities of servers on which a resource hierarchy has been defined. First, bring up the **Resource Properties** dialog just as you would for viewing resource properties. The **Resource Properties** dialog displays the priority for a particular resource on a server in the **Equivalencies t**ab as shown below.



Resource Properties for FileShareList1

Properties Panel

Select Resource: FileShareList1

Select Server for Resource: CARDINAL

Server	Priority	Tag	Type
CARDINAL	1	FSList.0	SHARED
BLUEJAY	10	FSList.0	SHARED

There are two ways to modify the priorities:

- Reorder the priorities by moving an equivalency with the **Up/Down** buttons ,or
- Edit the priority values directly.

Using the Up and Down Buttons

1. Select an equivalency by clicking on a row in the Equivalencies table. The **Up** and/or **Down** buttons will become enabled, depending on which equivalency you have selected. The **Up** button is enabled unless you have selected the highest priority server. The **Down** button is enabled unless you have selected the lowest priority server.
2. Click **Up** or **Down** to move the equivalency in the priority list.

The numerical priorities column will not change, but the equivalency will move up or down in the list.

Editing the Priority Values

1. Select a priority by clicking on a priority value in the **Priority** column of the **Equivalencies** table. A box appears around the priority value, and the value is highlighted.
2. Enter the desired priority and press **Enter**.

   Note: Valid server priorities are **1** to **999**

   After you have edited the priority, the **Equivalencies** table will be re-sorted.

Applying Your Changes

Once you have the desired priority order in the **Equivalencies** table, click **Apply** (or **OK**) to commit your changes. The **Apply** button applies any changes that have been made. The **OK** button applies any changes that have been made and then closes the window. The **Cancel** button closes the window without saving any changes made since **Apply** was last clicked.

Incomplete Resource Priority Modification

A hierarchy in LifeKeeper is defined as all resources associated by parent/child relationships. For resources that have multiple parents, it is not always easy to discern from the GUI all of the root resources for a hierarchy. In order to maintain consistency in a hierarchy, LifeKeeper requires that priority changes be made to all resources in a hierarchy for each server. The GUI enforces this requirement by displaying all root resources for the hierarchy selected after the **OK** or **Apply** button is pressed. You have the opportunity at this point to accept all of these roots or cancel the operation. If you accept the list of roots, the new priority values will be applied to all resources in the hierarchy.

You should ensure that no other changes are being made to the hierarchy while the Resource Properties dialog for that hierarchy is displayed. Before you have edited a priority in the Resource Properties dialog, any changes being made to LifeKeeper are dynamically updated in the dialog. Once you have begun making changes, however, the values seen in the dialog are frozen even if underlying changes are being made in LifeKeeper. Only after selecting the **Apply** or **OK** button will you be informed that changes were made that will prevent the priority change operation from succeeding as requested.

In order to minimize the likelihood of unrecoverable errors during a priority change operation involving multiple priority changes, the program will execute a multiple priority change operation as a series of individual changes on one server at a time. Additionally, it will assign temporary values to priorities if necessary to prevent temporary priority conflicts during the operation. These temporary values are above the allowed maximum value of 999 and may be temporarily displayed in the GUI during the priority change. Once the operation is completed, these temporary priority values will all be replaced with the requested ones. If an error occurs and priority values cannot be rolled back, it is possible that some of these temporary priority values will remain. If this happens, follow the suggested procedure outlined below to repair the hierarchy.

Restoring Your Hierarchy to a Consistent State

If an error occurs during a priority change operation that prevents the operation from completing, the priorities may be left in an inconsistent state. Errors can occur for a variety of reasons, including system and communications path failure. If an error occurs after the operation has begun, and before it finishes, and the program was not able to roll back to the previous priorities, you will see a message displayed that tells you there was an error during the operation and the previous priorities could not be restored. If this should happen, you should take the following actions to attempt to restore your hierarchy to a consistent state:

1. If possible, determine the source of the problem. Check for system or communications path failure. Verify that other simultaneous operations were not occurring during the same time that the priority administration program was executing.

2. If possible, correct the source of the problem before proceeding. For example, a failed system or communications path must be restored before the hierarchy can be repaired.

3. Re-try the operation from the Resource Properties dialog.

4. If making the change is not possible from the Resource Properties dialog, it may be easier to attempt to repair the hierarchy using the command line `hry_setpri`. This script allows priorities to be changed on one server at a time and does not work through the GUI.

5. After attempting the repair, verify that the LifeKeeper databases are consistent on all servers by executing the `eqv_list` command for all servers where the hierarchy exists and observing the priority values returned for all resources in the hierarchy.

6. As a last resort, if the hierarchy cannot be repaired, you may have to delete and re-create the hierarchy.

Editing Resource Properties

1. To edit the properties of a resource, bring up the **Resource Properties** dialog just as you would for viewing resource properties.

2. If you are logged into that server with the appropriate permissions, the following items will be editable.

    - Switchback

    - Resource Configuration (only for resources with specialized configuration settings)

    - Resource Properties

3. Once you have made changes, the **Apply** button will be enabled. Clicking this button will apply your changes.

4. When you are finished, click **OK** to save any changes and close the window or **Cancel** to close the window without applying changes.

Extending Resource Hierarchies

The LifeKeeper Extend Resource Hierarchy option copies an existing hierarchy from one server and creates a similar hierarchy on another LifeKeeper server. Once a hierarchy is extended to other servers, cascading failover is available for that resource. The server where the existing hierarchy currently resides is referred to as the template server. The server where the new extended hierarchy will be placed is referred to as the target server.

The target server must be capable of supporting the extended hierarchy and it must be able to communicate with equivalent hierarchies on other remote servers (via active LifeKeeper communications paths). This means that all recovery kits associated with resources in the existing hierarchy must already be installed on the target server as well as every other server where the hierarchy currently resides.

Note: When you create a new resource hierarchy, you will be prompted to extend that hierarchy immediately afterwards.

1. To extend an existing resource hierarchy, select that server hierarchy you want to Extend and then select Extend Resource Hierarchy from the resource context menu or resource context toolbar.
2. Select the Backup Server and click **Next**.
3. A dialog will then display the results of LifeKeeper's pre-extend checks. If these tests succeed, LifeKeeper will display a message stating that the pre-extend scripts were successful. Click **Next** to enter any remaining data needed for the specific type of resource hierarchy that you are extending.

Note: ALL roots in a multi-root hierarchy must be extended together, that is, they may not be extended as single root hierarchies.

Extending a DNS Resource Hierarchy

This operation can be started automatically after you have finished creating a DNS resource hierarchy, or from an existing DNS resource hierarchy, as described in the section on extending resource hierarchies. The following additional data is required to extend a DNS resource hierarchy.

Field	Tips
IP Address	Enter the IP address of the *A record* associated with the protected primary server or alias name. The record will be updated with this IP address when the DNS resource is brought in-service on this server.
DNS Server Name (Fully Qualified)	Enter fully qualified name of a DNS server, in format *<DNS ServerName>.<mydomain>.com*, where the Resource Records can be modified. The DNS server should be accessible from the backup server, preferably in the same site.

Extending a File Share Resource Hierarchy

This operation can be started automatically after you have finished creating a file share resource hierarchy, or from an existing file share resource hierarchy, as described in the section on extending resource hierarchies. No additional data is required to extend a file share resource hierarchy.

Extending a Generic Application Resource Hierarchy

This operation can be started automatically after you have finished creating a generic application resource hierarchy, or from an existing generic application resource hierarchy, as described in the section on extending resource hierarchies. No additional data is required to extend a generic application resource hierarchy.

Extending a LAN Manager Resource Hierarchy

This operation can be started automatically after you have finished creating a LAN manager resource hierarchy, or from an existing LAN manager resource hierarchy, as described in the section on extending resource hierarchies. No additional data is required to extend a LAN manager resource hierarchy.

Extending a Volume Resource Hierarchy

This operation can be started automatically after you have finished creating a volume resource hierarchy, or from an existing volume resource hierarchy, as described in the section on extending resource hierarchies. The following additional data is required to extend a volume resource hierarchy.

Field	Tips
Volume Type	Select **Shared Disk** if using shared storage, **Create Mirror** if using SteelEye DataKeeper and the mirror does not exist, or **Existing Mirror** if using SteelEye DataKeeper and the mirror has already been created.
Network end points (Target/Source)	If Volume Type **Create Mirror** or **Existing Mirror**, select the network end points for the mirror. End points must be IP addresses.
Mode	If Volume Type **Create Mirror**, then select the mode of the mirror. **Asynchronous Mirror**: Source writes are queued for transmission to the target, and return immediately. Less reliable than synchronous, but source writes are quicker. **Synchronous Mirror**: All writes to the source volume will be committed to the target volume immediately. Higher reliability, lower performance.
When extending the volume resource to a third system in the cluster, you must specify the volume type for each of the equivalent systems in the cluster.	
Volume Type (Shared or SteelEye DataKeeper))	Select **Shared Disk** or the network end points for the mirror between the equivalent systems.

Note: Mirrors created from the LifeKeeper GUI will be deleted when the volume resource hierarchy is deleted. To prevent the mirror deletion, set the LifeKeeper Delete Mirror Flag to **False**.

Extending an IP Address Resource Hierarchy

This operation can be started automatically after you have finished creating an IP address resource hierarchy, or from an existing IP address resource hierarchy, as described in the section on extending resource hierarchies. The following additional data is required to extend an IP address resource hierarchy.

Field	Tips
Subnet Mask	Enter the subnet mask to use for the IP resource on the target server. LifeKeeper will by default offer the subnet mask used on the template server.
Network Connection	Select the network connection to use on the target server.
Target Restore Mode	This feature applies to three node LifeKeeper clusters where two nodes are on a LAN (same subnet) and the third node is on a WAN (different subnet). The restore mode of the IP resource would be **enabled** on the LAN nodes and **disabled** on the WAN node. Select the appropriate *Restore Mode* for this IP resource on the target system. In some situations a protected IP address should not be used on a remote target system. For example, the remote target system may be connected to a different subnet than other systems in the cluster. In this situation the IP resource may be extended using the "Disable" Restore Mode. When using the "Disable" Restore Mode option, LifeKeeper will not configure the IP address on the target system when the resource is placed in-service there and monitoring for the IP resource will be disabled. In these situations, network redirection may be implemented some other way or by using a LifeKeeper DNS resource. You may use the IP resource properties page on the target system to change your selection at a later time. See Managing IP Resources.
Target Local Recovery	Click **Yes** if you wish to enable IP Local Recovery on the target server; otherwise choose **No**.
Target Local Recovery Network Connection	If you answered **Yes** to **Local Recovery**, you must select a network connection to use as the backup interface. Specify the backup NIC if one exists; otherwise, specify the primary NIC.

Note: A disabled IP resource can not be extended to another system. As part of the extend operation, a ping command is performed to verify that the IP resource is reachable from the remote system. In the case of a disabled IP resource, the IP resource will be In-service, but will not respond to the ping command.

Unextending a Hierarchy

The **Unextend Resource Hierarchy** option removes a complete hierarchy, including all of its resources, from a single server. This is different than the Delete Resource Hierarchy selection which removes a hierarchy from all servers.

When using **Unextend Resource Hierarchy**, the server from which the existing hierarchy is to be removed is referred to as the target server.

The **Unextend Resource Hierarchy** selection can be used from any LifeKeeper server that has active LifeKeeper communications paths to the target server.

1. Select a server-specific resource instance from the hierarchy that you want to unextend, and then select **Unextend Resource Hierarchy** from the resource context menu or resource context toolbar.
2. The dialog will display a message verifying the server and resource hierarchy that you have specified to be unextended. Click **Unextend** to perform the action.
3. If the output panel is enabled, the dialog closes, and the results of the commands to unextend the resource hierarchy are shown in the output panel. If not, the dialog remains up to show these results, and you click **Done** to finish when all results have been displayed.

Adding a Resource Dependency

While most Recovery Kits create their dependencies during the original resource hierarchy creation task, under certain circumstances, you may want to create new or additional resource dependencies or delete existing ones. An example might be that you wish to change an existing IP dependency to another IP address. Instead of deleting the entire resource hierarchy and creating a new one, you can delete the existing IP dependency and create a new dependency with a different IP address.

1. Select a server-specific resource instance as the parent to which you want to add a child dependency, and then select **Add Dependency...** from the resource context menu or resource context toolbar.
2. Select a **Parent Resource IP Address** from the drop down box. Click **Next**.
3. Select a **Child Resource Tag** from the drop down box of existing and valid resources on the server. The dialog will display all the resources available on the server with the following exceptions:

    - The parent resource, its ancestors and its children.
    - A resource that has not been extended to the same servers as the parent resource.
    - A resource that does not have the same relative priority as the parent resource.
    - Any resource that is not in-service on the same server as the parent, if the parent resource is in-service.

    Click **Next** to proceed to the next dialog.
4. The dialog will then confirm that you have selected the appropriate parent and child resource tags for your dependency creation. Click **Add Dependency** to create the dependency on all servers in the cluster to which the parent has been extended.
5. If the output panel is enabled, the dialog closes and the results of the commands to create the dependency are shown in the output panel. If not, the dialog remains up to show these results, and you click **Done** to finish when all results have been displayed.

Removing a Resource Dependency

1. Select a server-specific resource instance as the parent from which you want to remove a

child dependency, and then select **Remove Dependency** from the resource context menu or resource context toolbar.

2. Select the **Child Resource** from the drop down box. This should be the name of the child in the dependency that you want to delete. Click **Next** to proceed to the next dialog box.

3. The dialog then confirms that you have selected the appropriate parent and child resource tags for your dependency deletion. Click **Remove Dependency** to delete the dependency on all servers in the cluster.

4. If the output panel is enabled, the dialog closes, and the results of the commands to delete the dependency are shown in the output panel. If not, the dialog remains up to show these results, and you click **Done** to finish when all results have been displayed.

Deleting a Hierarchy from All Servers

1. Select a server-specific resource instance in the hierarchy that you want to delete, and then select **Delete Resource Hierarchy** from the resource context menu or resource context toolbar.

2. The dialog will display a message verifying the hierarchy you have specified for deletion. Click **Delete** to perform the action. Deletion will begin on the sever that you initially selected.

3. If the output panel is enabled, the dialog closes, and the results of the commands to delete the hierarchy are shown in the output panel. If not, the dialog remains up to show these results, and you click **Done** to finish when all results have been displayed.

Man Pages

- LCD - Miscellaneous LCD Programs
- LCDI Applications
- LCDI Flags
- LCDI Instances
- LCDI Relationship
- LCDI Resource Type
- LCDI Systems

LCD - Miscellaneous LCD Programs

Synopsis

lcdremexec [-e] -d destname -- cmd [arg1 arg2 ... argn] lcdsync [-d destname]

lcdrecover -g {remote|restore|delete} -- [arg1 arg2 ... argn] | -G {remote|restore|delete} -- [arg1 arg2 ... argn] | -p primarytest /| [-o resource]

lcdrcp file1 file2 file3 ... {dest:ofile | dest:odir}

lkstart [-w waitperiod]

lkstop [-f or -r|-n]]

Description

These programs have various uses by application developers. They are all found in the directory `%LKROOT%\bin`.

Exit Codes

The following exit codes could be returned by these commands:

0	The operation has succeeded.
1	A system call or library call has internally returned failure.
2	A user-specified syntax error occurred.
3	LifeKeeper internal error.
4	A request to perform an operation on an object that already exists.
5	An argument specified is illegal.
6	Index out-of-range.
7	A request has been made on an object that does not exist.
8	A request was made to delete a resource instance on which another non-deleted resource instance depends.
9	An attempt to communicate with another system failed.

lcdrcp

```
lcdrcp file1 file2 file3 ... {dest:ofile | dest:odir}
```

lcdrcp is a general purpose program used to transfer the ASCII files *file1 file2 file*3 ... to another system using the LifeKeeper communications path. Binary files cannot be copied using **lcdrcp**.

LifeKeeper transfers the files to dest in the directory *odir*. If only one file is sent, the alternate form including the destination file name at location *ofile* on system dest is provided. Take extra caution while using Windows drive names (like `D:`), as destination arguments as they could be misinterpreted as destination names if a destination name is missing. However, if a destination system name is specified, drive names are interpreted properly.

lcdrecover

```
lcdrecover -g {remove|restore|delete} -- [arg1 arg2 ... argn] | -G
{remote|restore|delete} -- [arg1 arg2 ... argn] | -p primarytest | [-o
resource]
```

The *-g* option takes one of three arguments, *remove, restore, or delete*. This option will run the preglobal scripts for the specified argument. The preglobal scripts are registered by applications to run before certain events. For example, with the restore argument, this option runs the prerestore script registered by LifeKeeper, then any prerestore scripts registered by all of the applications. Normally, **perform_action** [see LRACI-perform_action] automatically performs the prerestore scripts, except when the *-G* option is specified to **perform_action**.

The *-G* option of **perform_action** allows multiple **perform_action** commands to be run, with the preglobal scripts running only once before the first **perform_action** execution using **lcdrecover** *-g restore*. An application may register a preglobal script by installing the script at the path:

```
%LKROOT%\ subsys\<appname>\actions\prerestore.ksh
```

arg1, arg2, ... argn are arguments that will be passed to the preglobal scripts when they are executed.

Similar scripts (preremove) exist for the remove argument. They can be run before a **perform_action** *-G -a remove* is run. They are run when **lcdrecover** *-g remove* is executed.

The predelete scripts are similar, but they are run before the **ins_remove** *-G* ... [see LCDI-instance] program is run, unless *-G* for **ins_remove** is left out.

The *-G* option for **lcdrecover** is analogous to *-g*, except that it specifies that the postglobal scripts should be run. The *-G* option should not be used without running an earlier **lcdrecover** *-g arg*, and it should be run after all of the **perform_action** or **ins_remove** programs are run. If you are executing the *-G* option within a **getlocks** protected region (after **getlocks** and before **rlslocks**), set *arg1* to *-m* to avoid executing a second instance of **getlocks**, which would cause the operation to hang.

The following example runs multiple **perform_action** commands where the preglobal and postglobal scripts run only once:

```
     lcdrecover -g restore

     # run "preglobal" restore scripts

     perform_action -G -a restore -t tagname

     # neither scripts are run

     perform_action -G -a restore -t tagname2

     # neither scripts are run

     lcdrecover -G restore -- -m
```

```
    # run "postglobal" restore scripts

    # use -m arg when in getlocks protected region of code
```

This example runs multiple prerestore and postrestore scripts:

```
    perform_action -a restore -t tagname

    # all scripts once

    perform_action -a restore -t tagname2

    # all scripts again
```

The *-p* option for **lcdrecover** is used to determine if a particular resource is on a resource hierarchy that is on the primary system or the secondary system. Specify the resource tag name with primary test, and it will print out to standard output the string primary if the resource is on the primary hierarchy, or secondary if it is not.

The *-o* option can be used to retrieve the remote system associated with the resource tag specified.

lcdremexec

```
lcdremexec [-e] -d destname -- cmd [arg1 arg2 arg3 ... argn]
```

This program sends a remote request over the LifeKeeper communication paths to the system *destname*, to execute the command **cmd** remotely with arguments *arg1 arg2 arg3 ... of the* and returns the standard output and standard error of the remote command to standard output of the **lcdremexec** command. The exit code of the remote command is returned by **lcdremexec**.

Note: If destname is the current system, no messages are sent; **lcdremexec** will execute it locally.

The *-e* option will split standard output and standard error of the remote command and first print standard output of the remote command to standard output of **lcdremexec**, then print standard error of the remote command to standard error of the **lcdremexec** command. This option has no effect for local commands, which have their standard output and standard error unchanged.

cmd can be either a Korn shell script or a Win32 executable. It will be executed with %LKROOT% on *destname* as the current working directory, thus being able to accept path names relative to %LKROOT%.

Before executing, the directory `%LKROOT%\BIN` is always added to the head of the PATH variable on *destname*. If *destname* is DEAD or goes DEAD in the middle of the execution, **lcdremexec** returns a non-zero exit code.

lcdsync

```
lcdsync [-d destname]
```

This program checks to see if the LifeKeeper resource hierarchy configuration and communication path status data stored in shared memory has been modified. If it is different, the data is "synchronously" written to disk. Therefore, when this program returns, the data is guaranteed to be on disk properly. If *destname* is not specified, the current system is assumed.

Note: The commands used to modify resource hierarchy configurations or communication paths (such as **ins_create**, **dep_create**, **ins_setit**, **eqv_remove**,...) only modify the shared memory segment and are not reflected in the permanent file storage of LifeKeeper, until the **lcdsync** program is run.

lkstart

```
lkstart [-w waitperiod]
```

This program starts up LifeKeeper on the current system if it is currently not running. **lkstart** modifies entries in the `%LKROOT%\etc\LKinit.config` file pertaining to the LifeKeeper daemons so that they will be respawned if they die.

The *-w* option, with *waitperiod* in seconds, can be used to change the timeout interval. Use the *-w* argument to specify a wait period before the startup.

The LifeKeeper service can be started using the Services mmc under Administrative Tools, or from a command prompt using either "`sc start LifeKeeper`" or "`net start LifeKeeper`"'

Note: This program must be run from the console.

lkstop

```
lkstop [-n] [-f] [-r]
```

This script shuts down LifeKeeper on the system, if it is currently running. LifeKeeper will automatically restart at system boot.

The table below describes the actions taken by LifeKeeper when each **lkstop** option is entered:

Command Line	Action
`lkstop`	Resources in service are removed from service and are NOT switched over to a backup server.
`lkstop -n`	Same as **lkstop** with no options specified.
`lkstop -f`	Resources in service do not get removed from service.
`lkstop -r`	Same as -f.

The LifeKeeper services can also be stopped using the using the Services tool under Administrative Tasks in the Windows Control Panel.

LCDI Applications

Synopsis

app_create [-d destsys] -a appname

app_remove [-d destsys] -a appname

app_list [-d destsys]

Description

A LifeKeeper application is a group of related resource types. When an application is removed, all resource types installed under it are also removed.

These programs provide an interface for generating new applications in the configuration database and removing existing ones. All commands exit 0, if successful, and a nonzero code (see EXIT CODES section) and an error message prints to standard error for failure.

Exit Codes

The following exit codes could be returned by these commands:

0	The operation has succeeded.
1	A system call or library call has internally returned failure.
2	A user-specified syntax error occurred.
3	LifeKeeper internal error.
4	A request to perform an operation on an object that already exists.
5	An argument specified is illegal.
6	Index out-of-range.
7	A request has been made on an object that does not exist.
8	A request was made to delete a resource instance on which another non-deleted resource instance depends.
9	An attempt to communicate with another system failed.

app_create

```
app_create [-d destsys] -a appname
```

Creates a new application. The application is tied to system *destsys*, using the name called *appname*. If *destsys* is not supplied, the application is created locally.

app_list

```
app_list [-d destsys]
```

This command prints to standard output a list of applications that have installed software to work with LifeKeeper on the system *destsys*. If *destsys* is not specified, the current system is assumed.

app_remove

```
app_remove [-d destsys] -a appname
```

Removes the given application from the configuration database set of known applications. All resource types, resource instances, and equivalencies relating to this application are also removed. Failure can occur because the application is not known to the configuration database.

LCDI Instances

Synopsis

ins_gettag [-d destsys] -i id

ins_create [-d destsys] -a appname -r restyp [-I{AUTORES_ISP|INIT_ISP| INIT_OSU}] [-v info] -t tag -i id [-Q quickChkInt] [-DdeepChkInt] [-l localRecover{Y/N}] [-s AUTOMATIC/INTELLIGENT]

ins_remove [-d destsys] [-R roottag] [-a appname] [-r restyp] [-ttag] [-i id] [-v] [-I] [-N] [-G]

ins_setin [-d destsys] -t tag [-v info]

ins_setit [-d destsys] -t tag -I {AUTORES_ISP|INIT_ISP|INIT_OSU}

ins_setst [-d destsys] -t tag -S {ISP|ISU|OSU} [-R reason] [-A]

ins_list [-d destsys] [-fC] [-R top] [-a appname] [-r typ] [-t tag] [-i id]

ins_setchkint [-d destsys] -t tag -c {q=quick|d=deep} -vinterval

ins_setlocalrecover [-d destsys] -t tag -l {Y=enable|N=disable}

ins_setas [-d destsys] -t tag -s {INTELLIGENT|AUTOMATIC}

Description

Resources are used by LifeKeeper to represent volumes, applications, or system objects known by the system. Resource types are classifications of resources; resource instances are actual instances of a resource type. For example, resource types would include file system volumes, file shares, IP addresses, LAN Manager names and various servers like SQLServer. Generic, user-definable types permit users to build custom fault resilient setups. Multiple instances may exist for a resource type.

Resource instances may exist in a number of states. These states may take on the following values and meanings:

ISP	Resource is in service, protected. ISP is the normal state of resources on the primary node.
OSU	Out of service, unimpaired. The resource is not available on this system because it was brought out of service by executing its remove script. The OSU state also is used for objects that have dependencies on children in the OSF or OSU state or when the equivalent object on the backup machine is in the ISP or ISU state. OSU is the normal state of resources on the secondary node.
OSF	Out of service due to a failure. The resource is not available on this system because a failure has occurred trying to restore the object.

Exit Codes

All commands exit 0 if successful. For a failure, all commands exit to a nonzero code and prints to standard error. The following exit codes could be returned by these commands:

0	The operation has succeeded.
1	A system call or library call has internally returned failure.
2	A user-specified syntax error occurred.
3	LifeKeeper internal error.
4	A request to perform an operation on an object that already exists.
5	An argument specified is illegal.
6	Index out-of-range.
7	A request has been made on an object that does not exist.
8	A request was made to delete a resource instance on which another non-deleted resource instance depends.
9	An attempt to communicate with another system failed.

ins_list

```
ins_list [-d destsys] [-fC] [-R top] [-a appname] [-r typ][-t tag] [-i
id]
```

This command prints lines relating to a set of resource instances to standard output. Each line contains all of the current information known about a particular resource instance. Examples of the lines are:

```
LKSYS1-filesys-volume-ISSUTILS-K:--ISP-restore action hassucceeded-
AUTORES_ISP--0-0-
```

Each line contains fields separated by a delimiter character. The default delimiter character is ^A (\001). If the *-fC* option is specified, the delimiter is changed to the specified character. The above example shows a dash (-) as a delimiter. The fields in the example command above are as follows:

LKSYS1	Name of the system the resource the instance resides on.
filesys	Application name of resource type.
volume	Resource type name.
ISSUTILS	User-defined resource instance tag identifier.
K:	LifeKeeper internal identifier for resource instance.
- -	If this field is not empty, as in the example, it provides additional instance information (type dependent).
ISP	Current state of resource instance ISP, ISU, OSU, or OSF.
restore action has succeeded	Reason for last state change.
AUTORES_ISP	Available resource initialization options are: AUTORES_ISP, INIT_ISP, and INIT_OSU.
- -	If this field is not empty, as in the example, it indicates that the resource is currently being reserved for: RESTORE: restoring the resource to service REMOVE: removing the resource from service RECOVER: performing local recovery on resource
0	Process ID of process that has reserved resource.
0	Reserved.
180	Quick check interval, in seconds.
300	Deep Check interval, in seconds.
0	Local recovery option. 0 = disabled; 1 = enable

The other arguments limit the number of resource instances included in the list. If none of the arguments are used, then all resources on *destsys* are listed. These are the limiting arguments:

destsys. If *destsys* is not specified, the current system is assumed; otherwise, data from the remote system is printed.

top. If *top* is the space string " ", only the root resources will be printed. If top is specified (but not the space string), the report lists the top resource and all children resources below it, recursively.

appname. If *appname* is specified, all resource instances associated with all resource types defined by this application are printed. If appname is not specified, all resource instances for all applications defined on the system are printed.

typ. If *typ* is specified, all resource instances of type, *typ*, in application *appname* are printed.

tag or *id*. If *tag* or *id* is specified, the resource instance associated with that *tag* or *id* is printed.

Initialization Strategy

It is recommended that you accept the default Auto ISP Initialization Strategy. These are the actions taken when LifeKeeper starts (initializes):

Autores ISP	Resource is automatically brought into service if it is not in service on the paired node.
Init ISP	Resource is always initialized into the ISP state.
Init OSU	Resource is always initialized into the OSU state.

Initial State

The state is the current processing status for the resource. For example, the normal state for a resource on the primary system is **ISP** - in service, protected. The normal state for a resource on the secondary system is **OSU** - out of service, unimpaired.

It is recommended that you accept the default initial state. If you set the Initial State to **OSU**, you must manually bring the resource into service.

ins_create

```
ins_create [-d destsys] -a appname -r restyp [-I {AUTORES_ISP|INIT_
ISP|INIT_OSU}][-v info] -t tag -i id [-Q quickChkInt][-D deepChkInt][-l
localRecover{Y|N}] [-s AUTOMATIC|INTELLIGENT]
```

Defines a new resource instance on system *destsys* in the configuration database. The resource instance is described by the arguments given. If *destsys* is not specified, the current system is assumed. The command offers the following string tag options:

- The *-a* and *-r* options indicate the preexisting application and resource type associated with this new instance.
- Initialization type field specified by the *-I* option indicates how the resource instance should be initialized if LifeKeeper restarts (for example, at boot time).
- Optional string info specified by the *-v* option is a field that can contain additional resource type specific information and does not necessarily have to be unique per resource type.
- String tag specified by the *-t* option is a string that names the resource instance and is unique

on a system. It is a string that is meaningful externally to LifeKeeper.

- String id specified by the *-i* option is also unique per system, but may be meaningful only internally to LifeKeeper.
- Quick check interval provided with -*Q* option should be in seconds. The value should be zero if **quickchk.ksh** script doesn't exist for the resource. LifeKeeper waits this interval time between two consecutive execution of **quickchk.ksh** script. Valid range of value: 0 - 604800.
- Deep check interval provided with *-D* option should be in seconds. The value should be zero if **deepchk.ksh** script doesn't exist for the resource. LifeKeeper wait this interval time between two consecutive execution of **deepchk.ksh** script. Valid range of value: 0 - 604800.
- Local recover option indicates whether resource should be recovered byexecuting **recover.ksh** script. This option should be "*N*" if **recover.ksh** script doesn't exist for the resource.

ins_gettag

```
ins_gettag [-d destsys] -i id
```

Prints to standard output the tag name that corresponds to the internal identifier provided in *id* on the system with name *destsys*. If *destsys* is not specified, the current system is assumed.

Note: The tag name and *id* name for a resource are unique on a system, but may be reused to indicate different resource instances on different systems.

The resource tag provides an understandable handle (human readable name) for a resource instance, for example, `user-partition`, whereas the *id* is an internal descriptor. The resource name *id* is used by the application software associated with the resource to uniquely describe the resource.

ins_remove

```
ins_remove [-d destsys] [-R roottag] [-a appname] [-r restyp][-t tag] [-
i id] [-v] [-I] [-N] [-G]
```

Removes resource instance(s) on system *destsys* from the configuration database. Associated dependencies and equivalencies will also be removed. If *destsys* is not specified, the current system is assumed.

Note: All resources that depend upon any of the selected resources directly or indirectly will also be removed before the selected resource is removed.

When a resource instance is removed, and if a delete action was defined for the resource type of the instance being removed, the delete action is run before the instance is removed.

The command has the following options:

-R	The *-R* option is for removing entire sub-hierarchies and the resources that depend on them. The *roottag* string defines a list of instance tag names (separated by the ^A character) for which these resources and the resources below on the hierarchy will be recursively removed, until a resource is encountered for which a resource not being removed depends.
-a	If the *-a* option is specified, only resources from that application will be removed.
-r	If the *-r* option is specified, all resources of the specified resource type will be removed.
-t or *-i*	If the *-t* option or *-i* option is specified, the instance with the matching *tag* or *id* will be removed along with the resources that depend on them.
-v	If the *-v* option (verbose) is specified, the function prints a message to standard output including the tag names of all the removed resource instances.
-I	The *-I* option initializes the resource hierarchy so that **ins_remove** can work properly. This option should be used by the"highest-level" recursive call to **ins_remove**, but not necessarily in a lower-level recursive call such as inside a delete script. The *-I* option should NOT be used by a recursive call to ins_remove from inside a delete script.
-N	The *-N* option tells **ins_remove** NOT to reinitialize the resource hierarchy. The assumption when using this option is that a higher level call of **ins_remove** will perform the *-I* option. The *-N* option MUST be used inside a delete script, since a delete script is being called from a parent invocation of **ins_remove** and the *-N* option prevents hierarchy cycles from occurring.
-G	The *-G* option indicates that the predelete and postdelete scripts [see LCD] should not be performed as part of this call to **ins_remove**. This option is useful if you wish to perform multiple top-level calls to **ins_remove** and have the predelete and postdelete run manually [using **lcdrecover**-*g delete* of LCD] before and after (respectively) the calls to **ins_remove**. It would also be wise to use the *-G* option in the delete scripts since the highest-level **ins_remove** should perform the predelete and postdelete scripts.

ins_setas

```
ins_setas [-d destsys] -t tag -s {INTELLIGENT|AUTOMATIC}
```

Sets the switchback type of a root resource on system *destsys* with *tag* name *tag* to the strategy specified in the *-s* option. Use only on root resource to change the switchback type of the root resource and all its dependent resources.

ins_setchkint

```
ins_setchkint [-d destsys] -t tag -c {q=quick|d=deep} -v interval
```

This command modifies the quick check or deep check interval for the resource specified by the *tag* name "*-t*". The interval must be entered in seconds.

Examples:

To change the quick check interval for the file share resource FSList.0 to two minutes, run the following command from `$LKROOT\bin`:

```
ins_setchkint-t FSList.0 -c quick -v 120
```

To disable the deep check for the file share resource FSList.0, run the following command from `$LKROOT\bin`:

```
ins_setchkinst -t FSList.0 -c deep -v0
```

ins_setin

```
ins_setin [-d destsys] -t tag [-v info]
```

The string *info* specified by the *-v* option is a field that can contain additional resource type specific information and does not necessarily have to be unique per resource type. For example, an instance of a resource of file share type will have the names of all shares managed by this instance in its *info* value.

ins_setit

```
ins_setit [-d destsys] -t tag -I {AUTORES_ISP | INIT_ISP | INIT_OSU}
```

Indicates to LifeKeeper how it should initialize the state of a resource when LifeKeeper itself initializes (for example, at system boot time). If you do not set this option, LifeKeeper sets the initialization state to default options.

These are the restore options you can specify:

AUTORES_ISP. If resource initialization is set to *AUTORES_ISP*, the resource is first set to the OSU state, then the restore action is performed and, if successful, the resource is put into the ISP state. If restore fails, the resource is placed into the OSF state.

INIT_ISP. If *INIT_ISP* is set, LifeKeeper assumes resource initialization by other means and places the resource into the ISP state.

INIT_OSU. If *INIT_OSU* is set, LifeKeeper assumes the resource is not started up during initialization and that the system administrator will manually start up the resource using LifeKeeper Graphical User Interface (GUI) application.

ins_setlocalrecover

```
ins_setlocalrecover [-d destsys] -t tag -l {Y=enable|N=disable}
```

This command modifies the local recovery setting for the resource specified by the *tag* name "-t".

Example:

To disable local recovery for the file share resource FSList.0, run the following command from `$LKROOT\bin`:

```
ins_setlocalrecover -t FSList.0 -l N
```

ins_setst

```
ins_setst [-d destsys] -t tag -S {ISP|ISU|OSU} [-R reason] [-A]
```

Sets the resource state on system *destsys* with tag name *tag* to the state specified in the *-S* option. If *destsys* is not specified, the current system is assumed. Use this command cautiously because the resource state will be changed by the resource's action script (e.g. remove or restore script). The caller is responsible for making sure the new state reflects the actual state of the application.

Additional text explaining the reason for the change of state may be provided by the *-R* option. The *-A* option sets the state of the specified resource and all the resources that depend on it, recursively up the hierarchy.

LCDI-relationship

Synopsis

dep_create [-d destsys] -p partag -c chdtag

dep_remove [-d destsys] [-p partag] [-c chdtag]

dep_list [-d destsys] [-fC] [-C allchild | -P allparent | -c ofparenttag | -p ofchildtag] [-r typ] [-a app]

eqv_create [-d destsys] [-s sys] -t tag [-p sysPriority] [-Sothersys] -o othertag [-r othersysPriority] -e SHARED

eqv_remove [-d destsys] [-s sys] -t tag [-S othersys] -o othertag -e SHARED

eqv_list [-d destsys] [-s sys] [-t tag] [-fC]

Description

LifeKeeper resources exist in relationship to one another. Two resources may be unrelated or they may be in a dependency relationship. In a hierarchy, a resource may have several resources depending upon it,and it may depend upon several resources. Each resource also relates to a like resource on the paired system with a shared equivalency. This shared equivalency ensures that a resource is active on only one system at a time. Equivalency object also indicates priority of the system for the resource. This priority value determines the order of cascading failover. A higher priority system has precedence over a lower priority system in recovering the resource. A priority value of 1 is the highest. The higher the numerical value the lower the priority. Two systems can't be assigned same priority for a resource. Valid range is 1 to 1024.

Exit Codes

All commands exit 0, if successful, and a nonzero code and prints to standard error, for failure. The following exit codes could be returned by these commands:

0	The operation has succeeded.
1	A system call or library call has internally returned failure.
2	A user-specified syntax error occurred.
3	LifeKeeper internal error.
4	A request to perform an operation on an object that already exists.
5	An argument specified is illegal.
6	Index out-of-range.
7	A request has been made on an object that does not exist.
8	A request was made to delete a resource instance on which another non-deleted resource instance depends.
9	An attempt to communicate with another system failed.

dep_create

```
dep_create [-d destsys] -p parent -c child
```

This function creates a dependency relationship between the resource instances with tags *parent* and *child*. Both resources must be on the same system *destsys*. If *destsys* is not specified, the current system is assumed. This implies the *parent* resource now requires the *child* for proper operation. Both resource instances must already exist and must be in the same state (ISP or OSU) for proper operation.

dep_list

```
dep_list [-d destsys] [-fC] [-C allchild | -P allparent | -c ofparenttag
| -p ofchildtag] [-r typ] [-a appname]
```

This function prints strings to standard output describing dependency relationships between resource instances. If *destsys* is not specified, the current system is assumed. Each string is in the following form:

```
LK0-LKSYS:135.66.249.201

LK0-LKSYSA:FSLIST.0

FSLIST0:fi.vo.0
```

There are two fields in each string that are separated by a delimiter character. The default delimiter character is ^A (\001). If the *-fC* option is specified, the delimiter is changed to the character specified. The example above shows a colon (:) as a delimiter. The first field indicates the parent tag name of the relationship and the field on the right is the child tag name.

You can use options to limit the contents of the list. If you use **no** options, all dependencies are printed. The command has the following options:

-C	If the -C option is specified, this command will print out all direct and indirect child dependencies of the resource specified in *allchild*.
-P	If the -P option is specified, this command will print out all direct and indirect parent dependencies of the resource specified in *allparent*.
-c	If the -c option is specified, this command will print out only the direct child dependencies of the resource specified in *ofparenttag*.
-p	If the -p option is specified, this command will print out only the direct parent dependents of the resource specified in *ofchildtag*.
-r	Specifying the -r option lists all the dependencies of child *typ*.
-a	Specifying the -a option lists all the dependencies of application *appname*.

dep_remove

```
dep_remove [-d destsys] [-p parent] [-c child]
```

Removes the dependency relationship(s) from the database on system *destsys*. If *destsys* is not specified, the current system is assumed. If *child* is not specified, all dependencies of *parent* are removed. If *parent* is not specified, all dependents with tag *child* are removed. If both are not specified, all dependencies are removed.

eqv_create

```
eqv_create [-d destsys] [-s sys] -t tag [-p sysPriority][-S othersys] -o
othertag [-r othersysPriority] -e SHARED
```

Creates an equivalency in the configuration database on system *destsys* (local, if not specified).

LifeKeeper will automatically add a *SHARED* equivalency on a remote system, if either *sys* or *othersys* is specified. The resource specified as tag on system *sys* will be assumed by LifeKeeper to be the ”primary” resource that runs under normal conditions; the resource specified as *othertag* on system *othersys* will be the ”secondary” resource on the paired system. When LifeKeeper initializes, the primary resource is set depending upon resource initialization set up [see LCDI_instances]. If the spare system boots, LifeKeeper on the spare checks the primary system to see if the primary system is functioning and if the primary resource is in the ISP state. If both cases are true, LifeKeeper puts the secondary resource into the OSU state (resource initialization ignored). If either case is false, the secondary resource will be initialized according to ”resource initialization.” The priority value specified with *-p* option is the priority of system *sys* for the resource *tag*. The priority value specified with *-r* option is the priority of system *othersys* for the resource *othertag*.

eqv_list

```
eqv_list [-d destsys] [-s sys] [-t tag] [-e SHARED] [-fC]
```

This function prints strings to standard output describing equivalency relationships between resource instances. If *destsys* is not specified, the current system is assumed. Each line contains fields separated by a delimiter character. The default delimiter character is ^A (\001). If the *-fC* option is specified, the delimiter is changed to C.

The example listings below show a colon (:) as a delimiter.

```
LKSYSA:135.66.249.201:LKSYSB:135.66.249.201:SHARED

LKSYSA:FSLIST.0:LKSYSB;FSLIST.0:SHARED

LKSYSA:LK0-LKSYSA:LKSYSB:LK0-LKSYSA:SHARED
```

Using `LKSYSA:fi.vo.0:LKSYSB;fi.vo.0:SHARED`, these are the fields:

LKSYSA	Primary system name where the resource resides.
fi.vo.0	Volume resource tag on the primary system.
LKSYSB	System name for the secondary system, where the resource equivalency resides.
fi.vo.0	Volume resource tag for the equivalent resource on the secondary system
SHARED	Equivalency type.

The remaining arguments limit the information output as specified below:

-s sys. This option limits the output to include only the equivalencies relating to the system specified by the *sys* argument.

-t tag. This option limits the output to include only the equivalencies relating to the tag specified by the *tag* argument.

-e SHARED. This option prints all SHARED equivalency information.

eqv_remove

```
eqv_remove [-d destsys] [-s sys] -t tag [-S othersys]-o othertag [-e
SHARED]
```

Removes equivalency from the configuration database on system *destsys* (local if not specified) of equivalency type, specified by the *-e* option, between the resources *tag* and *othertag* existing on systems *sys* and *othersys*, respectively. If *sys* or *othersys* is not specified, the current system is assumed.

LCDI-resource_type

Synopsis

typ_create [-d destsys] -a appname -r restyp

typ_remove [-d destsys] -a appname -r restyp

typ_list [-d destsys] [-fC] [-a appname]

Description

Resources are used by LifeKeeper to represent volumes, applications or other objects known by the system. Resource types are classifications of resources and are distinguished by a common set of recovery procedures that can be applied to all instances. Resource type examples would include:

- File system volumes, for example K:
- File shares, for example UTIL_SHARE
- IP addresses, for example 153.66.232.21.

The `typ_create` and `typ_remove` commands provide an interface for generating new types in the configuration database. The command `typ_list` provides an interface to the configuration database for listing all resource types existing on a specific system.

Exit Codes

All commands exit 0 if successful. On failure, they return a nonzero code (see EXIT CODES section) and print to standard error. The following exit codes could be returned by these commands:

0	The operation has succeeded.
1	A system call or library call has internally returned failure.
2	A user-specified syntax error occurred.
3	LifeKeeper internal error.
4	A request to perform an operation on an object that already exists.
5	An argument specified is illegal.
6	Index out-of-range.
7	A request has been made on an object that does not exist.
8	A request was made to delete a resource instance on which another non-deleted resource instance depends.
9	An attempt to communicate with another system failed.

typ_create

```
typ_create [-d destsys] -a appname -r restyp
```

Creates a new resource type in the configuration database on system *destsys* (local, if not specified). The resource type is named *restyp* and is installed under the already-existing application, *appname*. Failure occurs if the system or application is not known or if the resource type already exists.

typ_list

```
typ_list [-d destsys] [-fC] [-a appname]
```

This command prints to standard output a list of resource types that have been defined on the application, *appname*, installed on system *destsys* (local,if not specified). If *appname* is not specified, all resource types for all applications are printed in the following format:

```
filesys:volume

comm:ip

database:informix
```

The application name is to the left of the delimiter and the resource typename is to the right. Each line contains fields separated by a delimiter character. The default delimiter character is ^A (\001). If the -*fC* option is specified, the delimiter is changed to the specified character. The above example shows a colon (:) as the delimiter.

typ_remove

```
typ_remove [-d destsys] -a appname -r restyp
```

Removes the given resource type from the configuration database set of known resource types of system *destsys* (local, if not specified). All resource instances, dependencies, and equivalencies associated with this type are also removed. Failure occurs if the resource type is not known to the configuration database.

LCDI-systems

Synopsis

sys_create [-d destsys] -s sys

sys_remove [-d destsys] -s sys

sys_getds [-d destsys] -s sys

sys_getst [-d destsys] -s sys

sys_list [-d destsys]

Description

The LifeKeeper configuration database knows about related systems. Because resources and resource types are specific to the systems on which they exist, it is necessary for the configuration database interface to contain the concept of a system.

The LCDI-systems commands return (or create) information into or remove information out of the database.

Exit Codes

All commands exit 0, if successful, and a nonzero code and prints to standard error, for failure. The following exit codes could be returned by these commands:

0	The operation has succeeded.
1	A system call or library call has internally returned failure.
2	A user-specified syntax error occurred.
3	LifeKeeper internal error.
4	A request to perform an operation on an object that already exists.
5	An argument specified is illegal.
6	Index out-of-range.
7	A request has been made on an object that does not exist.
8	A request was made to delete a resource instance on which another non-deleted resource instance depends.
9	An attempt to communicate with another system failed.

sys_create

```
sys_create [-d destsys] -s sys
```

Creates a new system definition in the configuration database on system *destsys* (local, if not specified). The -*s sys* option is required to identify the system to which the system name is assigned.

sys_getds

```
sys_getds [-d destsys] -s sys
```

Prints to standard output the optional text description of why the system has gone to the current state from the database on system *destsys* (local, if not specified).

sys_getst

```
sys_getst -s sys
```

Prints the system's state to standard output as one of the strings:

DEAD	The system is believed to be unavailable.
ALIVE	The system is believed to be available.
UNKNOWN	System state is unavailable.

sys_list

```
sys_list [-d destsys]
```

This command prints to standard output a list of systems that LifeKeeper knows about from the database on system *destsys* (local, if not specified).

sys_remove

```
sys_remove [-d destsys] -s sys
```

Removes a system definition from the configuration database on system *destsys* (local, if not specified). The *-s sys* option is required to identify the system to which the system name is assigned.

LifeKeeper Flags

Near the end of the detailed status display, LifeKeeper provides a list of the flags set for the system. A common type is a Lock LCD flag used to ensure that other processes wait until the process lock completes its action. The following is the standard LCD lock format:

```
!action!processID!time!machine:id.
```

These are examples of general LCD lock flags:

- **`!action!02833!701236710!<servername>:Restore_hierarchy`**. The creation of a file system hierarchy produces a flag in this format in the status display. The filesys designation can be `appdp` for applications with disk partition hierarchies or `appfs` for applications with file system hierarchies.
- Other typical flags include `!nofailover!machine` and `shutdown_switchover`. The `!nofailover!machine` flag is an internal, transient flag created and deleted by LifeKeeper which controls aspects of server failover. The `shutdown_switchover` flag indicates that the shutdown strategy for this server has been set to switchover such that a shutdown of the

server will cause a switchover to occur. See LCDI Flags for more detailed information on the possible flags.

flg_create

```
flg_create [-d destsys] -f flag
```

The flag *flag* is created on system *destsys*.

Note: This only modifies the ”shared memory” segment of the LifeKeeper configuration database.

The LifeKeeper **lcdsync** command should be run after this command to ensure that the shared memory changes are reflected into the permanent storage onto a disk file.

flg_list

```
flg_list [-d destsys]
```

flg_list prints to standard output a short listing, one flag per line, of all of the flags currently defined on this system (unless *destsys* is specified). The listing is not in any particular order.

flg_remove

```
flg_remove [-d destsys] -f flag
```

The flag *flag* is removed on system *destsys*.

Note: This only modifies the shared memory segment of the LifeKeeper configuration database.

The LifeKeeper **lcdsync** command should be run after this command to ensure that the shared memory changes are reflected into the permanent storage onto a disk file.

flg_test

```
flg_test [-d destsys] -f flag
```

A check is made to see if the flag *flag* exists on system *destsys*. Returns 0 or 7.

LCDI Flags

Synopsis

flg_create [-d destsys] -f flag
flg_remove[-d destsys] -f flag
flg_test[-d destsys] -f flag
flg_list[-d destsys]

Description

LifeKeeper provides a facility to dynamically set flags to perform various tasks. The following special purpose flags can exist.

!nofailover!uname

If this flag exists, failover is inhibited for resources on the system with name, uname, that have defined the system with the flag on it as their backup system. Note: This is a temporary flag that will be removed automatically when LifeKeeper detects that system uname is ALIVE.

!action!procid!timestamp!uname:identifier

This is an example of an ”admin lock flag” [see getlocks]. These flags are used for actions that require that no other action be performed at the same time on any of the systems in a LifeKeeper configuration. For example, you may not create a hierarchy on one system while creating a hierarchy on another. The ”admin lock flags” are used to ensure that one of these ”global” operations is not performed until the one currently running completes.

The identifier field of the ”admin lock flag” identifies the kind of action being performed. The system that the process that requested the”admin lock flag” was running on is specified by uname. The flag was created at timestamp number of seconds after Jan 1, 1970, by a process with a process ID of procid that called getlocks [see getlocks].

An example of such a flag is as follows:

!action!01525!701120147!cindy:Create_Hierarchy

This flag indicates that the action Create_Hierarchy is in progress, indicating a hierarchy is being created. The process with process ID1525 requested the ”admin lock flag,” at the time 701120147 on system cindy.

!restore

This flag is set by LifeKeeper when the prerestore scripts [see LCD] are run. It indicates that the postrestore scripts should be run. Normally, this is a transitory condition that LifeKeeper automatically fixes when the postrestore scripts [see LCD] run. The only exception is if the postrestore scripts are being explicitly run using the following command:

```
%LKROOT%\bin\lcdrecover -G restore
```

!restore!uname

When this flag is set it indicates that the postrestore scripts [see LCD] should be run remotely on system uname. When the postrestore scripts are run on this system, LifeKeeper sends a remote request to system uname to run its postrestore scripts. Normally, this is a transitory condition that LifeKeeper automatically fixes. The only exception is if the postrestore scripts are being explicitly run using the `%LKROOT%\bin\lcdrecover -G restore` command.

!remove

This flag is set by LifeKeeper when the preremove scripts [see LCD] are run. It indicates that the postremove scripts should be run at a later time. Normally, this is a transitory condition that LifeKeeper automatically fixes when the postremove scripts [see LCD] are run at a later time. The only exception is if the postremove scripts are being explicitly run using the following command:

```
%LKROOT%\bin\lcdrecover -G remove
```

!remove!uname

When this flag is set it indicates that the postremove scripts [see LCD] should be run remotely on system uname. When the postremove scripts are run on this system, a remote request is sent to system uname to run its postremove scripts. Normally, this is a transitory condition that LifeKeeper automatically fixes. The only exception is if the postremove scripts are being explicitly run using the `%LKROOT%\bin\lcdrecover -G remove` command.

!delete

This flag is set by LifeKeeper when the predelete scripts [see LCD] are run. It indicates that the postdelete scripts should be run at a later time. Normally, this is a transitory condition that LifeKeeper automatically fixes when the postdelete scripts [see LCD(1M)] are run. The only exception to this is if the postdelete scripts are being explicitly run by using the following command:

```
%LKROOT%\bin\lcdrecover -G delete
```

!delete!uname

When this flag is set it indicates that the postdelete scripts [see LCD] should be run remotely on system uname. When the postdelete scripts are run on this system, a remote request is sent to system uname to run its postdelete scripts. Normally, this is a transitory condition that LifeKeeper automatically fixes. The only exception is if the postdelete scripts are being explicitly run using the following command:

```
%LKROOT%\bin\lcdrecover -G delete
```

lk_chg_value

NAME

lk_chg_value.ksh — changes specified values in local LifeKeeper configuration database files

SYNOPSIS

```
lk_chg_value.ksh {-o old_value -n new_value | -f filename} [-vFIMT]
```

DESCRIPTION

This command is to be used to modify arbitrary values in local LifeKeeper configuration database files (e.g. LifeKeeper uname, communication path addresses, resource tag names, etc.). **lk_chg_**

value.ksh needs to be run locally with the Administrator login on each machine within a LifeKeeper configuration while LifeKeeper is not running. Also, you must use the LifeKeeper provided shell (sh.exe) to invoke the script as shown above. This command does not modify the system's uname or network interfaces. If the LifeKeeper uname or communication path addresses are to be modified, the system uname and network interfaces must be modified prior to the execution of this command using system utilities. In order for LifeKeeper to be properly updated, this command must be run on every system in the cluster.

The values to be modified may be specified on the command line using the *-o* and *-n* options or in a file using the *-f* option. The syntax of a file containing substitutions is *old_value=new_value*, with one substitution per line (lines not in this form are ignored).

To see the changes **lk_chg_value.ksh** will make without modifying any LifeKeeper files, use the *-M* option. To see the files **lk_chg_value.ksh** is examining, use *-v*. To not modify tag names, use the *-T* option. To not modify resource ids, use the *-I* option.

Because a resource id may contain structured information, **lk_chg_value.ksh** does not allow substitutions that completely replace the id field. To override this behavior, use the *-F* option.

EXAMPLES

Systems A, B, and C have been configured in a LifeKeeper configuration. Systems A and B manage a database resource and system A also manages a communication resource with System C. To modify the uname and comm path address of System A with comm path old address, the following must be performed:

1. Stop LifeKeeper by executing the lkstop command on each affected system. However, if the resources being managed are to stay available to the user community, execute lkstop -f.
2. Change System A's uname to X and change the network address to new_address. Create a substitutions file, /tmp/lksubs, containing the substitution pairs:

   A=X old_address=new_address

   As Administrator, login to System A and execute the following:

   `set LKROOT=<LKROOT>` (i.e. set LKROOT=C:\LK)

   ```
   <LKROOT>\bin\sh.exe  lk_chg_value.ksh  -vf /tmp/lksubs
   ```

   This changes all local occurrences of A and old_address found within the LifeKeeper core and recovery kits on System A, which is now identified by System X, to refer to X and new_address, respectively.
3. Copy the substitutions file from System A to Systems B and C. As Administrator, login to Systems B and C and execute the following:

   `set LKROOT=<LKROOT>` (i.e. set *LKROOT=C:\LK*)

   ```
   <LKROOT>\bin\sh.exe  lk_chg_value.ksh  -vf /tmp/lksubs
   ```

This changes all occurrences of A and old_address found within the LifeKeeper configuration database on Systems B and C to refer to X and new_address, respectively.

EXIT CODES

0 Execution of command completed successfully.

1 Interrupt occurred... files restored.

2 Invalid arguments passed to command.

3 LifeKeeper processes are still running.

4 Command needs to be executed by an Administrator login.

5 ID field change attempted. Resource ID cannot be changed without using -I option.

6 LKROOT environment variable not set.

7 No matches were found.

NOTES

The **lk_chg_value.ksh** utility is located in the *<LKROOT>\bin* folder.

The **lk_chg_value.ksh** utility is case sensitive.

As shown above you must use the LifeKeeper provided shell (sh.exe) to invoke the **lk_chg_value.ksh** script.

<LKROOT> refers to the LifeKeeper home directory. The default home directory is `C:\LK`, but this can be modified during LifeKeeper installation.

FILES

```
<LKROOT>\bin\lk_chg_value.ksh
```

lk_err

Synopsis

```
lk_err -c Category -n Error number -p Process Name [-d {TO_LOG |TO_
STDERR}] ”Message”
```

Description

This utility is used within recovery scripts to log errors to the Microsoft Event Log. It also prints messages to stderr.

The arguments are:

Category. The following is a list of LifeKeeper message categories and their Event Log classifications:

LK Category	Event Category	Event Type
FRS_MES	General	Information
FRS_WARN	General	Warning
FRS_ERR	General	Error

Error Number. Must be a positive integer.

Process Name. Name of the script calling **lk_err**.

Destination. The destination parameter is optional. By default, events generated by **lk_err** will be directed to both the Windows Event Log (TO_LOG) and to the system console stderr message stream (TO_STDERR). However, with the *-d* option you may direct events specifically to one destination or the other.

Please note, however, that messages directed to the stderr (TO_STDERR) by programs or scripts that are executed by the LifeKeeper core will not display on the system console because the LifeKeeper core runs them as background tasks without interactive properties. Therefore, directing messages to stderr is useful only as a manual script testing and debugging aid.

Message. Message string must be enclosed in ” ”.

perform_action

Synopsis

```
perform_action [-G] [-s] [-b] [-n] -t tag-name -a action-name [- - arg1
arg2 ... argn]
```

Description

The LRACI program **perform_action** performs processes in the following order:

- Finds the resource specified by the tag-name argument of the *-t* option.
- Finds the action script specified by the *action-name* argument of the *-a* option.
- Executes the action script on the *tag-name* resource instance.

The arguments after the -- argument are passed unchanged to the action script(s). These are arguments that the developer of the action may optionally require to use the action.

The **perform_action** program finds the action script by the following algorithm: it first searches in the actions directory for the resource type of the resource instance specified by *tag-name*:

```
%LKROOT%\subsys\appname\resources\restypname\actions\action-
name.ksh
```

and if not there, it checks the actions directory of the application the resource instance belongs to:

```
%LKROOT%\subsys\appname\actions\action-name.ksh
```

The restore and remove actions are special cases for LRACI. The restore action moves an application hierarchy that may be in-service on the remote system to the local system. For restore, LRACI first checks to make certain that the resource instance *tag-name* is not already in the ISP state. If it is not, it recursively checks all of the resource instances that this resource depends upon. The check continues until a resource is found that either depends on no resources, or all of the resources it depends on are already in the ISP state. If the resource was in the ISU state,it is placed in the ISP state.

If the resource was in the OSU or OSF state, LRACI executes the remove script for any affected resources on the remote system. When this completes, LRACI finds the restore action using the above algorithm and runs it. If the script fails, the resource is placed in the OSF state and LRACI stops. If it succeeds, LRACI recursively ”restores” the resources up the tree, until the resource specified by *tag-name* is restored. Then the LRACI recursively checks and ”restores” the parent resource instances in a similar fashion until all related root resource instances are restored. In each case, LRACI uses the above algorithm to find the correct restore script using the resource application and resource type of the resource currently being scanned by LRACI, not the resource application and resource type of the *tag-name* resource.

For the remove action, the resources are moved recursively in the opposite direction. LRACI calls the remove script of all resources starting at the root resources that depend directly or indirectly on the *tag-name* resource down to, and including, the tag-name resource if any of those resources are in the ISP or ISU state. Resources not in the ISP or ISU state are ignored. If one of the remove scripts fails, LRACI places the failing resource into the OSF state and stops. In each case, LRACI uses the algorithm to find the correct remove script using the resource application and resource type of the resource currently being scanned by LRACI, not the resource application and resource type of the *tag-name* resource.

The remove and restore actions automatically have the *-t tag-name* and *-i ident-field* arguments added to the argument list that corresponds to the resource instance being acted upon.

The following sections discuss the arguments accepted by **perform_action**.

-G	This option is only used if action-name is remove, restore, or delete. If this option is not specified, LRACI performs the preglobal and postglobal scripts before and after the actions are performed [see lcdrecover in LCD]. If the option is specified, LRACI does not run the preglobal and postglobal scripts. This option is useful if you need to run **perform_action** more than once, but you only want to run the preglobal and postglobal scripts once. It is also useful if you need to run perform_action while creating a resource hierarchy. The preglobal and postglobal scripts should not be run by **perform_action** during hierarchy create because the hierarchy creation scripts should be set up to obtain the ”admin lock flags” [see LCDI_flag] and postrestore also requires the ”admin lock flags” which would lead to contention problems.
-s	The default behavior for the restore action is to bring all objects above and below the specified tag into service, and the default behavior for the remove action is to bring all objects above the specified tag out of service. The *-s* option limits the scope of the operation to just the specified flag.
-b	The default behavior for the restore action is to bring all objects above and below the specified tag into service. The *-b* option changes this behavior to just objects below the specified tag. This option has no effect on the remove action.
-n	This option is only used if the *action-name* is remove or restore. If this option is specified, the resource reserves are not to be checked and the actions are performed whether the resources are reserved or not. **WARNING: EXTREME CAUTION SHOULD BE TAKEN WHEN USING THIS OPTION!** If this option is not specified, before any remove or restore scripts are executed, LRACI checks to see if any of the resources on which any of the actions will be run are currently reserved by another process. A resource can be reserved while the following operations are being performed on them: a resource ”remove from service” is in progress, a ”resource restore to service” is in progress, or a resource ”recovery” is in progress. If any resource is so reserved, LRACI waits a specified period of time for the process to remove the reserve on the resource. If that period expires, LRACI removes the reserve. In either case, LRACI reserves all of the resources, then follows the specified algorithm to perform the action(s).
-t tag-name	This is the last resource instance the action will be performed on.
-a action-name	This is the resource action that will be performed.
-- arg1 arg2 ... argn	Argument(s) the resource action developer can optionally define to be passed to the action script. When executing **perform_action** within a getlocks protected region and the *-G* option is not used, set *arg1* to *-m* to avoid executing a second instance of getlocks, which would cause the

	operation to hang.

Example

The following is an example of calling an action:

```
perform_action -t SCSI-USR-DISK -a reset-heads -- -h 7
```

The LRACI program **perform_action** would find the action corresponding to reset-heads and execute it with the arguments:

```
reset-heads -t SCSI-USR-DISK -h 7
```

Exit Codes

The following exit codes could be returned by LRACI:

0	The operation has succeeded.
1	A system call or library call has internally returned failure.
2	A user-specified syntax error occurred.
3	LifeKeeper internal error.
4	A request to perform an operation on an object that already exists.
5	An argument specified is illegal.
6	Index out-of-range.
7	A request has been made on an object that does not exist.
8	A request was made to delete a resource instance on which another non-deleted resource instance depends.
9	An attempt to communicate with another system failed.

sendevent

Synopsis

```
%LKROOT%\bin\sendevent -C class-of-event -E event -m monitor-name -
nname-of-obj-inst [-s severity]
```

Description

The event notification facility consists of two parts: an event notification mechanism (`%LKROOT%\bin\sendevent`) and an application registration environment. Applications wishing to use the event facility should ”register” to get notification of specific events or alarms (or all occurrences of event/alarms).

The **sendevent** command is a program invoked by a daemon monitor process when the monitor has detected an event (failure or recovery) in the objects that it is monitoring. This command is not intended to be run directly at the shell level by a regular user or by a system administrator (only by a daemon process or another command).

The **sendevent** command is used to notify a ”registered” application of the occurrence of an event. For example, an application may want to be notified of an impending system shutdown so it can appropriately save files and data; or, in a client-server environment, the application may need to reconfigure itself to an alternate service provider. The application is responsible for providing the appropriate command support to handle the event.

The **sendevent** command passes all of its options to the event-response commands of the application.

An application registers to receive notification of events or alarms by installing its event-response commands in a specific registration directory, *%LKROOT%\events*. This should be done at application installation time. The events under *%LKROOT%\events* are further categorized in classes of events. Create separate subdirectories, genclass (for general events) and allclass to be used by applications to register to be notified upon occurrence of any event.

Note: If an event occurs which causes an application to place application-response commands in both the specific event location and the all location, both scripts run.

Each class directory contains a subdirectory for each event within that class. Each add-on package that monitors events and uses this event notification mechanism documents events it monitors and supports.

It is the responsibility of the application object monitor package to maintain a file called ACTIVE in the events subdirectories. If the ACTIVE file exists, it is a signal to the applications that a monitor is currently actively running and monitoring its objects for that specific event. If the package containing the monitor program is removed, the files named ACTIVE for the affected monitored events are removed too (by the package remove script) to indicate to applications that the event is no longer being monitored. The removal of the package should not remove event-response commands or event directories even if they are empty.

For those applications that may depend upon standard commands from another application, the application registration environment provides other application-specific directories, *%LKROOT%\subsys\application-name\actions*, for applications to place ”sharable” action commands. For example, application X may depend upon application Y being up and running after an event recovery. If this is not the case, application X may invoke the start command for application Y from the *LKROOT%\subsys\Y\actions\start* directory. Interdependencies between applications must be resolved and specified by the application developers.

The *-C* (class of event), *-E* (event), *-m* (monitor name), and *-n* (name of object instance) options are required. If the *-s* (severity) option is not specified, sendevent will default to a severity of MAJOR alarm.

Upon invocation of the **sendevent** command by a monitoring process, **sendevent** determines which event class and event has occurred based upon the arguments to the *-C* and *-E* options. The **sendevent** command executes in the background until it finishes processing all the event-response commands (if any) placed in the registration directory corresponding to that class/event pair and all of the commands registered in the all directory.

The following options are supported:

-C class-of-event

Events are grouped together into classes. This required option indicates which class the event belongs to.

-E event

This required option indicates which event in a class is being triggered.

-m monitor-name

Each application object monitor that can send alarms/events is identified by a name in the form:

```
OM-product-name:OM-component-name
```

OM-product-name is an ASCII string, of up to eight characters. It is an abbreviated identifier specifying the product that monitors the objects that cause alarms or events. OM-component-name is an ASCII string, of up to 16 characters. It is defined by the object monitor to identify the component of the object monitor that detected the alarm or event.

The monitor names are used to distinguish between different products that may be used to monitor the same object.

-n name-of-obj-inst

This option is used to name a specific instance of an application object. It is an ASCII string with a maximum length of 64 characters. For example, D: may be the name of a volume application object, whereas 1234 could be used to identify a specific process object.

-s severity

Each alarm or event must specify the severity of the problem it is reporting. If this option is not specified, **sendevent** internally adds the default severity for MAJOR alarm. Severity is an ASCII represented integer interpreted as follows:

0	CLEARED alarm specified by ”id-of-alarm/event” has been recovered
1	INFORMATIONAL alarm (INFO message or cmn_err() NOTICE message)
2	WARNING alarm (WARNING message)
3	MINOR alarm (MINOR message)
4	MAJOR alarm (MAJOR or ERROR message) (default)
5	CRITICAL alarm (CRITICAL message or cmn_err() PANIC or HALT message)

Output

The output this command generates occurs in one of two conditions:

- Error messages are printed to standard error and a nonzero exit code is returned.
- The identifier for the alarm\event called id-of-alarm/event is printed to standard output at each call to sendevent.

Exit Codes

The following exit codes are returned by **sendevent**:

0	The **sendevent** command has completed successfully without errors.
1	Syntax error in the argument list.
2	No class corresponding to the string passed with the *-C* option exists in the `%LKROOT%\events` directory.
3	No event corresponding to the string passed with the *-E* option exists in the `%LKROOT%\events\<class>` directory.
4	The *-A* option is internally generated and may not be specified directly.
5	The *-i* option must be specified if the *-s 0* (severity CLEARED) option is used.

volume

Synopsis

```
volume [ -d | -D ]  [-l | -u | -p | -U  volume_letter ]
```

Description

This command is used to lock and unlock volumes on the Windows server. It may also be used to register with the LifeKeeper Service. When used in this fashion, it determines which volumes should be protected (locked) by LifeKeeper at startup. The lock provides LifeKeeper with exclusive access to the volume and will not allow any other process to access the volume.

LifeKeeper must be running in order for this command to succeed. The command interfaces with the LifeKeeper Service to provide the locking mechanism.

The following options are available where *volume_letter* is the drive letter to be locked\unlocked or protected\unprotected (i.e. C to Z).

-d	Display the currently locked volumes.
-D	Display the volumes that are registered with LifeKeeper. This would display volumes that have been added with the -p option. Generally, -D displays a different list than the one shown by the -d option.
-l	Lock the volume for exclusive access. The lock will fail if a remote user has opened the volume or a local application has opened the volume for a write operation.
-u	Unlock the volume from exclusive access.
-p	Register the volume with LifeKeeper, so that on subsequent reboots or restarts of LifeKeeper, the volume is automatically locked.
-U	Unregister the volume with LifeKeeper so that it is not automatically locked on LifeKeeper startup

Example

The following illustrates how the volume command should be used:

```
#

# Register drive volume e: to be locked by LifeKeeper

#

ret=`volume -p E`

if [ $ret -gt 0 ]

then

      #Report error that it wasn’t protected

fi

#

# Lock volume e: for exclusive access

#

ret=`volume -l E`

if [ $ret -gt 0 ]

then

      #Report error that it wasn’t locked

fi
```

Exit Codes

The following exit codes could be returned by this command:

0	The operation has succeeded.
greater than 0	The operation has failed. An error message is printed to standard error.

Setting Browser Security Parameters

In order to run the GUI web client, you must set your browser security settings to low. For Internet Explorer and Netscape, follow the procedures below.

WARNING: Be careful of other sites you visit with low security settings.

Internet Explorer

The most secure method for using Internet Explorer is to add the LifeKeeper server to the Trusted Sites zone as follows:

1. From the **Tools** menu, click **Internet Options**.
2. Click the **Security** tab.
3. Select **Trusted Sites** zone and click **Custom Level**.
4. Under **Reset custom settings**, select **Medium/Low**, then click **Reset**.
5. Click **Sites**.
6. Enter the **server name** and **port number** for the LifeKeeper server(s) to which you wish to connect (for instance: http://server1:81).

An alternative, but possibly less secure method is to do the following:

1. From the **Tools** menu, click **Internet Options**.
2. Select either **Internet** or **Local Intranet** (depending upon whether your remote system and the LifeKeeper cluster are on the same intranet).
3. Adjust the **Security Level** bar to **Medium** (for Internet) or **Medium-low** (for Local Intranet). These are the default settings for each zone.
4. Click **OK**.

Netscape Navigator and Netscape Communicator

1. From the **Edit** menu, select **Preferences**.
2. In the **Preferences** dialog box, double-click the **Advanced** Category.
3. Select the ”**Enable Java**” and ”**Enable Java Scrip**t” options.
4. Click OK.

IP Local Recovery

When IP Local Recovery is enabled and the IP resource fails its deepcheck (a periodic extensive check of the IP resource), then LifeKeeper will do the following:

- First, LifeKeeper will attempt to bring the IP address back in service on the current network interface.
- If that fails, LifeKeeper will check the resource instance to determine if there is a backup interface available. If so, it will then attempt to move the IP address to the backup interface.
- If all local recovery attempts fail, LifeKeeper will perform a failover of the IP address and all dependent resources to a backup server.

Even if you do not have a backup adapter, you can enable Local Recovery so that LifeKeeper will retry the primary network interface before initiating failover to a backup server.

Overview of LifeKeeper Event Forwarding via SNMP

The Simple Network Management Protocol (SNMP) defines a device-independent framework for managing networks. Devices on the network are described by MIB (Management Information Base) variables that are supplied by the vendor of the device. A SNMP agent runs on each node of the network, and interacts with a Network Manager node. The Network Manager can query the agent to get or set the values of its MIB variables, there by monitoring or controlling the agent's node. The agent can also asynchronously generate messages called traps to notify the manager of exceptional events. There are a number of applications available for monitoring and managing networks using the Simple Network Management Protocol (SNMP).

LifeKeeper has an event notification mechanism for registering applications that wish to be notified of specific events or alarms (see the sendevent man page). LifeKeeper can be easily enabled to send SNMP trap notification of key LifeKeeper events to a third party network management console wishing to monitor LifeKeeper activity. LifeKeeper installs a MIB file under `%LKROOT%\include\LifeKeeper-MIB.txt` which describes LifeKeeper trap definitions.

The remote management console receiving SNMP traps must first be configured through the administration software of that system; LifeKeeper provides no external SNMP configuration. The remote management server is typically located outside of the LifeKeeper cluster (i.e., it is not a LifeKeeper node).

LifeKeeper Events Table

The following table contains the list of LifeKeeper events and associated trap numbers. The entire Object ID (OID) consists of a prefix followed by a specific trap number in the following format:

prefix.0.specific trap number

The prefix is **.1.3.6.1.4.1.7359**, which expands to **iso.org.dod.internet.private.enterprises.7359** in the MIB tree. (7359 is SIOS's enterprise number, followed by 1 for LifeKeeper.) For example, the LifeKeeper Startup Complete event generates the OID: **.1.3.6.1.4.1.7359.1.0.100**

LifeKeeper Event/Description	Trap #	Object ID
LifeKeeper Startup Complete Sent from a node when LifeKeeper is started on that node	100	.1.3.6.1.4.1.7359.1.0.100
LifeKeeper Shutdown Initiated Sent from a node beginning LifeKeeper shutdown	101	.1.3.6.1.4.1.7359.1.0.101
LifeKeeper Shutdown Complete Sent from a node completing LifeKeeper shutdown	102	.1.3.6.1.4.1.7359.1.0.102
LifeKeeper Manual Switchover Initiated on Server Sent from the node from which a manual switchover was requested	110	.1.3.6.1.4.1.7359.1.0.110
LifeKeeper Manual Switchover Complete – recovered list Sent from the node where the manual switchover was completed	111	.1.3.6.1.4.1.7359.1.0.111
LifeKeeper Manual Switchover Complete – failed list Sent from the node where the manual switchover was completed	112	.1.3.6.1.4.1.7359.1.0.112
LifeKeeper Node Failure Detected Sent from each node within the cluster when a node in that cluster fails	120	.1.3.6.1.4.1.7359.1.0.120
LifeKeeper Node Recovery Complete – recovered list Sent from each node within the cluster that has recovered resources from the failed node	121	.1.3.6.1.4.1.7359.1.0.121

LifeKeeper Event/Description	Trap #	Object ID
LifeKeeper Node Recovery Complete – failed list Sent from each node within the cluster that has failed to recover resources from the failed node	122	.1.3.6.1.4.1.7359.1.0.122
LifeKeeper Resource Recovery Initiated Sent from a node recovering a resource; a 131 or 132 trap always follows to indicate whether the recovery was completed or failed.	130	.1.3.6.1.4.1.7359.1.0.130
LifeKeeper Resource Recovery Failed Sent from the node in trap 130 when the resource being recovered fails to come into service	131*	.1.3.6.1.4.1.7359.1.0.131
LifeKeeper Resource Recovery Complete Sent from the node in trap 130 when the recovery of the resource is completed	132	.1.3.6.1.4.1.7359.1.0.132
Mirror State Change Sent from the node, who is the source of the mirror, when the mirror state changes. Displays the volume letter, mirror state and IP address of the target node. Valid Mirror States: -1: Invalid State 0: No Mirror 1: Mirroring 2: Mirror is resyncing 3: Mirror is broken 4: Mirror is paused 5: Resync is pending	150	.1.3.6.1.4.1.7359.1.0.150
LifeKeeper replicated volume Split Brain detected Sent from the node where LifeKeeper has detected mirror is Source on both sides. Displays volume letter and IP address of the target node.	160	.1.3.6.1.4.1.7359.1.0.160
The following variables are used to "carry" additional information in the trap PDU:		
Trap message	all	.1.3.6.1.4.1.7359.1.1
Resource Tag	130	.1.3.6.1.4.1.7359.1.2
Resource Tag	131	.1.3.6.1.4.1.7359.1.2
Resource Tag	132	.1.3.6.1.4.1.7359.1.2
List of recovered resources	111	.1.3.6.1.4.1.7359.1.3

LifeKeeper Event/Description	Trap #	Object ID
List of recovered resources	121	.1.3.6.1.4.1.7359.1.3
List of failed resources	112	.1.3.6.1.4.1.7359.1.4
List of failed resources	122	.1.3.6.1.4.1.7359.1.4

* This trap may appear multiple times if recovery fails on multiple backup servers.

Chapter 5: User Guide

The User Guide is a complete, searchable resource containing detailed information on the many tasks that can be performed within the LifeKeeper GUI. Click User Guide to access this documentation.

The tasks that can be performed through the GUI can be grouped into three areas:

Common Tasks - These are basic tasks that can be performed by any user such as connecting to a cluster, viewing server or resource properties, viewing log files and changing GUI settings.

Operator Tasks - These are more advanced tasks that require Operator permission such as bringing resources in and out of service.

Administrator Tasks - These are tasks that require Administrator permission. They include server-level tasks such as editing server properties, creating resources, creating or deleting comm paths and resource-level tasks such as editing, extending, or deleting resources.

The table below lists the default tasks that are available for each user permission. Additional tasks may be available for specific resource types, and these will be described in the associated resource kit documentation.

Task	Permission		
	Guest	**Operator**	**Administrator**
View servers and resources	X	X	X
Connect to and disconnect from servers	X	X	X
View server properties and logs	X	X	X
Modify server properties			X
Create resource hierarchies			X
Create and delete comm paths			X
View resource properties	X	X	X
Modify resource properties			X
Take resources into and out of service		X	X
Extend and unextend resource hierarchies			X
Create and delete resource dependencies			X
Delete resource hierarchies			X

LifeKeeper GUI

LifeKeeper Graphical User Interface

The GUI components should have already been installed as part of the LifeKeeper Core installation.

The LifeKeeper GUI uses Java technology to provide a graphical user interface to LifeKeeper and its configuration data. Since the LifeKeeper GUI is a client/server application, a user will run the client component to monitor or administer LifeKeeper. The client and the server components may or may not be run on the same system.

GUI Overview

The GUI allows users working on any machine to administer, operate or monitor servers and resources in any cluster as long as they have the required group memberships on the cluster machines. (For details, see Configuring GUI Users.) The GUI Server and Client components are described below.

GUI Server

The GUI server is initialized on each LifeKeeper server at system startup. It communicates with GUI clients using Hypertext Transfer Protocol (HTTP) and Remote Method Invocation (RMI).

GUI Client

The GUI client can be run either as a web client on any Java-enabled system or as an application on a LifeKeeper server.

The client includes the following components:

- The status table on the upper left displays the high level status of connected servers and their resources.
- The properties panel on the upper right displays detailed information about the most recently selected status table object.
- The output panel on the bottom displays command output.
- The message bar at the very bottom of the window displays processing status messages.
- The context (in the properties panel) and global toolbars provide fast access to frequently-used tasks.
- The context (popup) and global menus provide access to all tasks.

Starting GUI Clients

Starting the Web Client

To run the web client on a LifeKeeper server, click **Start** then point to **All Programs**, then point to **SteelEye->LifeKeeper->LifeKeeper**. This will invoke a web browser and connect to the local GUI server using *http://localhost:81.*

On systems outside the LifeKeeper cluster, open a web browser and go to the URL http://<server name>:81 where <server name> is the name of a LifeKeeper server. This will load the web client from the GUI server on that machine.

After the web client has finished loading, you should see the Cluster Connect Dialog which allows you to connect the web client to any GUI server.

Note: When you run the web client, if your system does not have the required Java Plug-in, you will be automatically taken to the web site for downloading the plug-in. You must also set your browser security parameters to enable Java.

If you have done this and the client still is not loading, see Web Client Troubleshooting.

Starting the Application Client

Users with administrator privileges on a LifeKeeper server can run the application client from that server. Click **Start**, then point to **All Programs**, then **SteelEye->LifeKeeper->LifeKeeper (Admin Only)**.

If you have done this and the client still is not loading, see Network-Related Troubleshooting.

Exiting GUI Clients

Select **Exit** from the File Menu to disconnect from all servers and close the client.

Status Table

The status table provides a visual representation of the status of connected servers and their resources. It shows

- the state of each server in the top row,
- the global (cross-server) state and the parent-child relationships of each resource in the left-most column, and
- the state of each resource on each server in the remaining cells.

The states of the servers and resources are shown using graphics, text and color. An empty table cell under a server indicates that a particular resource has not been defined on that server.

If you select a server or a resource instance in the status table, detailed state information and a context-sensitive toolbar for that item are shown in the properties panel. You can also pop up the appropriate server context menu or resource context menu for any item by right-clicking on that cell.

The status table is split into two sections. The relative sizes of the left and right sections can be modified by moving the divider between them. The status table can also be collapsed to show only the highest level items in the hierarchy trees. Collapsing or expanding resource items in the tree causes the hierarchies listed in the table to also expand and collapse.

Properties Panel

The properties panel displays the properties of the server or resource that is selected in the status table. The properties panel has the same functionality as the server properties dialog or the resource properties dialog plus a context-sensitive toolbar to provide fast access to commonly used commands. The caption at the top of this panel is ***server_name*** if a server is selected, or ***server_name: resource_name*** if a resource is selected.

The context-sensitive toolbars displayed in the properties panel are the server context toolbar and the resource context toolbar. Server or resource toolbars may also be customized. For more information on customized toolbars, see the corresponding application recovery kit documentation.

The buttons at the bottom of the properties panel function as follows.

- The **Apply** button applies any changes that have been made to editable properties on the panel. This button is only enabled if you have changed an editable property.
- The **Refresh** button queries the server for the current values of all properties clearing any changes that you may have made. This button is always enabled.

You increase or decrease the size of the properties panel by sliding the separator at the left of the panel to the left or right. If you want to open or close this panel, use the Properties Panel checkbox on the View Menu.

Output Panel

The output panel collects output from commands issued by the GUI client. When a command begins to run, a time stamped label is added to the output panel and all of the output from that command is added under this label. If you are running multiple commands at the same time (typically on different servers), the output from each command is sent to the corresponding section making it easy to see the results of each.

You increase or decrease the size of the output panel by sliding the separator at the top of the panel up or down. If you want to open or close this panel, use the Output Panel checkbox on the View Menu. When the output panel is closed, the dialog that initiates each command will stay up, the output will be displayed on that dialog until you dismiss it, and you will not be able to review the output from any command after you have closed that dialog. After the output panel is reopened, the GUI will return to its default behavior.

Message Bar

The message bar appears beneath the status window. It is used for displaying messages in a single text line. Messages such as "`Connecting to Server X`" or "`Failure to connect to Server X`" might be displayed.

- To hide the message bar, clear the **Message Bar checkbox** in the View Menu.
- To display the message bar, select the **Message Bar checkbox** in the **View Menu**.
- To see a history of messages displayed in the message bar, see Viewing Message History.

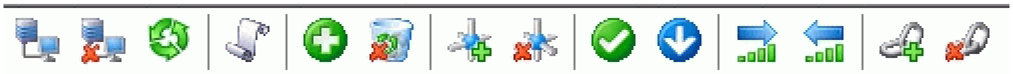
Toolbars

SteelEye LifeKeeper for Windows Toolbars

- Global Toolbar
- Resource Context Toolbar
- Server Context Toolbar

Global Toolbar

This toolbar is a combination of the default server context toolbar and resource context toolbar which are displayed on the properties panel, except that you must select a server and possibly a resource when you invoke actions from this toolbar.



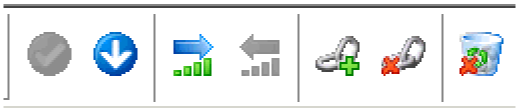
	Connect. Connect to a cluster.
	Disconnect. Disconnect from a cluster.
↻	Refresh. Refresh GUI.
	View Logs. View log messages.
⊕	Create Resource Hierarchy. Create a resource hierarchy.
	Delete Resource Hierarchy. Remove a resource hierarchy from all servers.
	Create Comm Path. Create a communication path between servers.
	Delete Comm Path. Remove communication paths from a server.
✓	In Service. Bring a resource hierarchy into service.
⬇	Out of Service. Take a resource hierarchy out of service.
➡	Extend Resource Hierarchy. Copy a resource hierarchy to another server for failover support.

	Unextend Resource Hierarchy. Remove an extended resource hierarchy from a single server.
	Add Dependency. Create a parent/child relationship between two resources.
	Remove Dependency. Remove a parent/child relationship between two resources.

Resource Context Toolbar

The resource context toolbar is displayed in the properties panel when you select a server-specific resource instance in the status table. The default toolbar is described here but this toolbar might be customized for specific resource types in which case the custom toolbar will be described in the appropriate resource kit documentation.

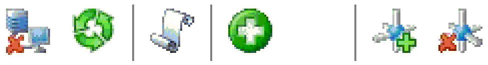
The actions are invoked for the server and the resource that you select. Actions that are not available for selection for a resource will be grayed out.



✓	In Service. Bring a resource hierarchy into service.
⬇	Out of Service. Take a resource hierarchy out of service.
→	Extend Resource Hierarchy. Copy a resource hierarchy to another server for failover support.
←	Unextend Resource Hierarchy. Remove an extended resource hierarchy from a single server.
	Add Dependency. Create a parent/child relationship between two resources.
	Remove Dependency. Remove a parent/child relationship between two resources.
	Delete Resource Hierarchy. Remove a resource hierarchy from all servers.

Server Context Toolbar

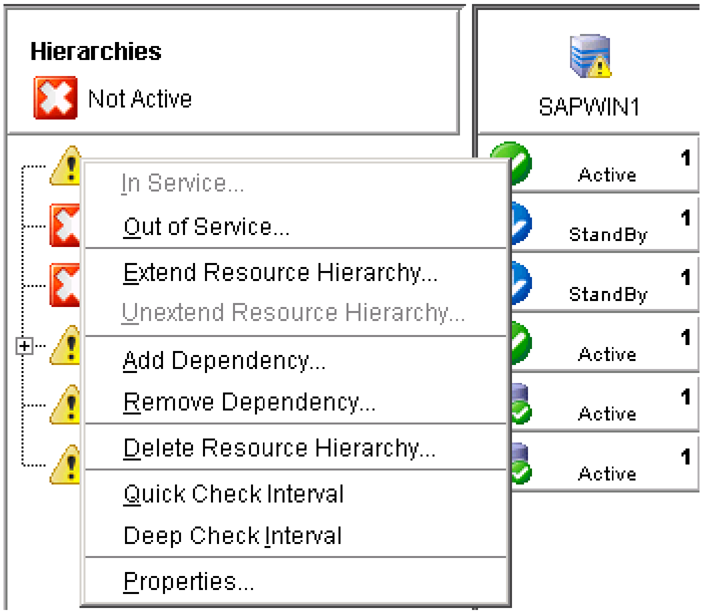
The server context toolbar is displayed in the properties panel when you select a server in the status table. The actions are invoked for the server that you select.



	Disconnect. Disconnect from a cluster.
	Refresh. Refresh GUI.
	View Logs. View log messages.
	Create Resource Hierarchy. Create a resource hierarchy.
	Create Comm Path. Create a communication path between servers.
	Delete Comm Path. Remove communication paths from a server.

Menus

Resource Context Menu



The resource context menu appears when you right-click on a global (cluster-wide) resource, as shown above, or a server-specific resource instance, as shown below, in the status table. The default resource context menu is described here, but this menu might be customized for specific resource types in which case the menu will be described in the appropriate resource kit documentation.

The actions are invoked for the resource that you select. If you select a resource instance on a specific server, the action is invoked for that server while if you select a global (cluster-wide) resource, you will need to select the server.

In Service. Bring a resource hierarchy into service.

Out of Service. Take a resource hierarchy out of service.

Extend Resource Hierarchy. Copy a resource hierarchy to another server for failover support.

Unextend Resource Hierarchy. Remove an extended resource hierarchy from a single server.

Add Dependency. Create a parent/child relationship between two resources.

Remove Dependency. Remove a parent/child relationship.

Delete Resource Hierarchy. Remove a resource hierarchy from all servers.

Local Recovery - Select **Yes** to enable Local Recovery for this Resource. Local recovery for a file share means that if the folder becomes inaccessible, LifeKeeper will attempt to re-share the folder.

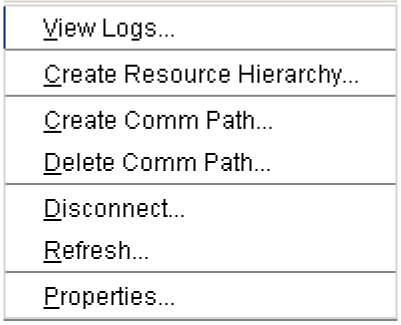
Quick Check Interval - Enter the interval (in minutes) between basic checks of the resource's availability. Different values can be specified for each system. The default value is 3 minutes. The value range is between 0 and 10080. Setting the interval value to 0 will disable the quick check feature.

Deep Check Interval - Enter the interval (in minutes) between extensive checks of the resource's availability. This program utilizes Quickcheck for its Deepcheck implementation. Different values can be specified for each system. The default value is 5 minutes. The valid entry range is between 0 to 10080. Setting the interval value to 0 will disable the Deepcheck feature.

Properties. Display the resource properties dialog.

Server Context Menu

The server context menu appears when you right-click on a server in the status table. The actions are always invoked on the server that you select.



View Logs. View LifeKeeper log messages.

Create Resource Hierarchy. Create a resource hierarchy.

Create Comm Path. Create a communications path between servers.

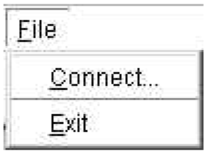
Delete Comm Path. Remove communications paths.

Disconnect. Disconnect from a cluster.

Refresh. Refresh GUI.

Properties. Display the server properties dialog.

File Menu

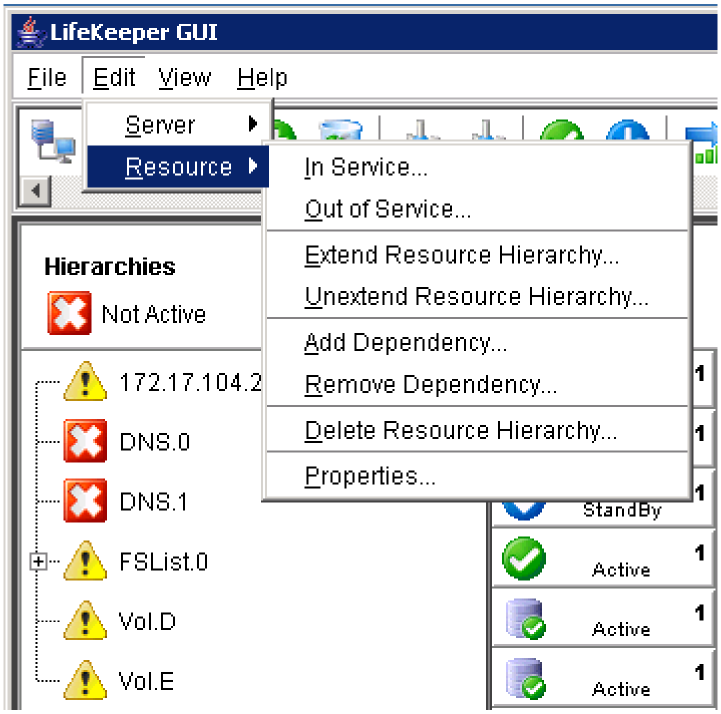


Connect. Connect to a LifeKeeper cluster (requires login authentication on each server).

Exit. Disconnect from all servers and close the GUI window.

Edit Menu - Resource

This submenu of the main menu bar is the same as the default resource context menu except that you must select a resource and server when you invoke actions from this menu. The **Edit > Resource** menu cannot be customized.



In Service. Bring a resource hierarchy into service.

Out of Service. Take a resource hierarchy out of service.

Extend Resource Hierarchy. Copy a resource hierarchy to another server for failover support.

Unextend Resource Hierarchy. Remove an extended resource hierarchy from a single server.

Add Dependency. Create a parent/child relationship between two resources.

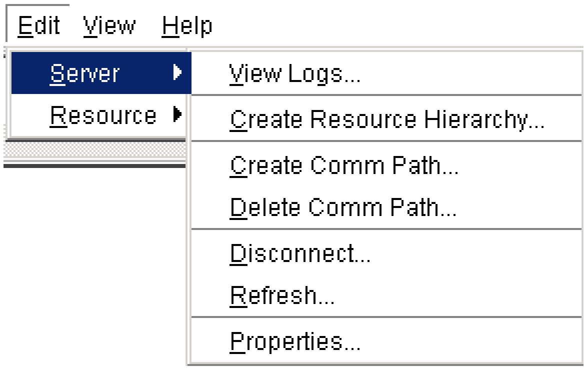
Remove Dependency. Remove a parent/child relationship.

Delete Resource Hierarchy. Remove a resource hierarchy from all servers.

Properties. Display the Resource Properties dialog.

Edit Menu - Server

This submenu of the main menu bar is the same as the default server context menu except that you must select a server when you invoke actions from this menu. The **Edit > Server** menu cannot be customized.



View Logs. View LifeKeeper log messages.

Create Resource Hierarchy. Create a resource hierarchy.

Create Comm Path. Create a communications path between servers.

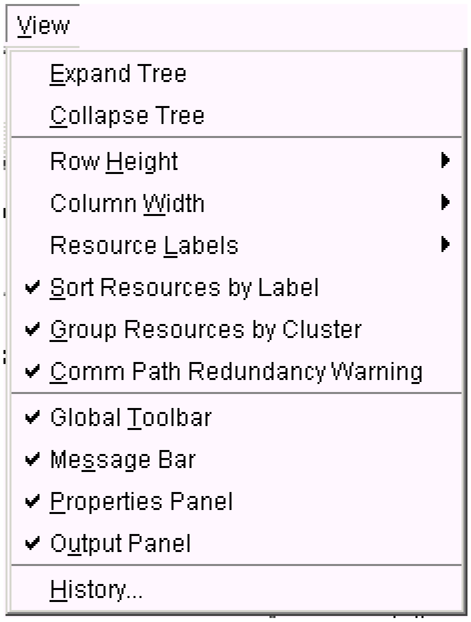
Delete Comm Path. Remove communications paths.

Disconnect. Disconnect from a cluster.

Refresh. Refresh GUI.

Properties. Display the server properties dialog.

View Menu



Expand Tree. Expand the status table to show all resources in all hierarchies.

Collapse Tree. Collapse the status table to show only the top resource in each hierarchy.

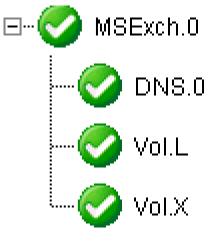
Row Height. Modify the row viewing size of the resources in the resource hierarchy tree and table. Select small, medium or large row height depending upon the number of resources displayed.

Column Width. Modify the column with viewing size of the resources in the resource hierarchy tree and table. Select **fill available space**, large, medium or small depending upon the resource displayed.

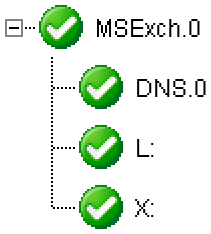
Resource Labels

This option group allows you to specify whether resources are viewed in the resource hierarchy tree by their tag name or ID.

By tag name:



By ID:



Sort Resources by Label will sort resources by resource label only.

Group Resources by Cluster will sort by server cluster and resource label such that resources belonging in the same cluster of servers will be grouped together.

Comm Path Redundancy Warning specifies the representation of comm path status in the server status graphic.

- If selected, the display will show a server warning graphic if the comm paths between a set of servers are not configured with a redundant comm path.
- If not selected, the display will ignore a lack of redundant comm paths between a pair of servers but will still present server warning graphic if there are comm path failures.

Global Toolbar. Display this component if the checkbox is selected.

Message Bar. Display this component if the checkbox is selected.

Properties Panel. Display this component if the checkbox is selected.

Output Panel. Display this component if the checkbox is selected.

History. Display the newest message bar messages in the message history dialog.

Help Menu

About. Displays GUI version information.

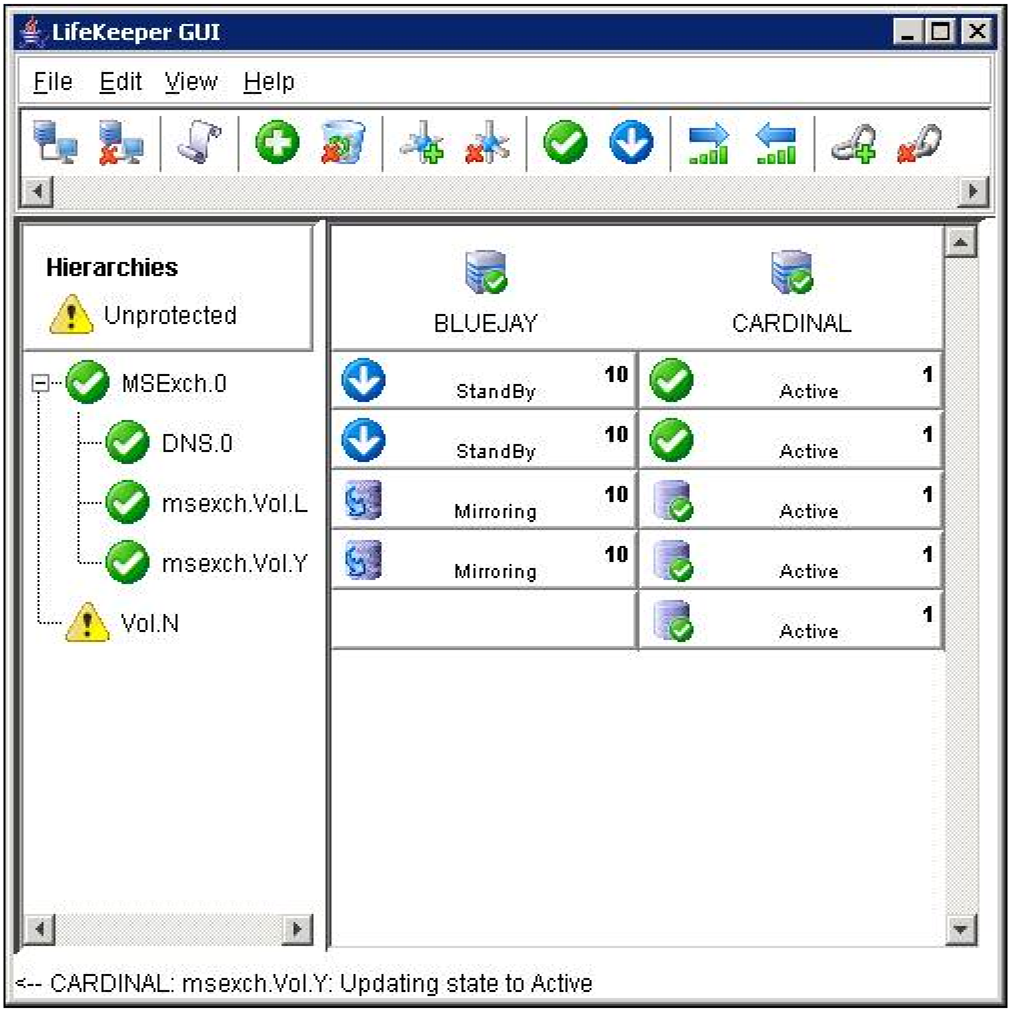
LifeKeeper GUI Server and Client Components

The LifeKeeper GUI server is initialized on each LifeKeeper server at system startup. It communicates with LifeKeeper GUI clients using Hypertext Transfer Protocol (HTTP) and Remote Method Invocation (RMI).

You can connect to the LifeKeeper GUI server with a web client that can be run from any system that can connect to Ports 81 and 82 of all servers in the cluster or with an application client that ships with LifeKeeper and is designed to run on a server in the cluster.

Both LifeKeeper clients include the same graphical components:

- Pop-up server and resource context menus provide access to server- and resource-related actions.
- The menu bar provides access to all LifeKeeper GUI actions.
- The toolbar provides quick access to many LifeKeeper actions.
- The status window displays a graphical representation of the servers connected in the cluster, resource hierarchies and the status of resources and servers.
- The message bar at the bottom of the window displays processing information to the user.



	BLUEJAY		CARDINAL	
MSExch.0	StandBy	10	Active	1
DNS.0	StandBy	10	Active	1
msexch.Vol.L	Mirroring	10	Active	1
msexch.Vol.Y	Mirroring	10	Active	1
Vol.N			Active	1

<-- CARDINAL: msexch.Vol.Y: Updating state to Active

Running the LifeKeeper Web Client

If you wish to administer LifeKeeper from a system outside your cluster, you must use the web client. This is possible for remote systems running any operating system. LifeKeeper for Windows cannot manage both Linux and Windows servers in a single session, but it can manage either Windows or Linux systems no matter what OS it is running on. Whichever type of server OS you first connect to will determine the type of OS you can manage in that session. If you need to simultaneously manage both Linux and Windows servers, you will need to open up two browser windows, one for each.

The remote system's browser must provide JRE 1.4 or later support. Refer to the LifeKeeper Release Notes for information on the supported platforms and browsers for the LifeKeeper web client. The following sections explain steps for configuring the web browser on a remote system.

Follow the procedure below to run the LifeKeeper web client.

1. Open the URL *http://<server name>:81* for the LifeKeeper web page (where <server name> is the name of the LifeKeeper server). The web page contains the LifeKeeper splash screen and applet.
   When you run the web client for the first time, if you are using Internet Explorer or Netscape and your system does not have the required Java plug-in, you will be automatically taken to the appropriate web site for downloading the plug-in.

   Notes:

   - You should close and restart your browser after installing the plug-in and whenever plug-in properties are changed. Thus you will need to enter the LifeKeeper server's URL again as stated above.

     When the web page is opened, the following actions take place:

   - the splash screen is displayed
   - the applet is loaded
   - the Java Virtual Machine is started
   - some server files are downloaded
   - the applet is initialized

   Depending upon your network and system configuration, these actions may take up to 20 seconds. Typically, browsers provide some minimal status as the applet is loading and initializing.

   Note: You may receive a Java Plug-In Security Warning stating "Unable to verify the certificate - code will be treated as unsigned." Click **OK**.

   Next, a Start button should appear in the applet area at the bottom of the splash screen. If the splash screen does not display a Start button or you suspect that the applet failed to load and initialize, refer to the GUI Network-Related Troubleshooting section in this guide.

2. Click **Start**. The LifeKeeper web client appears and the Cluster Connect dialog is automatically displayed. Enter the **Server Name** you wish to connect to followed by the **login** and **password**. Once a Server Name has been entered and connection to the cluster established, the GUI window appears.
   Note: Some browsers add "Warning: Applet Window" to windows and dialogs created by an applet. This is normal and should be ignored.

Configuring the Browser Security Level

In order to run the LifeKeeper web client, you may need to modify your browser security settings. Follow the procedures below.

Internet Explorer

Internet Explorer will likely automatically put your LifeKeeper servers in the Local intranet zone. If not, you should manually add all LifeKeeper servers to the Local intranet zone as follows:

1. From the Tools menu, click **Internet Options**.
2. Click the **Security** tab.
3. Select **Local intranet zone**
4. Click **Sites**.
5. Click **Advanced**.
6. Enter the server name(s) and port number(s) for all LifeKeeper server(s)to which you wish to connect (for instance: *http://server1:81*), clicking **Add** after each.
7. Click **OK** until you’re done.

Mozilla Firefox

1. From the **Tools** menu, select **Options**.
2. In the Options dialog box, click the **Content Category**.
3. Select the “**Enable Java**” and “**Enable Java Script**” options.
4. Click **OK**.

Netscape Navigator and Netscape Communicator

1. From the Edit menu, select **Preferences**.
2. In the Preferences dialog box, double-click the **Advanced Category**.
3. Select the “**Enable Java**” and “**Enable Java Script**” options.
4. Click **OK**.

   WARNING: Be careful of other sites you visit with security set to low values.

Running the GUI Application on a LifeKeeper Server

You can also run the LifeKeeper GUI as an application on a LifeKeeper server. By doing so, you are, in effect, running the GUI client and server on the same system. Only users with administrator privileges on the LifeKeeper server are allowed to run LifeKeeper applications.

1. Start the LifeKeeper GUI by clicking **Start->AllPrograms->SteelEye->LifeKeeper->LifeKeeper (Admin only)**.

2. After the application is loaded, the LifeKeeper GUI appears and the Cluster Connect dialog is displayed. Enter the **Server Name** you wish to connect to followed by the login and password. See LifeKeeper GUI User Accounts for additional information on logging in.

3. Once a connection to the cluster is established, the GUI window appears.

To run the LifeKeeper GUI on a LifeKeeper server using the web client, click **Start->All Programs-> SteelEye->LifeKeeper->LifeKeeper**. This will invoke a web browser and connect to LifeKeeper using *localhost:81.*

LifeKeeper GUI User Accounts

All LifeKeeper GUI users must belong to LifeKeeper security groups. The LifeKeeper administrator for a cluster can use local groups and user accounts on each server, or you can set up domain groups and users with local logon privileges.

Logging In

If the LifeKeeper account is the same for each server in the cluster (same login and password), then logging in to one server in the cluster will allow you to access the other servers without additional logins. You may need to enter the domain name with the user name, for example: “Southdomain\john”.

If the LifeKeeper account is different for each server in the cluster (different login names and/or passwords), then upon logging in to the first server in the cluster, you will receive the following message when LifeKeeper attempts to use the login for the next server in the cluster:

```
Access denied: invalid user name or bad password. Only
users with local privileges can use LifeKeeper. Would you
like to re-enter the authentication data?
```

Click **Yes** for a prompt to login to the next server.

Configuring GUI Users

There are three classes of GUI users with different permissions for each.

1. Users with Administrator permission throughout a cluster can perform all possible actions through the GUI.

2. Users with Operator permission on a server can view configuration and status information and can bring resources into service and take them out of service on that server.

3. Users with Guest permission on a server can view configuration and status information on that server.

The best practice is to always grant permissions on a cluster-wide basis. It is possible to grant permissions on a single-server basis, but that is confusing to users and makes it impossible to perform administrative tasks.

User administration is performed by assigning users to local user groups on each server. Users assigned to the local Administrators group have Administrator permission, users in the local LK_OPERATOR group have Operator permission and users in the local LK_GUEST group have Guest permission. The local Administrators group is built in to all Windows machines, but the other two local groups are not, so you will need to create them.

The group names can be configured on English-language machines by editing the entries in the file *Server_RB_en.properties* which can be found in the folder *$LKROOT/htdoc/com/steeleye/LifeKeeper/locale*. You can also localize the group names by creating a file *Server_RB_xx.properties* in the same folder,where "xx" is your locale code, and editing the entries in that file.

If you are working in a Domain Controller environment with no local groups or users on your servers, you can create the LK_OPERATOR and LK_GUEST groups as trusted global security groups. You will then need to set the group security policy to allow local logon to those groups.

To enable a user or a group to login locally on a Windows server, follow the instructions described below.

1. Log in to the machine using an account with local Administrator privileges.
2. Open the **Local Security Policy MMC** in the Administrative Tools program group.
3. Scroll down to **Local Policies -> User Rights Assignment.**
4. In the details pane, double-click **Allow Logon Locally** policy for Windows 2003.
5. Use the **Add User or Group**... button to add domain groups LK_OPERATOR and LK_GUEST previously created for local login right.

IMPORTANT: Please ensure that the domain GPO does not overwrite these local policy changes.

Finally, you need to propagate these changes by executing the command **SECEDIT /REFRESHPOLICY USER_POLICY gpupdate** for Windows 2003 (for more details, see http://support.microsoft.com/?kbid=227302). Once you have done this, LifeKeeper will be able to recognize members of those groups and assign them the appropriate permissions.

Note: If you create these groups and users locally on your server, the assignments affect GUI permissions only for that server. In that case, you should repeat the assignment on all servers in the cluster. This takes more work but does make the cluster more robust as it is then not dependent on access to the domain controller.

Common Tasks

The following are basic tasks that can be performed by any user.

Connecting To A Cluster

1. From the File Menu or the Global Toolbar, select **Connect**.
2. In the **Server Name** field of the Cluster Connect Dialog, enter the name of a server within the cluster to which you want to connect.

3. In the **Login** and **Password** fields, enter the login name and password of a user with LifeKeeper authorization on the specified server.

4. Click **OK**.

If the GUI successfully connects to the specified server, it will continue to connect to (and add to the status display) all known servers in the cluster until no new servers are found.

Note: If the initial login name and password fails to authenticate the client on a server in the cluster, the user is prompted to enter another login name and password for that server. If **Cancel** is selected from the Password Dialog, connection to that server is aborted and the GUI continues connecting to the rest of the cluster.

Disconnecting From A Cluster

This task disconnects your client from all servers in a cluster.

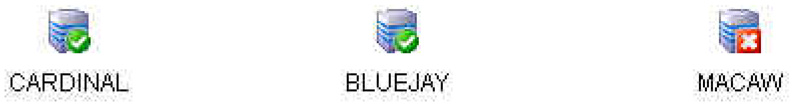
1. Select a server from which you want to disconnect and then select **Disconnect** from the Server Context Menu or Server Context Toolbar.

2. A **Confirmation dialog** listing all servers in the cluster is displayed. Click **OK** in the **Confirmation dialog** to confirm that you want to disconnect from all servers in the cluster.

After disconnecting from a cluster, all servers in that cluster are removed from the Status Table.

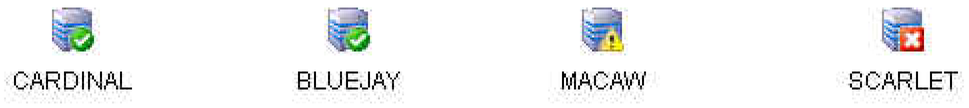
Viewing Connected Servers

The state of a server can be determined by looking at the graphic representation of the server in the GUI as shown below. See Viewing the Status of a Server for an explanation of the server states indicated visually by the server icon.



Viewing The Status Of A Server

The state of a server can be determined by looking at the graphic representation of the server in the table's header as shown below.



Server State	Visual State	What it Means
ALIVE	✓	Client has valid connection to the server. Comm paths originating from this server to an ALIVE remote server are ALIVE. Comm paths which may be marked DEAD and which target a DEAD server are ignored because the DEAD server will be reflected in its own graphic.
ALIVE	⚠	Client has valid connection to the server. One or more comm paths from this server to a given remote server are marked as DEAD. No redundant comm path exists from this server to a given remote server.
DEAD	✖	Reported as DEAD by other servers in the cluster.
UNKNOWN	?	Network connection was lost. Last known LifeKeeper state is ALIVE.

Viewing Server Log Files

To view server log files:

1. Select a server and then select **View Logs** from the Server Context Menu or Server Context Toolbar. This will bring up the Log Viewer Dialog.
2. When you are finished, click **OK** to close the dialog.

Viewing Server Properties

To view server properties:

- If the Properties Panel is enabled, simply select the server in the Status Table and the properties will be displayed in the **Properties Panel**.
- If the Properties Panel is disabled, select the server and then select **Properties** in the Server Context Menu.

Viewing Resource Tags and IDs

A resource's tag and ID can be viewed quickly by positioning the cursor over a resource icon in the status window and clicking the left mouse button once (single-click). The resource tag and ID of the server having the lowest priority number are displayed in the message bar. To display the resource tag and ID for a resource on a specific server, single-click the appropriate resource instance cell in the table.

Messages displayed in the message bar look similar to the following:

```
Resource Tag = F-Drive, Resource ID = F:
```

Under certain circumstances, the GUI may not be able to determine the resource ID in which case only the resource tag is displayed in the message bar.

Viewing the Status of Resources

The status or state of a resource is displayed in two formats: **global resource status** (across all servers) and **server resource status** (on a single server). The global resource status is shown in the **Resource Hierarchy Tree** in the left pane of the status window. The server resource status is found in the table cell where the resource row intersects with the server column.

Server Resource Status

Server Resource State	Visual State	What it Means
Active	✓	Resource is operational on this server and protected (ISP)
Degraded	⚠	Resource is operational on this server but not protected by a backup resource (ISU)
StandBy	⬇	Backup resource that can take over operation from the active resource (OSU)
Failed	☒	Problem with resource detected on this server. For example, an attempt to bring the resource in-service failed (OSF)
Unknown	?	Resource has not been initialized (ILLSTATE) or LifeKeeper is not running on this server
	Empty Panel	Server does not have the resource defined

Global Resource Status

Description	Visual State	What it Means / Causes
Normal	✓	Resource is active (ISP) and all backups are active
Warning	⚠	Resource is active (ISP); one or more backups are marked as unknown or failed (OSF)
Failed Resource is not active on any servers (OSF)	☒	Resource has been taken out of service for normal reasons Resource has stopped running by unconventional means Recovery has not been completed or has failed
Unknown Cannot determine state from available information	?	More than one server is claiming to be active Lost connection to server All server resource instances are in an unknown state

Viewing Resource Properties

To view resource properties:

- If the Properties Panel is enabled, simply select the server-specific resource instance in the Status Table and the properties will be displayed in the **Properties Panel**.
- If the Properties Panel is disabled, select the server-specific resource instance and then select **Properties** in the Resource Context Menu.

Viewing Message History

1. On the View Menu, click **History**. The **Message History** dialog is displayed (see below).
2. If you want to clear all messages from the history, click **Clear**.
3. Click **OK** to close the dialog box.

The **Message History** dialog displays the most recent messages from the message bar. The history list can display a maximum of 1000 lines. When the maximum number of lines is exceeded, the new messages will "push out" the oldest messages.

These messages represent only the actions between the client and the server and are displayed in chronological order, the most recent messages appearing at the top of the list.

Reading the Message History

<-- indicates that the message is incoming from a server and typically has a format of:

```
<--"server name":"action"

<--"server name":"app res": "action"

<--"server name":"res instance":"action"
```

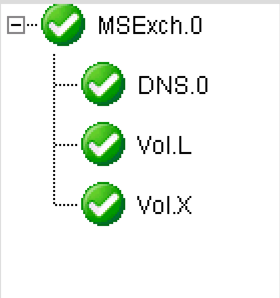
--> indicates that the message is outgoing from a client and typically has a format of:

```
-->"server name":"action"

-->"server name":"app res": "action"

-->"server name":"res instance":"action"
```

The **Clear** button clears the history but does not close the dialog.

The **OK** button closes the dialog without clearing the history.

Expanding and Collapsing A Resource Hierarchy Tree

	The resource *MSExch.0* is collapsed in this tree. The resource *MSExch.0* is expanded in this tree. ⊞ appears if it is collapsed. ⊟ appears to the left of a resource icon if it is expanded.

To expand a resource hierarchy,

- click the ⊞ or
- double-click the resource icon to the right of a ⊞.

To expand all resource hierarchies,

- On the View Menu, click **Expand Tree** or
- Double-click the **Hierarchies** button in the top left corner of the Status Table.

Note: The resource tag/ID shown in the resource hierarchy belongs to the server having the lowest priority number. If you wish to see the tag/ID for a resource on a specific server, left-click the resource instance cell in the table and its tag/ID will be displayed in the message bar.

To collapse a resource hierarchy,

- click the ⊟┈ or
- double-click the resource icon to the right of a ⊟┈.

To collapse all resource hierarchies,

- On the View Menu, click **Collapse Tree**, or
- Double-click the **Hierarchies** button in the top left corner of the Status Table.

Operator Tasks

The following topics are more advanced tasks that require Operator permission, such as bringing resources in and out of service.

Bringing A Resource In Service

To bring a resource in service:

1. Select a server-specific resource instance that you want to bring in service and then select **In Service** from the Resource Context Menu or Resource Context Toolbar.
2. A dialog appears confirming the server and resource(s) that you have selected to bring into service. This dialog will include a warning if you are bringing a dependent child resource into service without bringing its parent resource into service as well. Click **In Service** to bring the resource(s) into service along with any dependent child resources.
3. If the output panel is enabled, the dialog closes and the results of the commands to bring the resource(s) in service are shown in the output panel. If not, the dialog remains up to show these results, and you click **Done** to finish when all results have been displayed.
4. Errors that occur while bringing a resource in service are logged in both the LifeKeeper log and the GUI log of the server on which you want to bring the resource into service.

Taking a Resource Out Of Service

To take a resource out of service:

1. Select a server-specific resource instance that you want to take out of service and then select **Out of Service** from the Resource Context Menu or Resource Context Toolbar.
2. A dialog appears confirming the selected resource(s) to be taken out of service. This dialog will include a warning if you are taking a dependent child resource out of service without taking its parent resource out of service as well. Click **Out of Service** to take the resources out of service.
3. If the output panel is enabled, the dialog closes and the results of the commands to take the resource(s) out of service are shown in the output panel. If not, the dialog remains up to show these results and you click **Done** to finish when all results have been displayed.

4. Errors that occur while taking a resource out of service are logged in both the LifeKeeper log and the GUI log of the server on which you want to take the resource out of service.

Taking Volume Resources In and Out Of Service

Some background processes such as virus scanners and Windows Services will try to access a shared volume for write access. These processes may be required to run applications and cannot be arbitrarily turned off for long periods of time.

In most cases, these applications will have no effect on the manual switchover of LifeKeeper volumes. However, if you see error messages similar to the following during a manual switchover (actual failovers are not affected), you should use the Volume Remove Stop/Restart feature described below.

```
*ERROR*  [No. 12035]  Unable to lock volume <volume ID> on <system
name> machine at this time as it may be in use by some application.
Please free this volume and try again.
```

If the volume has users doing non-write access only (remote links, local opens), removing the volume from service succeeds as does restoring the volume on the other system (for example, manual switchover). The existing user "opens" are, of course, no longer valid. However, they prevent the volume from being restored to service on the original system whether it is manually switched back over or automatically failed back over to the original system. Attempts to do so result in the following error message:

```
*ERROR* [No. 12046] LifeKeeper RESTORE VOLUME <volume ID>
FAILED(err=<error number>).
```

The end result is that removing or restoring a volume, switching a volume over or back or switching any hierarchy that includes a volume resource over or back fails if the volume has any users, local or remote.

In addition, the system’s PATH variable must not contain any file shares that exist on the volume to be protected by LifeKeeper. File shares in the PATH variable may also cause volume operations to fail. Remove any such shares prior to creating the volume resource in LifeKeeper. The PATH variable may be modified by selecting **System** then **Environment** from the **Windows Control Panel**.

Volume Remove Stop/Restart Programs and Services

LifeKeeper provides configurable registry keys that permit the user to specify programs and services to be stopped and restarted on a system during a volume remove operation. This feature starts and stops user specified programs and services when LifeKeeper detects open handles on any volume that is being removed and taken out of service on that system. However, if LifeKeeper detects no open handles on the volume being removed from service, it will not attempt to stop anything on that system. Instructions for using this feature are as follows:

1. Determine which programs and services are accessing the volume in question and preventing successful volume failovers.

2. Specify programs to stop and start by adding a subkey for each program under:

- **32-Bit:** `HKEY_LOCAL_MACHINE\SOFTWARE\SteelEye\LifeKeeper\VolumeRemoveStopPrograms\*`
- **64-Bit:** `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\SteelEye\LifeKeeper\VolumeRemoveStopPrograms\*`
  For example, to stop a program called "myapp.exe", add the following subkey: `HKEY_LOCAL_MACHINE\SOFTWARE\SteelEye\LifeKeeper\VolumeRemoveStopPrograms\myapp.exe\`

3. Under the subkey for each program, add the following values (all are REG_SZvalues):

ProgramName	the program name only (Note: Must match subkey name from Step 2.)
ProgramPath	the program name, including full path
Restart	a value of 0 indicates that the program should not be restarted; a value of 1 indicates that the program should be restarted
StartCmdLine	optional command line arguments to be used when starting the program
WasRunning	used by LifeKeeper to save the number of instances of the program that were running before stopping them; should be initialized to 0

For example, values could be entered to stop and start "myapp.exe /a /t /p" with its associated arguments as follows:

ProgramName	myapp.exe
ProgramPath	C:\mydir\myapp.exe
Restart	1
StartCmdLine	/a /t /p
WasRunning	0

Note: By default, LifeKeeper includes a subkey for stopping perfmon.exe and the restart option is disabled.

1. Specify services to stop and restart by adding a subkey for each service under:

- **32 Bit:** `HKEY_LOCAL_MACHINE\SOFTWARE\SteelEye\LifeKeeper\VolumeRemoveStopServices\`
- **64 Bit:** `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\SteelEye\LifeKeeper\VolumeRemoveStopServices\`
  For example, to stop a service called "mysvc", you would add the

following subkey: `HKEY_LOCAL_MACHINE\SOFTWARE\SteelEye\LifeKeeper\VolumeRemoveStopServices\mysvc\`

2. Under the subkey for each service, add the following values (all are REG_SZvalues):

ServiceName	the service display name (Note: Must match subkey name from Step 4.)
Restart	a value of 0 indicates that the program should not be restarted; a value of 1 indicates that the program should be restarted
WasRunning	used by LifeKeeper to save the number of instances of the program that were running before stopping them; should be initialized to 0
StopWait	the number of seconds to wait for the service to reach the STOPPED state. If StopWait is negative, then it (and the failover) will wait indefinitely for the service to reach the STOPPED state
StartWait	the number of seconds to wait for the service to reach the RUNNING state. If the service does not reach the RUNNING state in the configured period, an error is logged in the Event Log. If StartWait is negative, then the failover will wait indefinitely for the service to start. If it is set to 0, then the service will be started but will not wait for it to reach the RUNNING state (and thus doesn't generate an Event Log message if the service can't be restarted)

For example, values could be entered to stop and start "mysvc" as follows:

ServiceName	mysvc
Restart	1
WasRunning	0
StopWait	120
StartWait	120

Volume Restore Stop/Restart Programs and Services

LifeKeeper provides configurable registry keys that permit the user to specify programs and services to be stopped and restarted on a system during a volume restore operation. This feature is similar to the Volume Remove Stop/Restart Programs and Services feature described above with the following differences:

- Programs to stop and restart are specified as subkeys under:
  - **`32 Bit: HKEY_LOCAL_MACHINE\SOFTWARE\SteelEye\LifeKeeper\VolumeStopPrograms\`**
  - **`64 Bit:  HKEY_LOCAL_`**

MACHINE\SOFTWARE\Wow6432Node\SteelEye\LifeKeeper\VolumeStopPrograms\

- Services to stop and restart are specified as subkeys under:
  - **32 Bit: HKEY_LOCAL_MACHINE\SOFTWARE\SteelEye\LifeKeeper\VolumeStopServices\**
  - **64 Bit: HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\SteelEye\LifeKeeper\VolumeStopServices\**

Volume Shadow Copy (VSS)

Using Volume Shadow Copy (VSS) with DataKeeper/LifeKeeper Volumes

On Windows 2003 and 2003 R2, VSS Shadow Copy cannot be enabled on LifeKeeper-protected (shared or replicated) volumes. Configuring a snapshot of a protected volume, even if the snapshot is stored on a different volume, will prevent LifeKeeper and DataKeeper from being able to lock the volume making it impossible to protect the data on the volume.

On Windows 2008 and 2008 R2, VSS Shadow Copy can be enabled for LifeKeeper-protected (shared or replicated) volumes. However, the following guidelines apply:

- VSS snapshot images must not be stored on a LifeKeeper-protected volume.
  Storing VSS snapshots on a LifeKeeper-protected volume will prevent LifeKeeper from being able to lock the volume and switch it over to another node.
- When a LifeKeeper-protected volume is switched or failed over, any previous snapshots that were taken of the LifeKeeper protected volume are discarded and cannot be reused.
- VSS snapshot scheduling is not copied between the LifeKeeper servers. If snapshots are scheduled to be taken twice a day on the primary server and a switchover occurs, this schedule will not be present on the backup server and will need to be redefined on the backup server.
- There is a slight difference in behavior when switching back to a server where snapshots were previously enabled:
- If the volume is a shared volume, VSS snapshots must be re-enabled.
- If the volume is a replicated volume, VSS snapshots are automatically re-enabled.

Volume Locking for Shared SCSI Volumes

When you want to protect resources on shared SCSI disks, you partition the shared disk into logical volumes using the **Windows Disk Management** tool. LifeKeeper can protect shared volumes by defining a volume resource instance. Each instance is assigned a drive letter (for example, G:).

LifeKeeper brings the volume resource instance into service on the primary server and provides software locks so that a backup server cannot access the volume while it is active on the primary server. In case of a failure of the primary server, LifeKeeper automatically brings the volume resource into service on the backup server and locks the primary server from accessing the volume resource when it is repaired.

LifeKeeper also automatically changes the primary and designations so that the failed server is now locked from access to the volume resource. In this way, the resource is protected from inappropriate access while you repair the failed server.

This dynamic redefinition of primary and backup servers is LifeKeeper's intelligent switchback feature that allows you to select the appropriate time to bring the resource back into service on the repaired system.

Since LifeKeeper maintains the volume locks, do not stop LifeKeeper, as this would disable the locks.

Advanced Topics

LifeKeeper Configuration Database (LCD)

The LifeKeeper configuration database (LCD) maintains the object-oriented resource hierarchy information and stores recovery direction information for all resource types known to LifeKeeper. The data is cached within system shared memory and stored in files so that configuration data is retained over system restarts. The LCD also contains state information and specific details about resource instances required for recovery.

See the following topics for information on the LCD directory structure, types of data stored, resource types available and use of application scripts:

- LCD Directory Structure
- Diagram of LCD Directory in $lkroot/LifeKeeper
- LCD Configuration Data

LCD Directory Structure

Major subdirectories under $LKROOT (by default, c:\LK):

- **Admin**. Scripts for LifeKeeper core and Recovery Kits.
- **Config**. LifeKeeper configuration files, including shared equivalencies.
- **Bin**. LifeKeeper executable programs.
- **Subsys**. Resources and types. LifeKeeper provides resource and type definitions in subdirectories of Subsys. For instance, communications resources are stored in comm, volume resources are stored in filesys, and generic application resources are stored in gen. Optional Recovery Kits may create different resource types stored in different directories. For example, database application resources are stored in database.

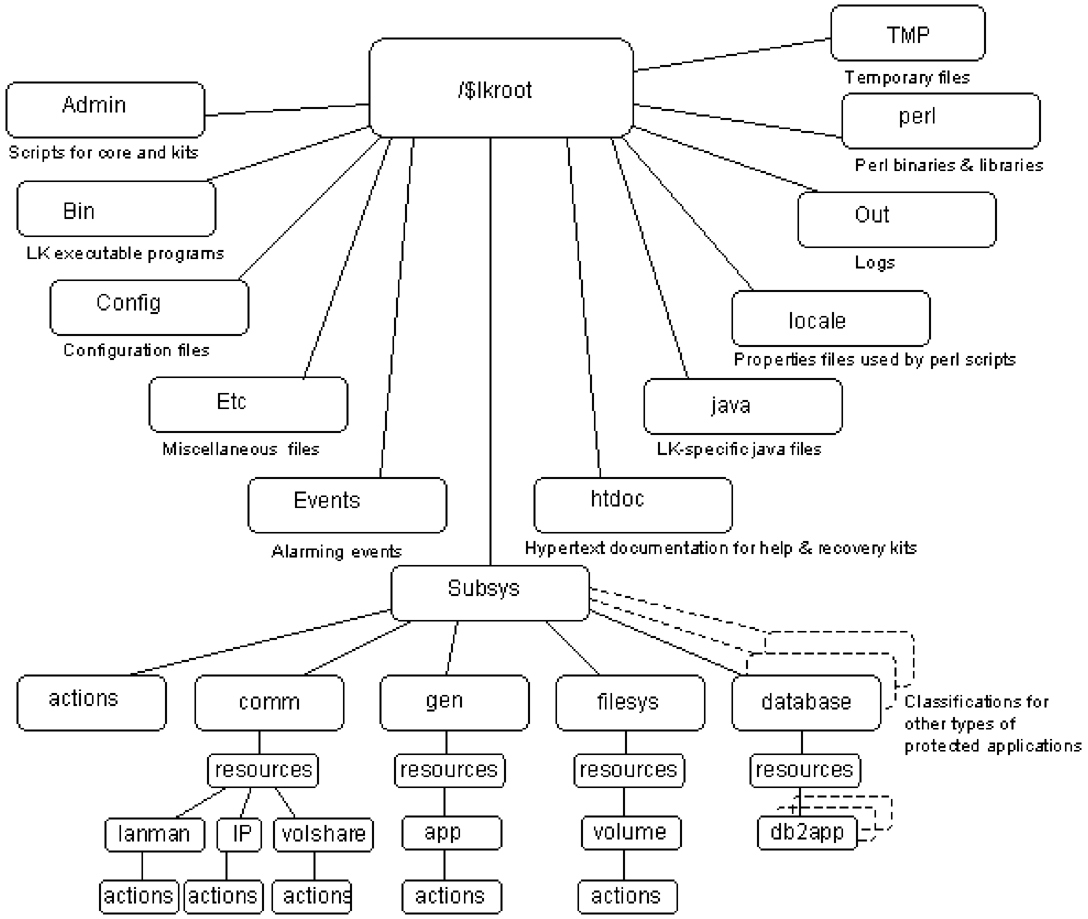
- **Events**. Event alarms.
- **Out**. LifeKeeper logs. LifeKeeper sends a variety of error and status messages to several different logs in this directory.
- **perl**. Perl binary executables and libraries.

The structure of the LCD directory in $LKROOT is shown in the topic Diagram of LCD Directory.

Note: The location of these subdirectories can be changed by modifying the value of LKROOT in the environment.

Diagram of LCD Directory

The following diagram shows the directory structure of \$lkroot.



LCD Configuration Data

LCD stores the following related types of data:

- Dependency Information
- Resource Status Information
- Inter-Server Equivalency Information

Dependency Information

For each defined resource, LifeKeeper maintains a list of dependencies and a list of dependents (resources depending on a resource). For more information, see the LCDI_relationship and LCDI_instances manual pages.

Resource Status Information

LCD maintains status information in memory for each resource instance. The resource states recognized by LCD are **ISP**, **OSF**, **OSU**, and **ILLSTATE**. Resources may change from one state to another when a system event occurs or when an administrator takes certain actions. When a resource changes states, the status change is reflected in the LCD on the local server as well as in the database of the backup servers for that resource.

Inter-Server Equivalency Information

Relationships may exist between resources on various servers. A shared equivalency is a relationship between two resources on different servers that represent the same physical entity. When two servers have a resource with a shared equivalency relationship, LifeKeeper attempts to ensure in its actions that only one of the two servers has the resource instance in the in-service, protected [ISP] state at any one time. Both servers can have the resource instance in an out-of-service state [OSU or OSF], but for data integrity reasons, only one server can have the resource in service at any given time.

Disks on a Small Computer System Interface (SCSI) bus are one example of equivalent resources.

Furthermore, the dependency relationships within a hierarchy guarantee that all resources that depend upon the volume, such as a file share, are in service on only one server at a time.

LCD Resource Types

The LCD is maintained in both shared memory and in the $LKROOT directory. As highlighted on the directory structure diagram, subsys contains application resource sets you can use to define your application interface:

- filesys - file system related resources like volume
- comm - communications related resources like IP, volshare (fileshare) and lanman
- database - database resources such as Oracle

These subdirectories are discussed in Resources Subdirectories.

Resources Subdirectories

The *filesys*, *comm*, *WebServer*, *database*, *mail* and *appsuite* directories each contain a *resources* subdirectory. The content of those directories provides a list of the resource types that are currently defined and managed by LifeKeeper:

- **filesys resource types**. You find these resource types in the *$LKROOT\LifeKeeper\subsys\filesys\resources* directory:
  - **volume**—disk partitions or virtual disk devices
- **comm resource types**. You find these resource types in the */$LKROOT/LifeKeeper/subsys/comm/resources* directory:
  - **IP**—created by the IP recovery kit
  - **DNS**—created by the DNS recovery kit
  - **volshare**—fileshare resources created by the LAN Manager recovery kit
  - **lanman**—computer alias created by the LAN Manager recovery kit
- **WebServer resource types**. You find these resource types in the *$LKROOT\LifeKeeper\subsys\WebServer\resources* directory:
  - **IIS**—created by the IIS recovery kit
- **database resource types**. You find these resource types in the *$LKROOT\LifeKeeper\subsys\database\resources* directory:
  - **Oracle**
  - **Microsoft SQL Server**
- **mail resource types**. You find these resource types in the *$LKROOT\LifeKeeper\subsys\mail\resources* directory:
  - **Microsoft Exchange Server**

Each resource type directory contains one or more of the following:

- ***instances***. This file reflects the permanent information saved in the LCD about resource instances. It contains descriptive information for the resource instances associated with this resource type.
- ***actions***. This directory contains the set of recovery action programs that act only on resource instances of the specific resource type. If, for your application, any actions apply to all resource types within an application, place them in an *actions* subdirectory under the *application* directory rather than under the *resource type* directory.

Recovery direction software is used to modify or recover a resource instance. Two actions, **remove** and **restore**, must exist in the *actions* directory for each resource type.

Resource Actions

The *actions* directory for a resource type contains the programs (most often shell scripts) that describe specific application functions. Two actions are required for every resource type—restore and remove.

The remove and restore programs should perform symmetrically opposite functions; that is, they undo the effect of one another. These scripts should never be run manually. They should only be run by executing the LifeKeeper Recovery Action and Control Interface (LRACI) perform_action shell program described in the LRACI-perform_action man page.

LCDI Commands

LifeKeeper provides two mechanisms for defining an application resource hierarchy:

- LifeKeeper GUI
- LifeKeeper Configuration Database Interface (LCDI) commands

The LCDI is a set of interface commands provided by LifeKeeper that you can use to create and customize resource hierarchy configurations to meet your application needs. You use the command interface when an application depends upon multiple resources (such as two or more file systems).

For a description of the commands, see the LCDI manual pages. This topic provides a development scenario that demonstrates the way you can use both the GUI and command functions to create a resource hierarchy.

LifeKeeper Communications Manager (LCM)

The LifeKeeper Communication Manager (LCM) provides reliable communication between processes on one or more LifeKeeper servers. This process can use redundant communication paths between systems so that failure of a single communication path does not cause failure of LifeKeeper or its protected resources. The LCM supports a variety of communication alternatives including RS-232 (TTY), TCP/IP and shared disk connections.

The LCM provides the following:

- **LifeKeeper Heartbeat**. Periodic communication with other connected LifeKeeper systems to determine if the other systems are still functioning. LifeKeeper can detect any total system failure that is not detected by another means by recognizing the absence of the heartbeat signal.
- **Administration Services**. The administration functions of LifeKeeper use the LCM facilities to perform remote administration. This facility is used for single-point administration, configuration verification and sanity checking of administrative actions.
- **Configuration and Status Communication**. The LifeKeeper configuration database (LCD) tracks resource status, availability and configuration through the LCM facilities. These facilities allow the LCD to maintain consistent resource information between the primary and secondary systems.

- **Failover Recovery**. If a resource fails on a system, the LCM notifies LifeKeeper to recover the resource on a backup system.

Communications Status Information

The **Communications** tab of the Server Properties dialog lists the servers known to LifeKeeper and their current state followed by information about each communication path.

Maintenance Tasks

The following are tasks for maintaining LifeKeeper.

Starting and Stopping LifeKeeper

Because LifeKeeper is typically started automatically after installation and each time the server is booted, you should not normally need to start/stop LifeKeeper. (The only exception is if you chose to do a Custom installation and opted not to start LifeKeeper at that time.)

In the event that you need to start or stop LifeKeeper manually, you should do so using the **Services** tool under **Administrative Tasks** in the Windows Control Panel.

Starting LifeKeeper

LifeKeeper consists of two services:

- LifeKeeper
- LifeKeeper External Interfaces

Generally, these two services should be stopped and started together. However, since LifeKeeper External Interfaces is a dependency of the LifeKeeper service, stopping it will also stop the LifeKeeper service. Likewise, it must be started before the LifeKeeper service can be started.

Select **LifeKeeper** and click **Start**. This will automatically start the **LifeKeeper External Interfaces** service.

Stopping LifeKeeper

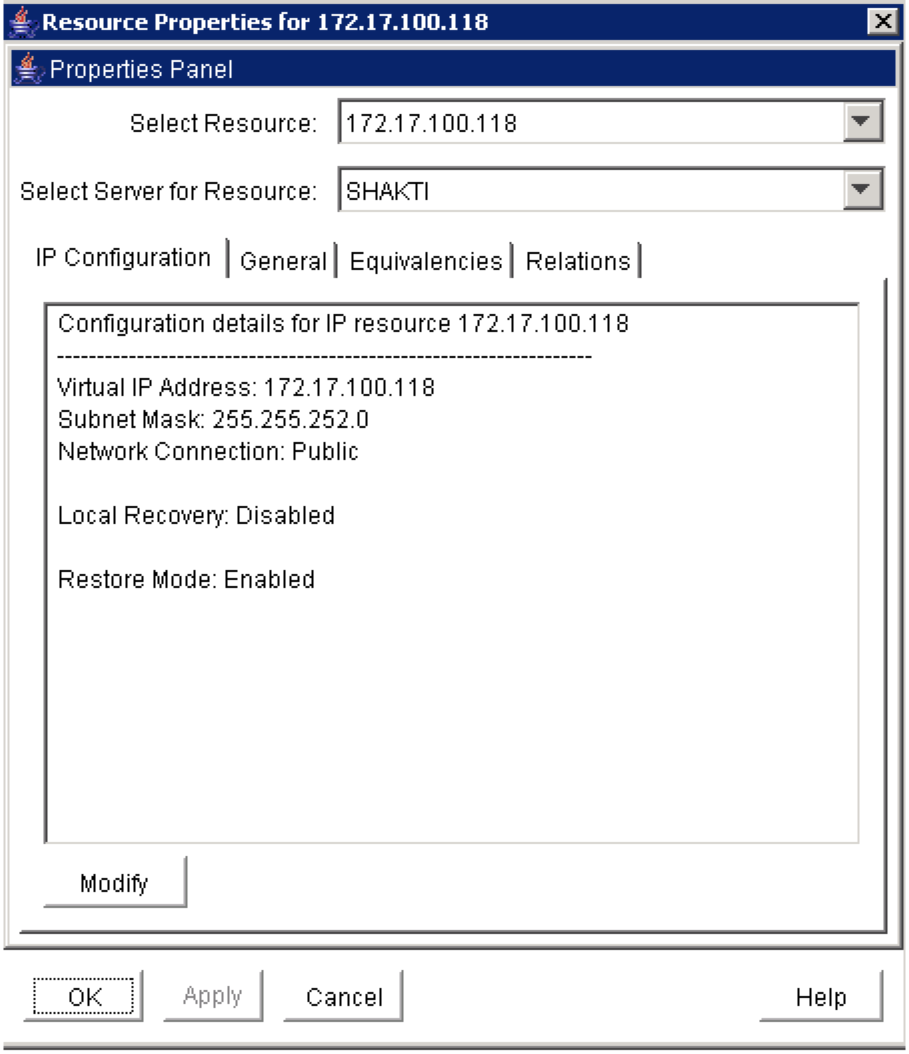
In the **Services** tool, select **LifeKeeper External Interfaces** and click **Stop**. This will stop both services. Note that the length of time that it takes to stop LifeKeeper will vary depending upon the hierarchies currently configured although the Services tool shows the services as stopped immediately.

Using the command line to enter **$LKROOT\bin\lkstop** will more accurately show the services being stopped, and it will confirm with the message **"LIFEKEEPER NOW STOPPED"**.

Note: Stopping LifeKeeper takes all protected hierarchies out of service. This means that any protected applications will not be accessible.

Managing IP Resources

To view configuration information associated with a protected IP resource from the LifeKeeper GUI, right-click on the IP resource (on the right-hand side of the LifeKeeper GUI) and select **Properties**, then select the **IP Configuration** tab. The example below shows the configuration details for the LifeKeeper protected IP resource 172.17.100.118.



Select Resource: 172.17.100.118

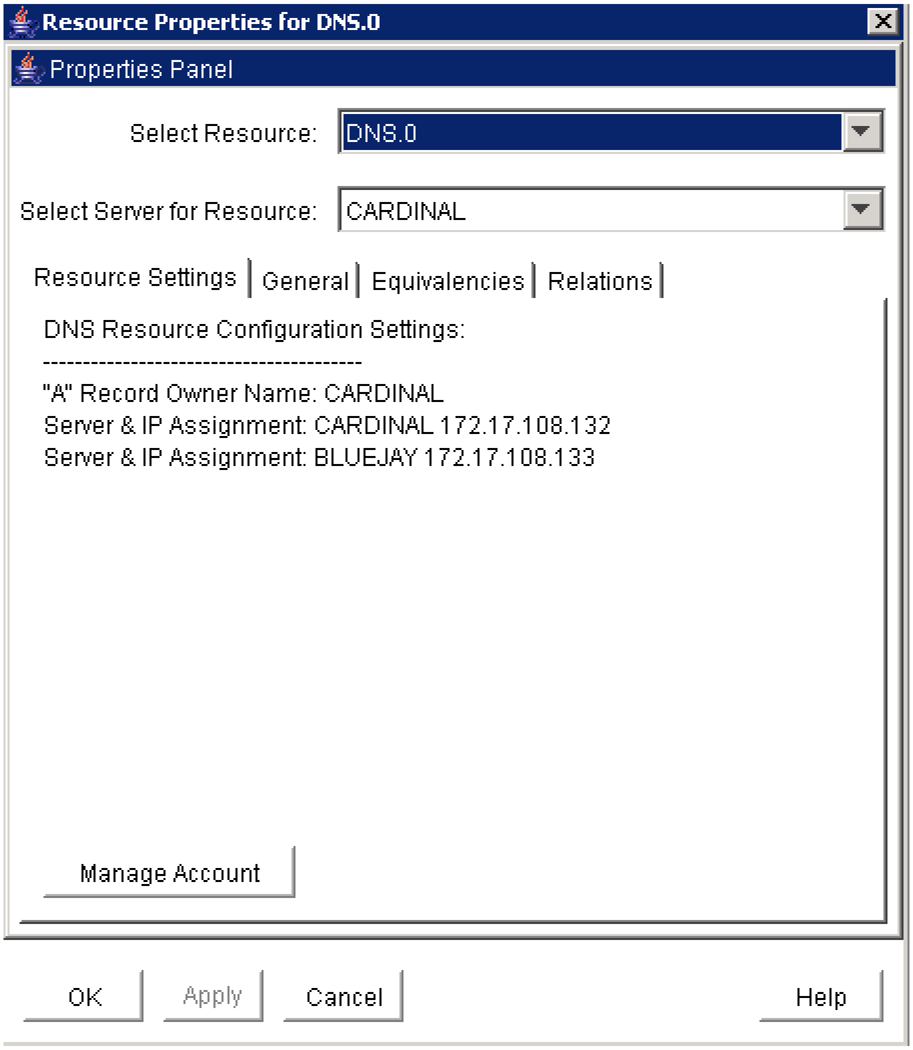
Select Server for Resource: SHAKTI

```
Configuration details for IP resource 172.17.100.118
-----------------------------------------------------------------
Virtual IP Address: 172.17.100.118
Subnet Mask: 255.255.252.0
Network Connection: Public

Local Recovery: Disabled

Restore Mode: Enabled
```

To enable or disable the IP address restore capabilities on the selected server while still allowing the LifeKeeper IP resource to report a successful in-service operation, click the **Modify** button, then select **Enable or Disable** for the restore mode. This feature applies to three node LifeKeeper clusters where two nodes are on a LAN (same subnet) and the third node is on a WAN (different subnet). The restore mode of the IP resource would be enabled on the LAN nodes and disabled on the WAN node.

Managing DNS Resources

To change the Domain administrative user and password associated with a protected DNS resource from the LifeKeeper GUI, right-click on the DNS resource (on the right-hand side of the LifeKeeper GUI) and select **Properties**, then select the **Resource Settings** tab. Select **Manage Account** on the Resource Settings page to change the Domain administrative user and password for your DNS resource.



Select Resource: DNS.0

Select Server for Resource: CARDINAL

DNS Resource Configuration Settings:

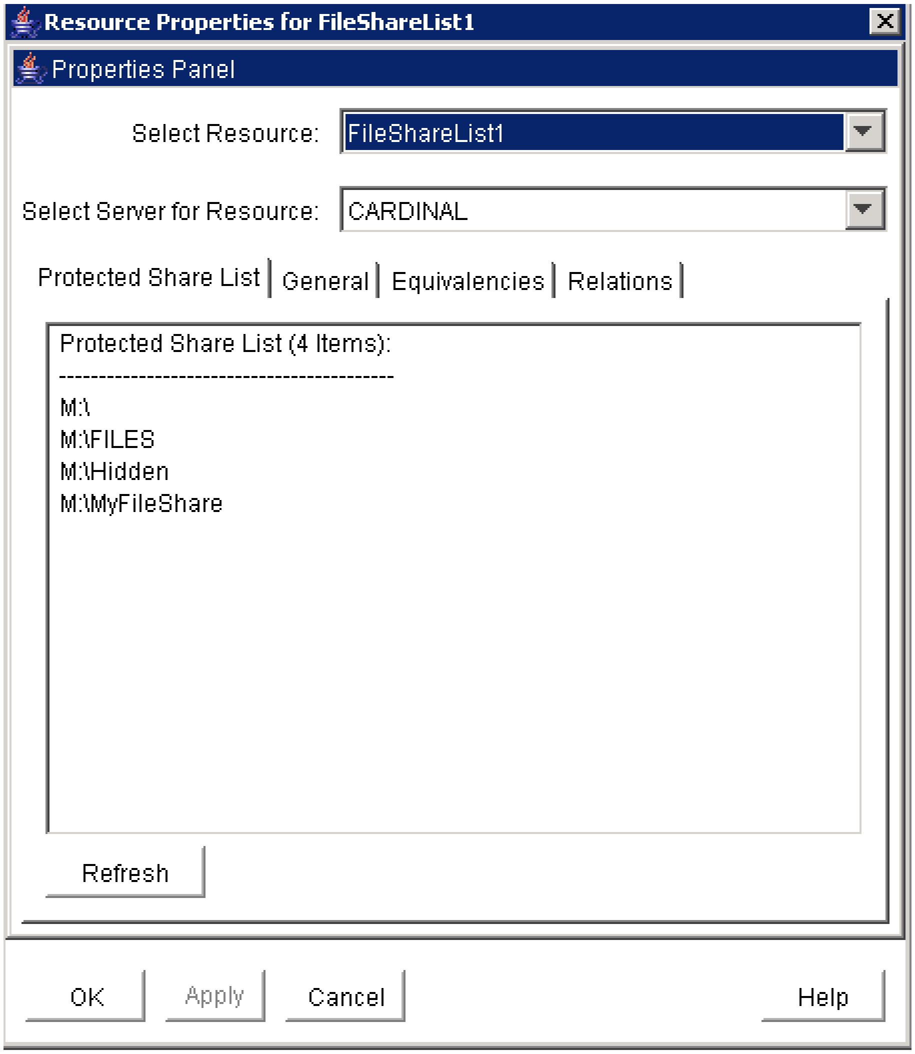
"A" Record Owner Name: CARDINAL
Server & IP Assignment: CARDINAL 172.17.108.132
Server & IP Assignment: BLUEJAY 172.17.108.133

Manage Account :

Field	Tips
Enter User ID (Domain\UserID)	Enter the user name of the Windows DNS/Domain administrator. This user account should have privileges to make changes in the DNS configuration and should be a member of the "Domain Admins" group in the same domain as the DNS server. Enter the user ID in <*DomainName*>\<*UserID*> format where <*DomainName*> is the NetBIOS name of the domain.
Enter Password	Enter the password for the account previously entered.

Displaying List of Protected File Shares

To display the list of file shares associated with a protected file share resource from the LifeKeeper GUI, right-click on the File Share resource (on the right-hand side of the LifeKeeper GUI) and select **Properties**, then select the **Protected Share List** tab.



Select Resource: FileShareList1

Select Server for Resource: CARDINAL

```
Protected Share List (4 Items):
-------------------------------------------
M:\
M:\FILES
M:\Hidden
M:\MyFileShare
```

EditFileShareResource Utility

The EditFileShareResource utility can be used to update a file share resource with all current file shares on the associated volume(s). This can be useful in environments where there are a large number of file shares and file shares have been added or deleted since the resource was created. Using the utility can prevent the need to delete and re-create the file share resource.

To invoke the utility, on the command line enter:

```
EditFileShareResource <Tag name>
```

where <Tag name> is the tag name of a file share resource that is currently in service.

The utility protects **all eligible file shares** defined on the protected volumes that are associated with the file share hierarchy. It deletes any previously protected shares that have been deleted from the system and adds newly defined shares (meeting the eligibility criteria) to the list. It will also update the file share permissions defined on the file share.

Transferring Resource Hierarchies

When you need to perform routine maintenance or other tasks on a LifeKeeper Server, you can use the LifeKeeper GUI to move in-service resources to another server. To transfer in-service resource hierarchies from Server A to Server B, use the GUI to bring the hierarchies into service on Server B. Repeat until all of Server A’s resources have been placed in-service on their respective backup servers. See Bringing a Resource In Service for instructions.

When all of Server A’s resources are active on their backup server(s), you can shut down Server A without affecting application processing. For the maintenance period, however, the resources may not have LifeKeeper protection depending on the number of servers in the cluster.

Performing Off-Line Maintenance On A Shared Disk

When performing off-line maintenance on a shared SCSI host adapter or a disk on a shared bus, you must stop LifeKeeper and power down all servers and shared disks. Perform these actions in the following order:

1. **Stop LifeKeeper.** Use the **Services** tool to stop the LifeKeeper and LifeKeeper External Interfaces services on each LifeKeeper server. Your resources are now unprotected.
2. **Shut down Windows.** Shut down the Windows operating system on all servers in the cluster.
3. **Power down all servers.**
4. **Power OFF all shared disks.**
5. **Perform maintenance.** Perform the necessary maintenance on the shared SCSI host adapter or shared disk.
6. **Power ON all shared disks.**
7. **Power ON all servers, one at a time.** Let each server boot the Windows operating system

completely before powering on the next server.

8. **Start LifeKeeper.** Log on as administrator, then use the **Services** tool to start the LifeKeeper and LifeKeeper External Interfaces services on each LifeKeeper server. LifeKeeper automatically mounts all shared file systems and restarts and brings into service all databases on shared disks.

Maintaining a LifeKeeper Protected System

When performing shutdown and maintenance on a LifeKeeper-protected server, you must put that system's resource hierarchies in-service on the backup server before performing maintenance. This process stops all activity for shared disks on the system needing maintenance. For off-line maintenance of a shared disk, see Off-Line Maintenance of a Shared Disk.

Perform these actions in the order specified where Server A is the primary system in need of maintenance and Server B is the backup server:

1. **Bring hierarchies in-service on Server B.** On the backup, Server B, use the LifeKeeper GUI to bring in-service any resource hierarchies that are currently in-service on Server A. This will unmount any file systems currently mounted on Server A that reside on the shared disks under LifeKeeper protection. See Bringing a Resource In Service for instructions.

2. **Stop LifeKeeper on Server A.** In the **Services** tool, select **LifeKeeper External Interfaces** and click **Stop**. This will stop both services. Your resources are now unprotected.

3. **Shut down Server A.** Shut down the Windows operating system on Server A, then power off the server.

4. **Perform maintenance.** Perform the necessary maintenance on Server A.

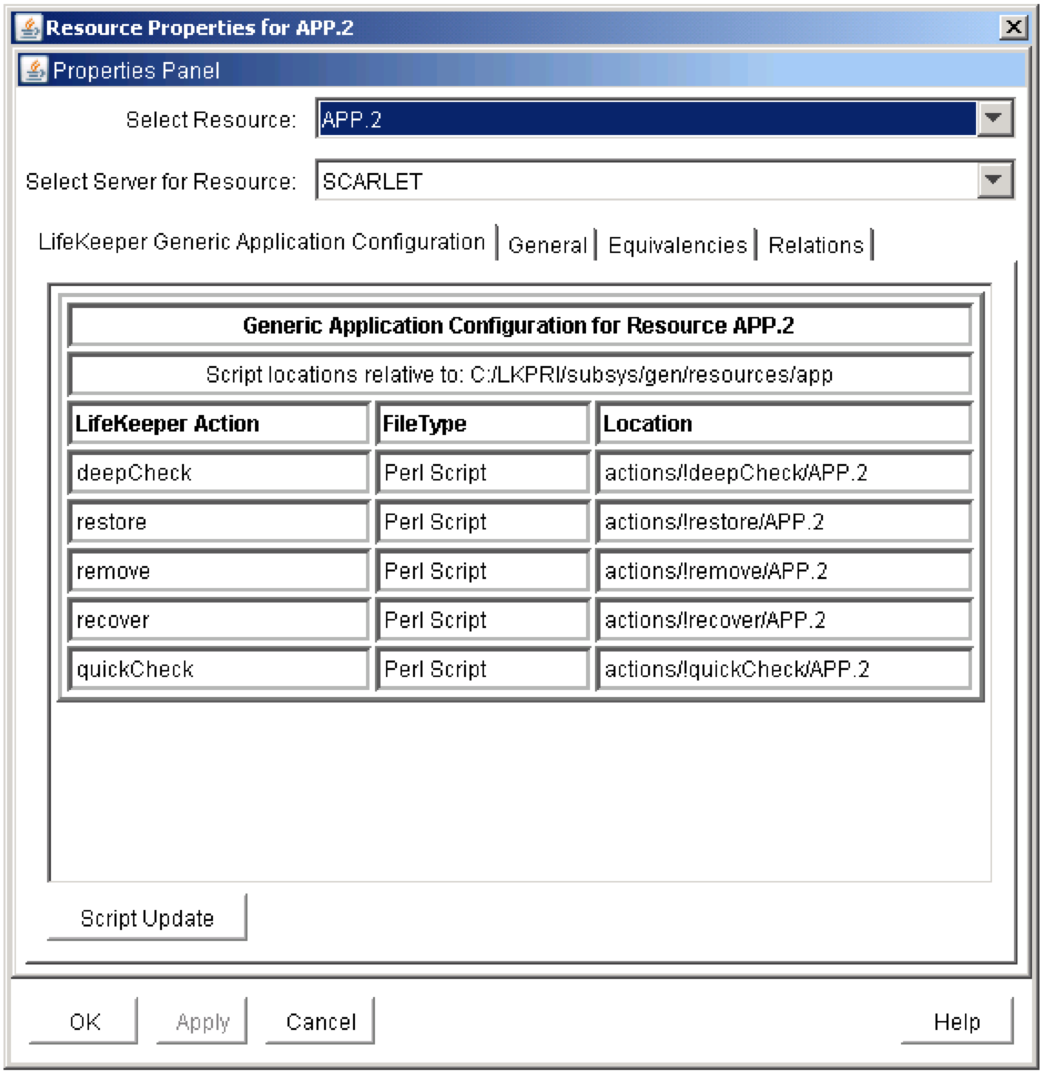
5. **Power on Server A.** Power on Server A and bring up the Windows operating system.

6. **Bring hierarchies back in-service on Server A, if desired.** On Server A, use the LifeKeeper GUI to bring in-service all resource hierarchies that were switched over to Server B.

Configuring Generic Application Scripts

Use this feature to update a script that has been created to protect an application that has no associated SteelEye recovery kit.

1. Right-click on the generic application resource and select **Properties**. Select the **LifeKeeper Generic Application Configuration** tab.



Select Resource: APP.2

Select Server for Resource: SCARLET

Generic Application Configuration for Resource APP.2

Script locations relative to: C:/LKPRI/subsys/gen/resources/app

LifeKeeper Action	FileType	Location
deepCheck	Perl Script	actions/!deepCheck/APP.2
restore	Perl Script	actions/!restore/APP.2
remove	Perl Script	actions/!remove/APP.2
recover	Perl Script	actions/!recover/APP.2
quickCheck	Perl Script	actions/!quickCheck/APP.2

2. Select the **Script Update** button. Use the following table to complete the fields in the Generic Application Configuration procedure.

Field	Tips
Select the Action Script to Update	Select the LifeKeeper action name for the resource that will be updated. Select: • **restore** to update the script responsible for in-service operations. • **remove** to update the script responsible for out-of-service operations. • **quickCheck** to update the script responsible for monitoring the application. • **deepCheck** to update the script that performs in-depth monitoring of the application. • **recover** to update the script responsible for resource recover operations. • **delete** to update the script that performs any additional actions required to remove the application from LifeKeeper protection. • **extend** to update the script responsible for additional actions required to prepare the application for protection with LifeKeeper on the target server(s)
Full Path to New Script	Enter the pathname for the shell script or object program for the application. • The **restore** script is responsible for bringing a protected application resource in-service. (Required) • The **remove** script is responsible for bringing a protected application resource out-of-service. (Required) • The **quickCheck** script is responsible for monitoring a protected application resource after a failure event. • A copy of this script or program will be saved by LifeKeeper in the resource hierarchy on the server. • There may be a short wait while LifeKeeper validates the pathname to remove monitoring or recovery. • Do not specify a shell script or object program. • Valid characters allowed in the script pathname are letters, digits and the following special characters: - _ ! . /

3. The **Basic File Statistics** dialog displays old and new configuration information about the current script. Click **Continue**.

4. The **Update All Systems** dialog displays. Select **Yes** to update all systems in this cluster. Select **No** to only update the current system. If you choose **No**, you must separately update the corresponding script for the configuration on the backup servers. Click **Next**.

5. Click **Done** to complete.

Maintaining a Resource Hierarchy

You can perform maintenance on a resource hierarchy while maintaining LifeKeeper protection of all other hierarchies on the system. This involves taking the hierarchy in need of maintenance out of service and then bringing it back in service after you complete the maintenance tasks.

To perform maintenance on a resource hierarchy:

1. **Take the hierarchy out of service.** Use the LifeKeeper GUI to take as much of the resource hierarchy out of service as you need to perform the maintenance. See Taking a Resource Out of Service for instructions.

2. **Perform maintenance.** Perform the necessary maintenance on the resource hierarchy.

3. **Restore the hierarchy.** Use the LifeKeeper GUI to bring the resource hierarchy back in service. See Bringing a Resource In Service for instructions.

Recovering After a Failover

After LifeKeeper performs a failover recovery from a primary server (ServerA) to a backup server (Server B), perform the following steps:

1. **Monitor failover.** When LifeKeeper on Server B performs a failover recovery from Server A, status messages are displayed during the failover ending with the following message:

   ```
   FAILOVER RECOVERY OF MACHINE Server A

   FINISHED AT: date time year
   ```

   The exact output depends upon the configuration. Some messages on failure to mount or unmount are expected and do not suggest failure of recovery. These messages as well as any errors that occur while bringing the resource in-service on Server B are logged in the LifeKeeper log.

2. **Perform maintenance.** Determine and fix the cause of the failure on Server A. Server A may need to be powered down to perform maintenance.

3. **Reboot Server A, if necessary.** Once maintenance is complete, reboot Server A if necessary.

4. **Start LifeKeeper, if necessary.** If LifeKeeper is not running on Server A, go to the **Windows Services tool**, select **LifeKeeper** and click **Start**. This will automatically start the **LifeKeeper External Interfaces** service.

5. **Move application back to Server A.** At a convenient time, use the LifeKeeper GUI to bring the application back into service on Server A. See Bringing a Resource In Service for instructions. Note that this step may be unnecessary if the application on Server A was configured for Automatic Switchback.

Removing LifeKeeper

Before Removing LifeKeeper

Included below are the requirements for removing LifeKeeper software.

1. **Move or stop applications.** Before removing the software, verify that applications requiring LifeKeeper protection are not on the server. Never remove LifeKeeper from a server where an application resource hierarchy is in service. Removing LifeKeeper removes all configuration data, such as equivalencies, resource hierarchy definitions and log files. See Transferring Resource Hierarchies for additional information.

2. **Ensure LifeKeeper is running.** LifeKeeper Recovery Kits may require LifeKeeper to be running when you remove the recovery kit software. Use the **Services MMC** snap-in to ensure that LifeKeeper services are running. If it is not running, the removal process cannot remove the resource instances from other LifeKeeper servers in the cluster which would leave the servers in an inconsistent state.

3. **Remove resource hierarchies.** Unextend or delete any resource hierarchies from the server where LifeKeeper will be removed. Never remove a Recovery Kit from a server where the resource hierarchy is in service. This will corrupt current hierarchies and they will need to be recreated when reinstalling the Recovery Kit.

4. **Remove all packages.** If removing the LifeKeeper core, first remove other packages that depend upon LifeKeeper; for example, LifeKeeper Recovery Kits. It is recommended that before removing a LifeKeeper Recovery Kit, first remove the associated application resource hierarchy.

Removing or Reinstalling LifeKeeper

To remove LifeKeeper software, run **Add/Remove Programs** from the **Windows Control Panel**. Select **LifeKeeper** and click **Remove**. InstallShield offers the following options:

- **Modify** - Select this option to add or remove individual LifeKeeper components.
- **Repair** - Select this option to reinstall all LifeKeeper components that were installed in the previous setup (not recommended).
- **Remove** - Select this option to remove all installed LifeKeeper components.

Notes

- **Important**: Uninstallation of LifeKeeper software requires that the Microsoft Visual C++ 2005 Redistributable package be installed. Do not remove this package until LifeKeeper has been uninstalled.
- **Modify** or **Repair** must be run from the LifeKeeper setup program.

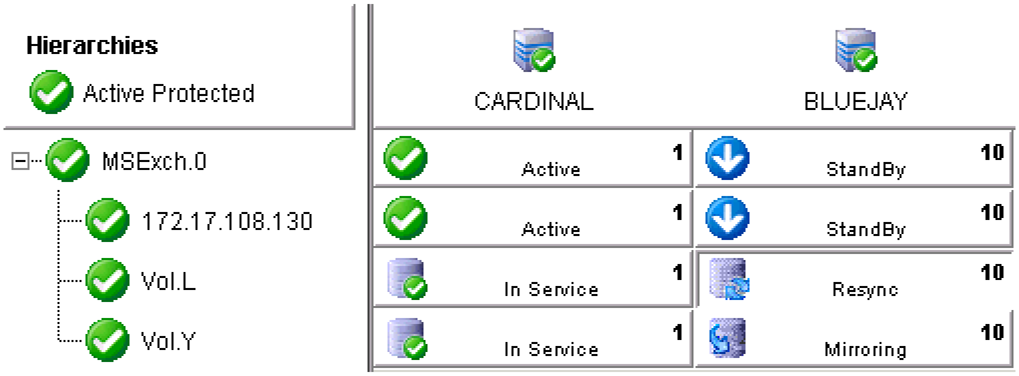
- Removal of LifeKeeper does NOT remove SUperior SU. SUperior SU can be removed separately using **Add/Remove Programs**.
- Removal of LifeKeeper may NOT delete the LifeKeeper directory. This directory can be deleted manually after the **Add/Remove** operation is complete.
- A reboot of the system is required to completely remove LifeKeeper remnants.

Data Replication

Monitoring Replicated Volume Resources

The state of all LifeKeeper protected replicated volume resources is displayed in the LifeKeeper GUI. Refer to the SteelEye DataKeeper topic, Mirror State Definitions, for details on mirror states.

The example below shows that the mirror state of the replicated volume resource Vol.L is **Resync** and that the mirror state of the replicated volume resource Vol.Y is **Mirroring**.

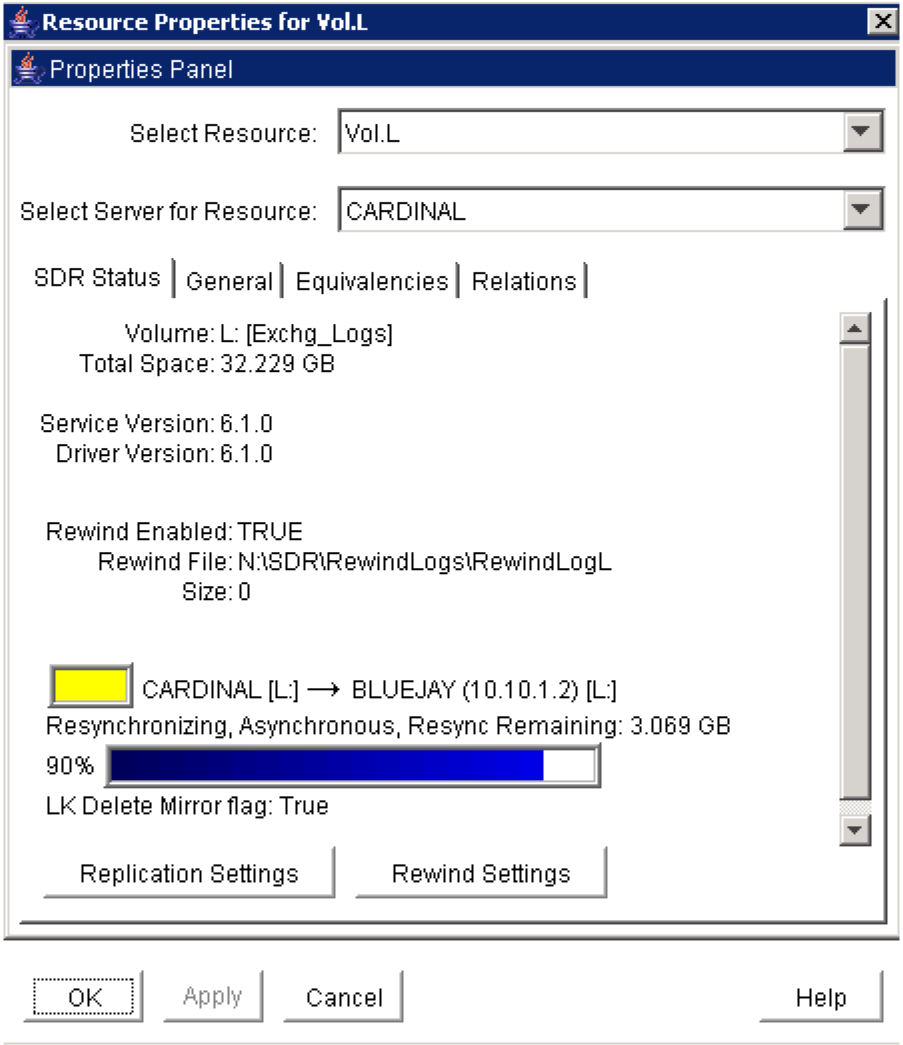


Hierarchies ✓ Active Protected	CARDINAL		BLUEJAY	
✓ MSExch.0	✓ Active	1	StandBy	10
✓ 172.17.108.130	✓ Active	1	StandBy	10
✓ Vol.L	In Service	1	Resync	10
✓ Vol.Y	In Service	1	Mirroring	10

The table below describes the different states for replicated volume resources and their meaning.

Resource State	Visual State	What it Means
Active	✓	Resource is operational on the primary server and protected (ISP)
Degraded	⚠	Resource is operational on the primary server but not protected by a backup resource (ISU)
Unknown	?	Resource has not been initialized (ILLSTATE) or LifeKeeper is not running on this server.
Failed	✖	Problem with resource detected on this server. For example, an attempt to bring the resource in-service failed. (OSF)
Offline	🔒	Resource is out of service on this server. Volume is not accessible for read/write operations.
Resync Pending	🕒	Resource state on the backup server is **Resync Pending**.
Mirroring	🔄	Resource state on the backup server is **Mirroring**.
Paused	⏸	Resource state on the backup server is **Paused**.
Resync	🔄	Resource state on the backup server is **Resync**.
Broken	✖	Resource state on the backup server is **Broken**.

To view the configuration information for a replicated volume resource from the LifeKeeper GUI, right-click on the volume resource and select **Properties**, then select the **SDR Status** tab. The example below shows Vol.L is the source on Cardinal and has one target - 10.10.1.2, and that it is resycning to the target Bluejay.


Resource Properties for Vol.L
Properties Panel

Select Resource: Vol.L

Select Server for Resource: CARDINAL

SDR Status | General | Equivalencies | Relations

Volume: L: [Exchg_Logs]
Total Space: 32.229 GB

Service Version: 6.1.0
Driver Version: 6.1.0

Rewind Enabled: TRUE
Rewind File: N:\SDR\RewindLogs\RewindLogL
Size: 0

🟨 CARDINAL [L:] ⟶ BLUEJAY (10.10.1.2) [L:]
Resynchronizing, Asynchronous, Resync Remaining: 3.069 GB
90%
LK Delete Mirror flag: True

Replication Settings
Rewind Settings
OK
Apply
Cancel
Help

Replication Settings

From the **Volume Resource Properties** page, select the **Replication Settings** button to set the compression level, the network throttling or the LifeKeeper Delete Mirror flag for a replicated volume.

Field	Tips
Select Targets	Select the target server to which the action should be applied.
Set Compression Level	Specify the compression level for the selected replicated volume. Valid values are 0 to 9. Level 0 is "no compression". Values from1 to 9 specify increasingly CPU-intensive levels of compression. Compression level 1 is a "fast" compression - it does not require as much CPU time to compress the data but results in larger (less compressed) network packets. Level 9 is the maximum amount of compression - it results in the smallest network packets but requires the most CPU time. The level can be set to somewhere in between to balance CPU usage and network efficiency based on your system, network and workload. Default is 0.
Set Network Throttling	The Bandwidth Throttle setting (specified in kilobits per second) limits the amount of network bandwidth that the replicated volume can use for resync and normal volume writes. Default is 0.
Set LifeKeeper Delete Mirror Flag	The LifeKeeper Delete Mirror Flag controls the behavior during delete of the LifeKeeper resource for the replicated volume. When deleting the LifeKeeper volume resource, if the flag is set to **True**, then LifeKeeper will delete the mirror; otherwise, the mirror will remain. Select **True** if you want the mirror deleted when the volume resource is unextended or removed from LifeKeeper. Select **False** if you want the mirror to remain intact. Default is **True** if mirror is created using LifeKeeper GUI. Default is **False** if mirror is created outside of LifeKeeper GUI.

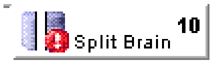
Performing Actions on Replicated Volumes

To perform actions on a replicated volume resource using the LifeKeeper GUI, right-click on the replicated volume resource and select the action you wish to perform from the context menu. If you have the **Properties Panel** enabled (View->Properties Panel), the resource toolbar will be displayed for the selected volume.

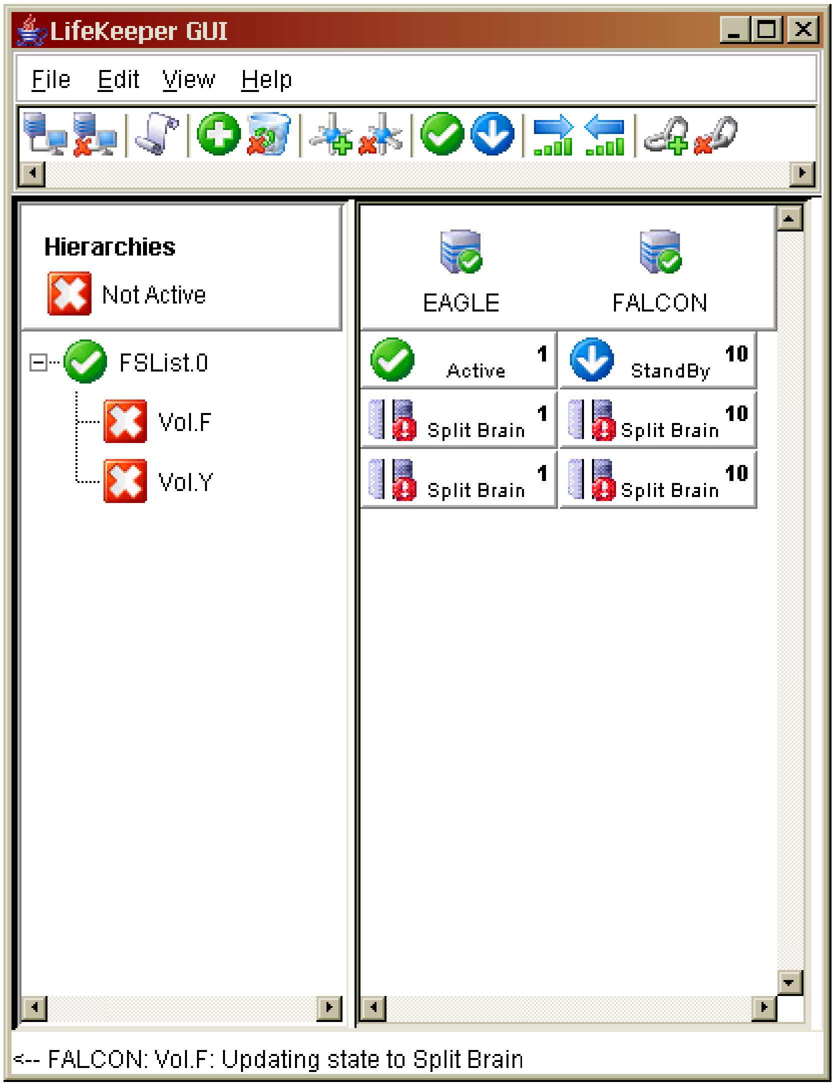
Action	Icon	Meaning
Select Target		Select the target system to which the action should be applied.
Pause Mirror	⏸	Select **Pause Mirror** to temporarily stop the data from being mirrored. A partial resync will be performed when you click **Continue** to un-pause the mirror. After pausing a mirror, it is possible to unlock the target volume using the **Unlock Target** action.
Continue Mirror/Lock Target	▶	To continue a mirror after the mirror has been paused, select the **Continue/Lock Target** action. This un-pauses the mirror, re-locks the target volume (if unlocked) and resumes the mirroring process. While pause temporarily stops the writes from being mirrored, the writes are recorded during the pause interval. When the mirror is resumed, the recorded writes are sent to the target volume and the mirror is automatically re-synchronized (partial resync).
Unlock Target	🔓	To unlock the target volume of a mirror, select the **Unlock Target** action. This pauses the mirror (if not already paused) and unlocks the mirrored volume on the target system. This allows read/write access to the data on the volume. **Continue Mirror** will relock the target volume, perform a partial resync and resume the mirroring process. **Warning**: Do not write to the target volume while the mirror is unlocked! Any writes to the target while the mirror is unlocked will be lost when the mirror is re-synchronized.
Break Mirror/Unlock Target		Breaking a mirror discontinues the mirror for the selected volumes and unlocks the target volume but does not remove the mirror from the volume list. A full resync must be performed in order to re-establish the mirror after a **Break Mirror/Unlock Target** action. **Warning**: Do not write to the target volume while the mirror is broken! Any writes to the target while the mirror is broken will be lost when the mirror is re-synchronized.
Resync Mirror/Lock Target	🔄	To re-establish a broken mirror, select **Resync Mirror/Lock Target** action. A full resync will be performed.
Rewind Target Volume	🕒	This wizard rewinds the data on the target system until you reach a good data set and then walks you through the recovery process.
Add Rewind Log Bookmark		This wizard guides you through the process of viewing bookmarks and adding bookmarks to the rewind log. 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.

What is Split-Brain

When all of LifeKeeper's comm paths are disconnected and if **Automatic Node Failover** is enabled, each side of LifeKeeper assumes that the other side is dead and attempts to bring all the resources in service. In the case of a SteelEye DataKeeper resource, both sides become mirror sources and allow data to be written to the volume. This condition is defined as "split-brain" and will be indicated in the LifeKeeper GUI with the following icon:

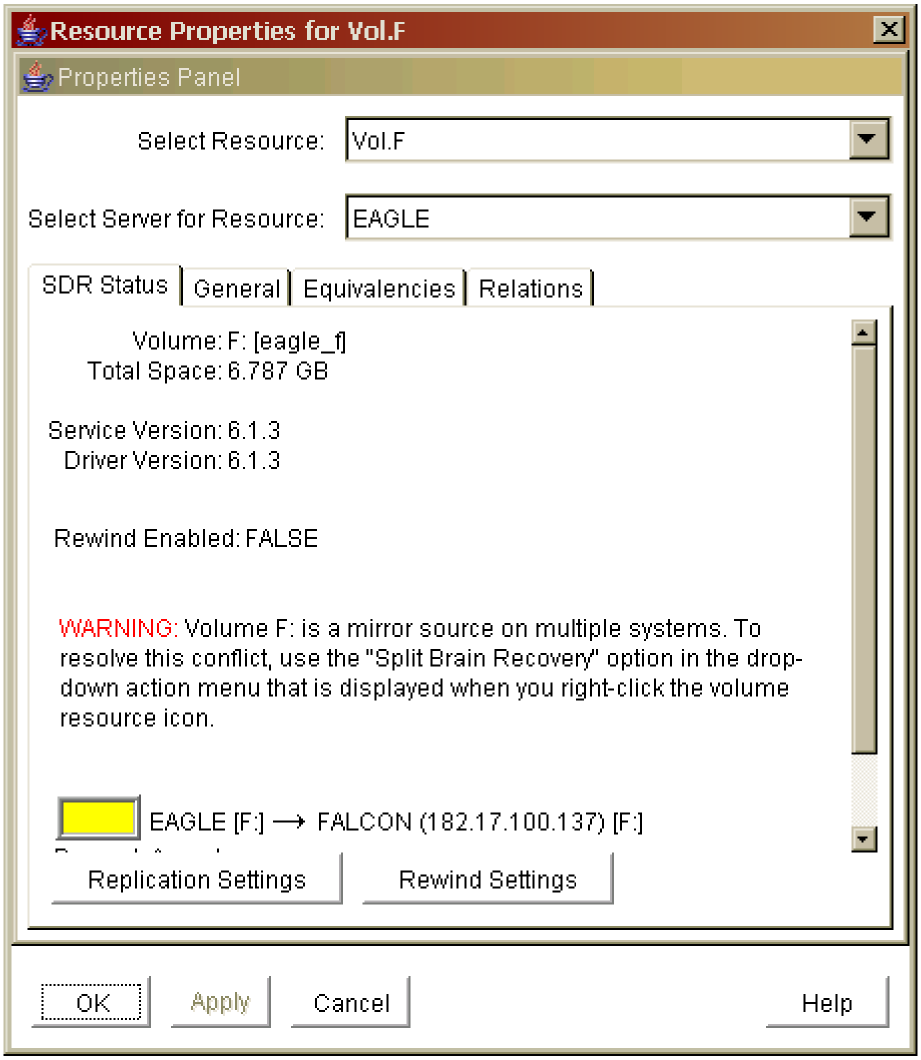


Refer to the topic Split Brain Recovery for the steps required to resolve this situation.



<-- FALCON: Vol.F: Updating state to Split Brain

The **Properties Panel** for the selected volume displays additional information about the split-brain condition and instructions for resolving this problem.



Resource Properties for Vol.F

Properties Panel

Select Resource: Vol.F

Select Server for Resource: EAGLE

Volume: F: [eagle_f]
Total Space: 6.787 GB

Service Version: 6.1.3
Driver Version: 6.1.3

Rewind Enabled: FALSE

WARNING: Volume F: is a mirror source on multiple systems. To resolve this conflict, use the "Split Brain Recovery" option in the drop-down action menu that is displayed when you right-click the volume resource icon.

EAGLE [F:] ⟶ FALCON (182.17.100.137) [F:]

Split Brain Recovery

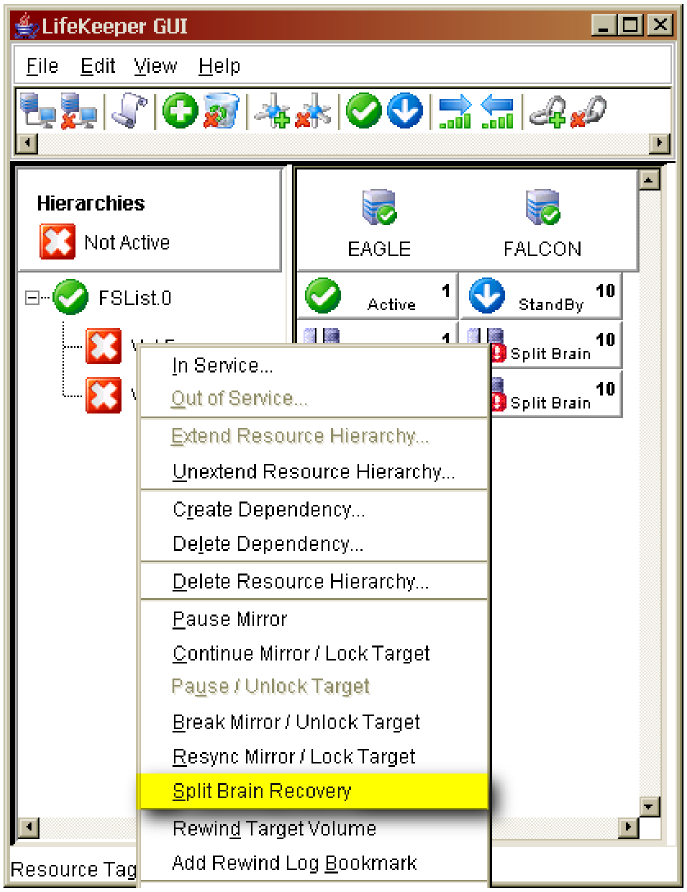
After the system's comm paths have been restored and the servers detect that the volumes are in the Split-Brain state, you will have to perform the **Split-Brain Recovery** procedure below.

Note: If multiple volumes are detected in different resource hierarchies, you will have to perform the Split-Brain Recovery procedure on each volume. Split-Brain volumes that are in the same hierarchy will be recovered together.

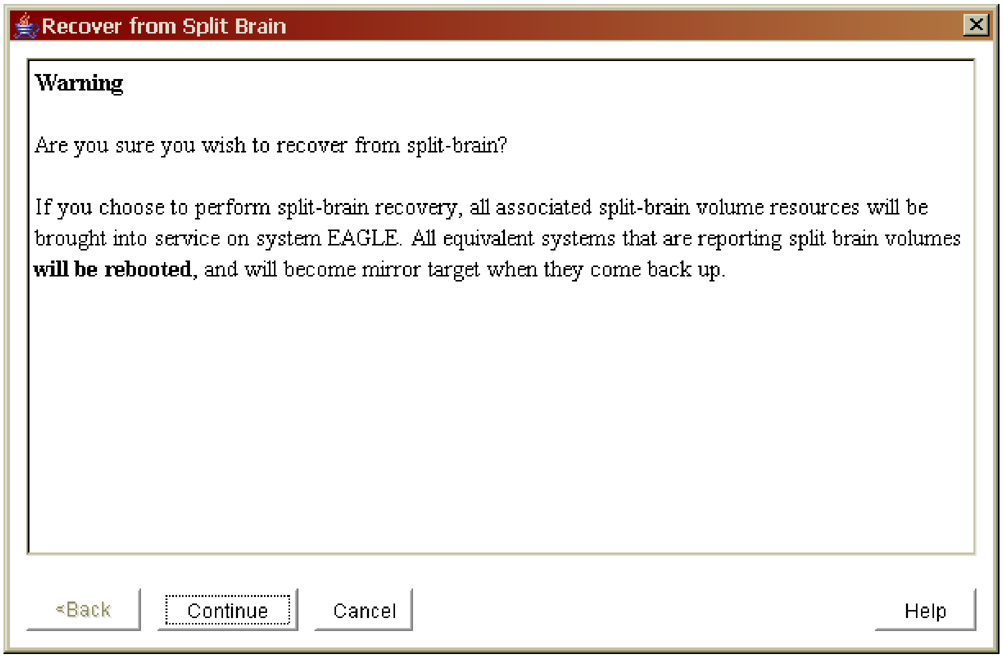
1. Right-click on the volume instance icon under the system that will be the source. The **Resource Context** menu displays. You can also right-click on the volume instance icon in the **Hierarchies** list in the far left panel. Choose **Split-Brain Recovery** from the menu, and you

will be prompted to select which server should be the mirror source.

2. Select **Split-Brain Recovery** from the menu.



3. The following warning message will display. Select the **Continue** button to complete the Split-Brain Recovery process. **Note**: During this procedure, all other systems that are in the split-brain condition will be rebooted and will become the mirror target when they complete the rebooting process.

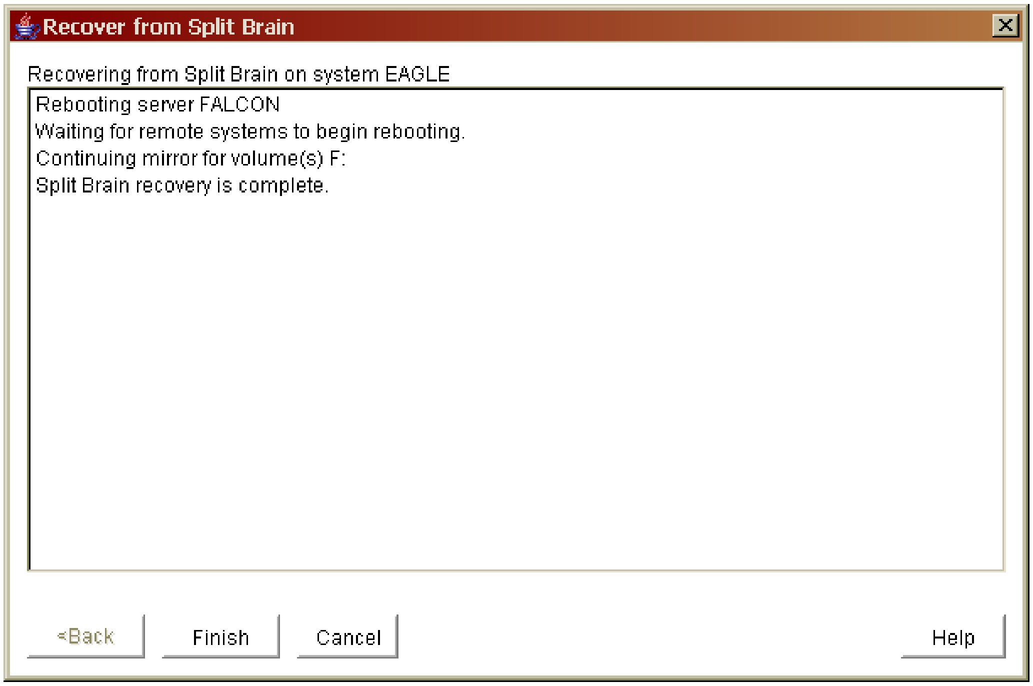


Warning

Are you sure you wish to recover from split-brain?

If you choose to perform split-brain recovery, all associated split-brain volume resources will be brought into service on system EAGLE. All equivalent systems that are reporting split brain volumes **will be rebooted**, and will become mirror target when they come back up.

4. The following message will display as the Split-Brain Recovery process proceeds. There will be a delay while the remote systems are rebooted. Select **Finish** to complete.

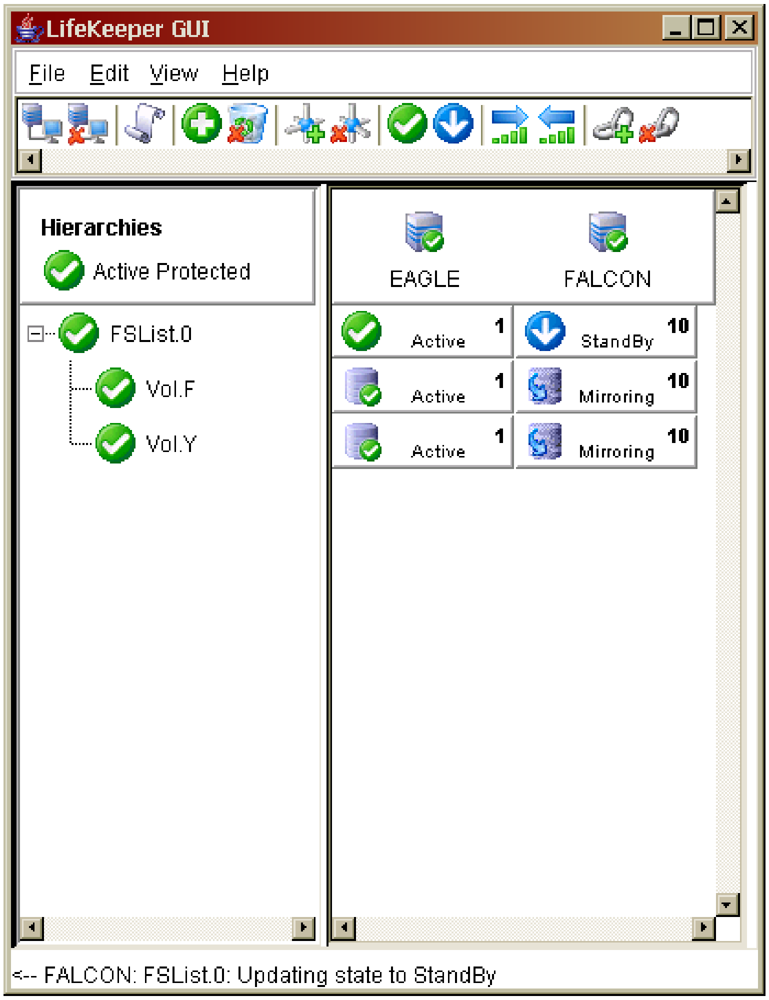


Recovering from Split Brain on system EAGLE

```
Rebooting server FALCON
Waiting for remote systems to begin rebooting.
Continuing mirror for volume(s) F:
Split Brain recovery is complete.
```

5. Once the recovery is complete, the recovered resources will appear in the GUI as follows. The Split-Brain Recovery process will be completed when the target system has rebooted and the

mirror will be re-synced.



<-- FALCON: FSList.0: Updating state to StandBy

Data Rewind Overview

Data Rewind allows administrators to change the contents of a replicated volume to any point in time using rewind and advance operations. This capability is made possible through the **Rewind Log** which records all data changes that occur on the volume. As long as a timestamp is contained in the Rewind Log, the volume can be rewound to that point in time and can then be rewound further back in time or advanced forward in time.

The Rewind Log is created and maintained on a mirror Target system. All rewind and advance operations are performed on the Target to ensure minimal impact on the Source (primary) server. By default, rewind is disabled on all replicated volume resources. From the **Volume Resource Properties** page, select the **Rewind Settings** button to enable rewind for the selected replicated volume resource.

For optimal performance, the rewind log file should be on a separate physical disk. It is also recommended that write caching be enabled for the physical disk where the log is located and for the physical disk where the replicated data is located. To enable write caching, go to **My Computer ->right-click Volume -> Properties -> Hardware tab -> Properties ->Policies tab -> Enable write caching on the disk**.

If the Rewind Log file is to be stored at a different location than the default, this change must be made prior to enabling rewind for the selected replicated volume resource. From the **Volume Resource Properties** page, select the **Rewind Settings** button to change the location of the rewind log for the selected replicated volume resource.

To save space on your rewind log disk, you can choose to enable NTFS compression for the rewind log file. NTFS compression must be enabled prior to enabling rewind for the selected replicated volume resource. From the **Volume Resource Properties** page, select the **Rewind Settings** button to enable NTFS compression for the rewind log file.

After performing rewind/advance operations and locating good data, there is one recovery option available:

- Manually copy data to the Source (primary) server from the Target (backup) server. After manual copy is complete, a partial resync to the Target will be performed.

Rewind Settings

From the **Volume Resource Properties** page, select the **Rewind Settings** button for enabling/disabling rewind, changing the rewind log location, associating volumes that should be rewound at the same time (as a single entity), setting the maximum age limit for the log file or setting NTFS compression for the selected replicated volume resource.

Field	Tips
Enable / Disable Rewind	Specifies whether to enable or disable rewind for the selected volume resource. Rewind is disabled by default.
Rewind Logfile Location	Specifies the folder where the rewind log file for this volume should be stored. The folder must not be on any volume that is part of a mirror or that is protected by LifeKeeper - it must be on a volume that is writable whenever this volume is a mirror Target. Default is `%extmirrbase%\RewindLogs`. **Note**: The location of the log file must be changed prior to enabling rewind.
Compressed Logfile	Specifies whether the rewind log file should be created using NTFS compression. Select **True** to compress the logfile. Default is **False**. **Note**: The **compress log file** option must be set prior to enabling rewind.
Associated Volumes	Specifies which volumes should be rewound at the same time as a single entity. Associated volumes are specified in comma separated groups. For example, if you have two applications, one that relies on both the D: and E: volumes and another which relies on the P: and Q: volumes, then enter: DE,PQ. **Note**: In most cases, it is not necessary to change this setting. As long as your LifeKeeper hierarchies contain the necessary volume resources that your applications depend on (which is normally the case), the rewind procedure will automatically associate those volumes when performing a rewind. You only need to override the default behavior if you have special volume associations that cannot be determined automatically by looking at the LifeKeeper resource hierarchies.
Max Logfile Size	Specifies the maximum size, in MB, that this volume's rewind log will be allowed to grow to. A value of 0 specifies unlimited file size. Default is 0.
Max Logfile Age	Specifies the maximum number of minutes of data to be stored in the rewind log. A value of 0 specifies unlimited age. Default is 0.
Min Log Volume Free Space	Specifies the minimum amount of space in megabytes that should be reserved on the Rewind log file volume to avoid filling up the volume and running out of space. Default is 100MB.

Performing Rewind on Replicated Volume Resources

To perform a rewind operation on a replicated volume resource using the LifeKeeper GUI, right-click on the replicated volume resource and select **Rewind Target Volume** from the **context menu**. If you have the **Properties** panel enabled **(View->Properties Panel)**, select the target of replicated volume resource to be rewound and click on ⏱ on the resource toolbar. This will start the rewind operation.

The displayed **Warning** message will show the volume or volumes that will be rewound during the rewind operation. The target volume(s) will be paused and unlocked as part of the rewind operation. During the time that the mirrors are paused, the target server will not be eligible to provide LifeKeeper recovery for the volumes. While data is being rewound on the target, all changes to the source will be tracked and written to the target by performing a **Continue Mirror** action. Click on **Continue** to proceed with the rewind operation or **Cancel** to exit the rewind operation. The table below shows the information that is required to perform the rewind operation.

Field	Tips
Stop all applications on <Source> that are using volumes <driverletter(s)>?	Select **Yes** if you want LifeKeeper to stop all applications running on the primary <Source> server that are using volumes specified by <driveletter(s)> as part of their LifeKeeper hierarchy. Select **No** to leave applications running on the primary <Source> server. Default is **No**.
Prepare to Rewind and Recover data for <volume tag> on<target>	After LifeKeeper completes the process of preparing the volume(s) identified by <volume tag> for rewind, click on **Next** to continue the rewind operation.
Specify Rewind Point	Choose a rewind timestamp from the list provided or type in the start time you want to use. There are many date and time formats accepted for this value including: 5/4/07 08:27 April 29 23:00 last Saturday 13:21 yesterday 08:23:17 The time string that you enter will be validated before being used. The progress bar at the bottom of the dialog indicates the state of the data in the rewind log as it is currently known. At the beginning of the rewind process, all data is unknown. As the data is rewound and advanced to various points in time, sections of the data are marked as good or bad depending on your test results with the various data sets.
Perform rewind	LifeKeeper is now rewinding the data on the target server to the rewind point specified from above. Click **Next** to continue with the rewind operation.
Please enter your comments on the test results; Comments are optional	Evaluate the test results and enter any comments here. Comments will be stored and displayed in the list of **Rewind Points** to help you locate good data.
Is the data valid?	Indicate whether the volume contents are now valid. The answer you provide will be used to construct the list of timestamps that you are given for future rewind/advance operations. LifeKeeper tries to guide you to the latest known good data by limiting the start and end timestamps that are listed based on your answers to this question. If you choose **Yes**, you will not be given any choices that are earlier than the current selected timestamp. If you choose **No**, you will not be given any choices that are later than the current timestamp. If you are unsure whether the data is valid, choose **Not Sure** and the rewind points that you see will not be limited.

Field	Tips
Processing result for rewind time below	Storing evaluation and comments and adjusting Rewind Points interval. Click **Next** to continue with the rewind operation.
Proceed with recovery or try a different data set. Select **Next Action**	You may choose to rewind or advance to a different timestamp or may stop rewinding and use the current state of the volume(s) for recovery. The progress bar at the bottom of the dialog indicates the state of the data in the rewind log as it is currently known. At the beginning of the rewind process, all data is unknown. As the data is rewound and advanced to various points in time, sections of the data are marked as good or bad depending on your test results with the various data sets.
Data Recovery	To recover lost or corrupt data, you can leave applications running on the Source (primary) server or stop them as necessary. Manually copy any missing or corrupt data from the rewound Target (backup) server to the Source (primary) server. After copying has been completed, click on **Next**. The rewound volumes will be re-locked and the mirrors continued. This completes the rewind operation and data recovery operation.

Chapter 6: Troubleshooting

Applet Troubleshooting

Description

If the web client does not display the Cluster Connect dialog, try the following:

1. Check whether the applet failed. Usually a message is printed somewhere in the browser window specifying the state of the applet. In Netscape and Internet Explorer, an icon may appear instead of the applet, in addition to some text status. Clicking this icon may bring up a description of the failure.

2. Open the Java Console.

- For FireFox, Netscape and older versions of Internet Explorer, run the **Java Plug-In applet** from your machine's **Control Panel** and select the option to show the console, then restart your browser.

- For recent versions of Internet Explorer, select **Tools > Sun Java Console**. If you do not see the Sun Java Console menu item, select **Tools >Manage Add-Ons** and enable the console, after which you may need to restart your browser before the console will appear.

- For Mozilla, select **Tools > Web Development > Sun Java Console.**

3. If the web client is not open, reopen the URL *http://<server name>:81* to start it.

4. Check the console for any messages. The messages should help you resolve the problem. If the problem appears to be network related, refer to the Network-Related Troubleshooting section.

CYGWIN Crash on Win2008 May Occur When LifeKeeper Stopping

Symptom

CYGWIN crash on Win2008 (e.g. GREP segmentation fault) may occur when LifeKeeper is stopping.

With Terminal Service enabled as well as Data Execution Prevention enabled for the application, there is a bug in Win2008 in tsappcmp.dll which occasionally causes the crash to happen.

Solution

There is a page of memory whose protection level gets changed by tsappcmp.dll, and the "EXECUTE" permissions on that page get removed for some unknown reason. The page contains executable code and should be able to be executed, but since the permission was modified, an exception occurs and the program crashes.

See the following articles for additional information and a workaround solution.

http://www.eggheadcafe.com/software/aspnet/33380656/corinnayou-wrote-that-th.aspx
http://www.mail-archive.com/cygwin@cygwin.com/msg91569.html

Error When Attempting to Run LifeKeeper Command From Command Prompt

Symptom

When attempting to run a LifeKeeper command from a command prompt, you receive the following error:

```
[File:lock.CLine:1610] Win32 Error: 2
 *CRITICAL* (No. 472) Can’t run this application without LCD Daemon
running.
```

Solution

LifeKeeper commands require “console” rights to run. If using Remote Desktop, invoke Remote Desktop with the “/console” switch when the LifeKeeper core is running on Server 2003. When the LifeKeeper core is running on Server 2008, invoke Remote Desktop connections with the “/admin” switch.

e.g. `%SystemRoot%\system32\mstsc.exe/console`

You may also run the LifeKeeper command from the command prompt on the LifeKeeper system itself.

Firewall

Symptom

Firewall was disabled during installation but is now enabled. How do I add the rules to Windows firewall?

Solution

There is a script in the LifeKeeper directory that will allow you to enable the firewall rules. The script is located in:

```
<LifeKeeper Root Directory>\support\firewallsetup.bat
```

By opening a command prompt and executing firewallsetup.bat <LifeKeeper Root directory>, you can add the rules. If the rules had been added, the script will not add duplicate rules.

If you open the Windows firewall (wf.msc), you will see the inbound rules prefixed with the LifeKeeper label.

Note: If you have created specific rules that have disabled the ports required by LifeKeeper, the installation program will disable but will not delete those rules.

GUI Error Messages

Description

Error 101: Illegal argument was passed.

Error 102: This program requires a Java Virtual Machine Version 1.5 or greater to run properly. Please refer to the LifeKeeper GUI documentation to verify your setup.

Error 103: Could not set Look and Feel for LifeKeeper GUI.

Error 104: <filename> Image could not be loaded.

Error 106: Error trying to get data over RMI. Could not complete action.

Error 107: Failed to create Global Resource Instance.

Error 108: Failed to create Global Resource.

Error 109: Dialog requires a Server to be selected.

Error 112: Could not match Resource Instance to Global Equivalency.

Error 114: <server name> Security Exception caused connection failure to this server. Please note that this failure may result in degraded cluster resource rendering. Verify that your Java Policy file is installed properly. See Running the LifeKeeper Web Client.

Error 115: <server name> Name of this server could not be resolved resulting in a connection failure to this server. Please note that this failure may result in degraded cluster resource rendering. Please refer to the LifeKeeper GUI documentation to verify network naming conventions. See Unknown Host Exception.

Error 116: <server name> This server could not resolve the name of this client host resulting in a connection failure to this server. Please note that this failure may result in degraded cluster resource rendering. Please refer to the LifeKeeper GUI documentation to verify network naming conventions. See Unknown Host Exception.

Error 117: Initial connection to server failed. LifeKeeper GUI will continue trying to connect. Please refer to the LifeKeeper GUI documentation to verify that LifeKeeper and the LifeKeeper GUI server are active on this system. See Java RMI Binding Problem.

Error 118: Incompatible client and server packages caused connection failure. Please verify that the versions are compatible between the target server and the server from which the client was started.

Error 119: Could not export remote object.

Error 120: Encountered exception when invoking remote method.

Error 121: Administrative java bean could not be initialized.

Error 122: Administrative java bean has no dialog classes to load. The properties file describing the administrative java bean is missing the "list" property.

Error 123: The properties file describing the administrative java bean has a missing property.

Error 124: Failed to find property bundle.

Error 125: Security Exception trying to create URLClassLoader. Please verify that the .java.policy file grants the proper permissions. You should typically create a .java.policy file in your home directory. The contents of the .java.policy file are case sensitive, so it is best to copy the sample file that is distributed with the LifeKeeper GUI package.

If you are using a browser plug-in for Java, then the user home directory that is being used for the java environment can be verified by enabling the Java console and examining the first few lines that are displayed. Refer to Configuring the LifeKeeper GUI for more information on configuring the GUI client.

Error 126: Could not find resource on server.

Error 127: Could not find extend properties file for this kit.

Error 128: Internal properties file error.

Error 129: Cannot establish an RMI connection to the server. Verify the LifeKeeper GUI Server is running on the server.

Error 130: The tag entered is being used by another resource. Please enter another tag.

Error 131: Exception calling invokeAndWait method to update the user interface.

Error 132: Encountered exception when invoking administrative java bean.

Error 133: Invalid value entered for equivalency priority. The priority value must be in the range of 1 through 999.

Error 134: The equivalency priority value conflicts with another priority in the table. Each equivalency priority value must be unique in the table.

GUI Network Related - Initial Connection to Server Failed (Error 117)

Symptom

Initial Connection to server failed (Error 117).

If you are attempting to connect to a server that has two or more network interface cards (NICs), it could indicate a Java RMI binding problem where the first NIC (the one that appears first in the output of ipconfig utility) has a non-reachable IP address.

Solution

You may need to reorder the protocol binding for use by the network services of the LifeKeeper server. On each LifeKeeper server, open "Network and Dial-up Connections", and on the **Advanced** menu, select **Advanced Settings**. The **List Box** at the top of the dialog shows the current order of the NIC cards. Click the **arrow button** to reorder them so that the reachable NIC is at the top of the list. This should enable Java RMI to allow client to connect to the server. A reboot of the server is required for this to take effect.

GUI Network Related - Long Connection Delays on Windows Platforms

Symptom

Long Connection Delays on Windows Platforms.

Solution

From Sun FAQ:

“Most likely, your host’s networking setup is incorrect. RMI uses the JavaAPI networking classes, in particular java.net.InetAddress, which will cause TCP/IP host name lookups for both host to address mapping and address to hostname. On Windows, the lookup functions are performed by the native Windows socket library, so the delays are not happening in RMI but in the Windows libraries. If your host is set up to use DNS, then this could be a problem with the DNS server not knowing about the hosts involved in communication and what you are experiencing are DNS lookup timeouts. If this is the case, try specifying all the hostnames/addresses involved in the *local file\winnt\system32\drivers\etc\hosts* or *\windows\hosts*. The format of a typical host file is:

```
IPAddress Server Name

e.g.: 208.2.84.61 homer.somecompany.com
```

This should reduce the time it takes to make the first lookup.”

In addition, incorrect settings of the Subnet Mask and Gateway address may result in connection delays and failures. Verify with your Network Administrator that these settings are correct.

GUI Network Related - NoRouteToHostException Message Generated During Connection Attempt

Symptom

NoRouteToHostException Message Generated During Connection Attempt.

A socket could not be connected to a remote host because the host could not be contacted.

Solution

Typically, this indicates that some link in the network between the local and remote server is down or that the remote server is behind a firewall.

GUI Network Related - Unknown Host Exception Message Generated During Connection Attempt

Symptom

Unknown Host Exception Message Generated During Connection Attempt.

The LifeKeeper GUI Client and Server use Java RMI (Remote Method Invocation) technology to communicate. For RMI to work correctly, the client and server must use resolvable hostname or IP addresses. When unresolvable names, WINS names, or unqualified DHCP names are used, this causes Java to throw an UnknownHostException.

This error message may also occur under the following conditions:

- Server name does not exist. Check for misspelled server name.
- Misconfigured DHCP servers may set the fully qualified domain name of RMI servers to be the domain name of the resolver domain instead of the domain in which the RMI server actually resides. In this case, RMI clients outside the server's DHCP domain will be unable to contact the server because of the incorrect domain name.
- The server is on a network that is configured to use Windows Internet Naming Service (WINS). Hosts that are registered under WINS may not be reachable by hosts that rely solely upon DNS.
- The RMI client and server reside on opposite sides of a firewall. If your RMI client lies outside a firewall and the server resides inside of it, the client will not be able to make any remote calls to the server.

Solution

When using the LifeKeeper GUI, the hostname supplied by the client must be resolvable from the server and the hostname from the server must be resolvable by the client. The LifeKeeper GUI catches this exception and alerts the user. If the client cannot resolve the server hostname, this exception is caught and Message 115 is displayed. If the server cannot resolve the Client hostname, this exception is caught and Message 116 is displayed. Both of these messages include the part of the Java exception which specifies the unqualified hostname that was attempted.

Included in the following sections are some procedures that may be used to test or verify that hostname resolution is working correctly.

From Windows

1. Verify communication with the LifeKeeper server. From a prompt, ping the target using the hostname:

   ```
   ping<TARGET_NAME>
   ```

   For example;

   ```
   ping homer
   ```

   A reply listing the target's qualified hostname and IP address should be seen.

2. Verify proper configuration.
   a. Check configuration of DNS or install a DNS server on your network.
   b. Check the settings for **ControlPanel->Network->Protocols->TCP/IP**. Verify with your Network Administrator that these settings are correct. Note that the hostname in the DNS tab should match the name used on the local name server. This should also match the hostname specified in the GUI error message.
   c. Try editing the hosts file to include entries for the local host and the LifeKeeper servers that it will be connected to.

   On Windows 2003/2008 systems, the hosts file is:

   ```
   %SystemRoot%\system32\drivers\etc\HOSTS (e.g.
   C:\windows\system32\drivers\etc\HOSTS)
   ```

   Note: On Windows 2003/2008, if the last entry in the hosts file is not concluded with a carriage-return/line-feed, then the hosts file will not be read at all.

   For example, if my system is called HOSTCLIENT.MYDOMAIN.COM and uses the IPaddress 153.66.140.1, add the following entry to the hostsfile:

   ```
   153.66.140.1 HOSTCLIENT.MYDOMAIN.COM
   ```

3. Try setting the hostname property to be used by the GUI client. To do this from a browser with the Plug-in, open the **Java Plug-In Control Panel** and set the host name for the client by adding the following to "Java Run Time Parameters."

   ```
   -Djava.rmi.server.hostname=<MY_HOST>
   ```

4. Check for Microsoft network-related patches at www.microsoft.com.

From Linux

1. Verify communication with the other server by pinging the target server from Linux using its hostname or IP address:

   ```
   ping -s<TARGET_NAME>
   ```

   For example:

   ```
   ping -s homer
   ```

   A reply listing the target's qualified hostname should be seen.

2. Verify that *localhost* is resolvable by each server in the cluster using ping with its hostname or IP address. If DNS is not implemented, edit the */etc/hosts* file and add an entry for the *localhost* name. This entry can list either the IP address for the local server or it can list the default entry (127.0.0.1).

3. Check that DNS is specified before NIS. DNS should be put before NIS in the host's line of */etc/nsswitch.conf*, and */etc/resolv.conf* should point to a properly configured DNS server(s).

4. If DNS is not to be implemented or no other method works, edit the */etc/hosts* file to add an entry for the hostname.

5. Try setting the hostname property to be used by the GUI client. This will need to be changed for each administrator.

   To do this from a browser with the Plug-in, open the **Java Plug-In Control Panel** and set the hostname for the client by adding the following to **Java RunTime Parameters**:

   ```
   -Djava.rmi.server.hostname=<MY_HOST>
   ```

   To do this from the HotJava browser, append the following to the hotjava command line:

   ```
   -Djava.rmi.server.hostname=<MY_HOST>
   ```

   For Example:

   ```
   -Djava.rmi.server.hostname=153.66.140.1
   ```

   ```
   -Djava.rmi.server.hostname= homer.somecompany.com
   ```

GUI Server Troubleshooting

Symptom

The LifeKeeper GUI uses Ports 81 and 82 on each server for its administration web server and Java remote object registry. If another application is using the same ports, the LifeKeeper GUI will not function properly.

Solution

These values may be changed by editing the following registry entries:

```
GUI_WEB_PORT=81
```

```
GUI_RMI_PORT=82
```

These entries are located in the following registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\SteelEye\LifeKeeper\JavaGUI\Server
```

Note: The port values are initialized in the GUI server when it is started. If you alter them, you will need to stop and restart the GUI server. These values must be the same across all clusters to which you connect.

Incomplete Resource Creation

Description

If the resource setup process is interrupted leaving instances only partially created, you must perform manual cleanup before attempting to install the hierarchy again. Use the LifeKeeper GUI to delete any partially-created resources. See Deleting a Hierarchy from All Servers for instructions. If the hierarchy list does not contain these resources, you may need to use the `ins_remove` and `dep_remove` to clean up the partial hierarchies.

Installation - Access is Denied

Symptom

During upgrade or re-installation, LifeKeeper generates “Access is denied” error message.

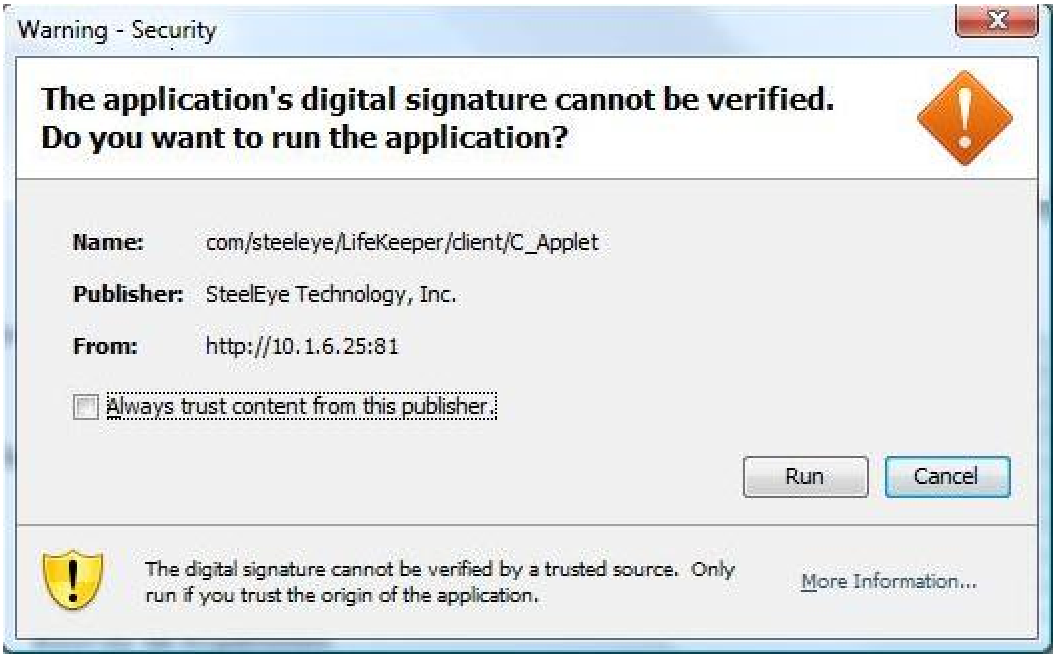
Solution

The LifeKeeper services have not all stopped. This can occur if Setup was unable to stop the LifeKeeper services. Open a command window and enter `$LKROOT\bin\lkstop` to stop all the LifeKeeper services, then wait until you see “**LIFEKEEPER NOW STOPPED**” before running Setup.

Java Mixed Signed and Unsigned Code Warning

Symptom

When loading the LifeKeeper Java GUI client applet from a remote system, the following security warning may be displayed:



Warning - Security

**The application's digital signature cannot be verified.
Do you want to run the application?**

Name: com/steeleye/LifeKeeper/client/C_Applet

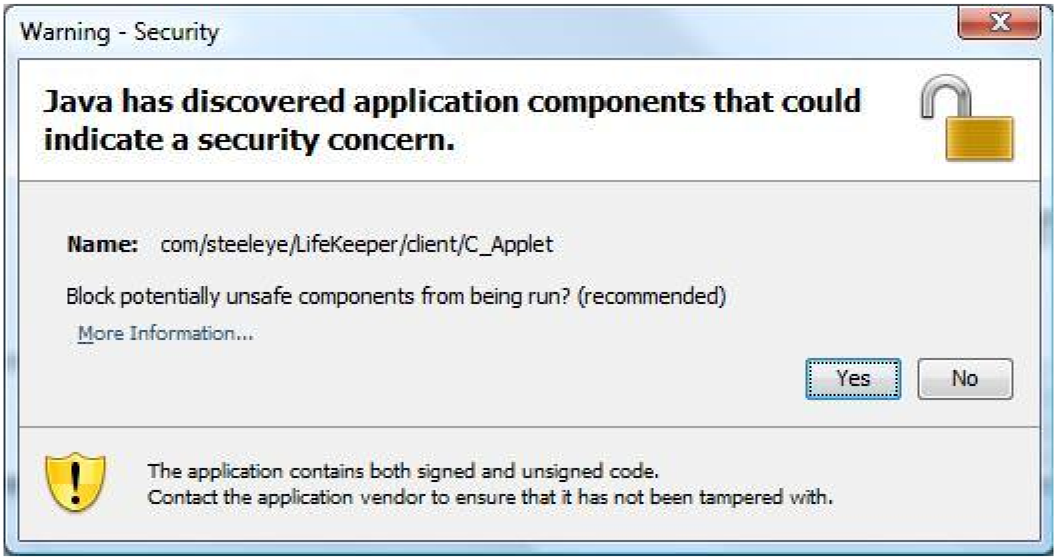
Publisher: SteelEye Technology, Inc.

From: http://10.1.6.25:81

☐ Always trust content from this publisher.

The digital signature cannot be verified by a trusted source. Only run if you trust the origin of the application.

Enter “**Run**” and the following dialog will be displayed:



Warning - Security

Java has discovered application components that could indicate a security concern.

Name: com/steeleye/LifeKeeper/client/C_Applet

Block potentially unsafe components from being run? (recommended)

The application contains both signed and unsigned code.
Contact the application vendor to ensure that it has not been tampered with.

Block? Enter “**No**” and the LifeKeeper GUI will be allowed to operate.

Solution

To reduce the number of security warnings, you have two options:

1. Check the “Always trust content from this publisher” box and select “Run”. The next time the LifeKeeper GUI Java client is loaded, the warning message will not be displayed.

   or

2. Add the following entry to your Java “deployment.properties” file to eliminate the second dialog about blocking. The security warning will still be displayed when you load the Java client, however, the applet will not be blocked and the Block “Yes” or “No” dialog will not be displayed. Please note this setting will apply to all of your Java applets.

   ```
   deployment.security.mixcode=HIDE_RUN
   ```

To bypass both messages, implement 1 and 2.

LANMAN Name May Be Seen Twice in Browse List

Symptom

After creating a LANMAN resource, the LANMAN name may be seen twice in the browse list.

Solution

One of these entries is an unusable Workstation record. Please disregard this entry and use the other LANMAN Server name.

Licensing - Licensed Recovery Kit Resource Fails to Come In Service

Symptom

After upgrade, licensed recovery kit resource fails to come in service and the following error is logged to the **Application Event Log** by LifeKeeper: “Process: lcdmachfail(3176) *ERROR* (No. 1001) resource <tag name> requires a license (for Kit <recovery kit type>) but none is installed.

Solution

Use the LifeKeeper licensing utility to install your recovery kit license key. See Obtaining and Installing the License for information on installing a license.

Licensing - License Key Not Found

Symptom

After installing licensed recovery kit, the following error is logged to the **Application Event Log** by LifeKeeper: “Process: Lkinit:(1832) *ERROR* (No. 20042) LifeKeeper Recovery Kit <licensed recovery kit> license key NOT FOUND”.

Solution

Use the LifeKeeper licensing utility to install your recovery kit license. See Obtaining and Installing the License for information on installing a license.

LifeKeeper Web Client May Lock Up

Symptom

The LifeKeeper web client may lock up when used from a server machine if the "-" (reduce resource height) or "+" (increase resource height) accelerator keys are hit during the initial paint of LifeKeeper resources for that server.

Solution

To recover, open Windows Task Manager and select "End Task" for the "LifeKeeper - <web browser, e.g., Microsoft Internet Explorer>" application. It may take up to one minute for the processes to end. Restart the LifeKeeper web client from the Start->All Programs->SteelEye shortcut and wait for initial screen paint to complete before using accelerator keys.

LifeKeeper Web Client May Lock Up - Multiple Changes Made to Existing Hierarchy

Symptom

The LifeKeeper web client may lock up when used from the server machine if multiple changes are made to an existing hierarchy (i.e. create/delete dependencies), closing and reopening the LifeKeeper client between changes.

Solution

To recover, open **Windows Task Manager** and select **End Task** for the "LifeKeeper - <web browser, e.g., Microsoft Internet Explorer>" application. It may take up to one minute for the processes to end. For hierarchy administration on the server, use the LifeKeeper GUI (Admin Only) application. Shortcut is Start->All Programs->SteelEye->LifeKeeper->LifeKeeper (Admin Only).

New Evaluation License Key Error

Symptom

When evaluating LifeKeeper for Windows v7.2, an error may occur if a new evaluation license key is not used. The old evaluation licenses will not work on this release.

Solution

A new evaluation license key must be obtained. Restart the **License Key Manager** and enter the new, properly formatted license key.

Oracle Service Failure

Symptom

It has been reported that the Oracle Service fails to start during resource restore in Oracle Release 10.2.0.1 and 10.2.0.2.

Oracle Service enters the **RUNNING** state but returns to the **START PENDING** state. LifeKeeper detects that the Oracle service has failed to remain in the RUNNING state. The Oracle Service is running, but LifeKeeper does not detect this since the Oracle state is not correctly set to RUNNING before the restore timeout is exceeded.

Solution

Upgrade to Oracle Release 10.2.0.4.

Recovering Out-of-Service Hierarchies

Description

As a part of the recovery following the failure of a LifeKeeper server, resource hierarchies that are configured on the failed server but are not in service anywhere at the time of the server failure are recovered on the highest priority alive server at the time of the failure. This is the case no matter where the out-of-service hierarchy was last in service including the failed server, the recovering server or some other server in the hierarchy.

Remove Hangs During Recovery Kit Uninstall

Symptom

If a recovery kit is uninstalled while there are resource hierarchies of that kit in service, the **Remove** hangs.

To avoid this situation, it is recommended to always take a recovery kit’s resource hierarchies Out of Service and delete them before uninstalling the recovery kit software.

Solution

If you encounter this situation, you will most likely need to re-boot your system since there are many related processes that hang, and clearing them all can be difficult.

Replicated Volume Switchover Failure

Description

If the DELETEMIRROR command fails during switchover of a replicated volume, the following error message will display in the Output panel:

```
*WARNING* (No. 12104) Delete mirror action for volume R: failed with status 51
(command=C:/SDR/EMCmd10.10.1.2 DELETEMIRROR R: 10.10.1.1)
```

Refer to the table below for the Error Status number, description and recommended action. You can also use the following command to obtain more information about the error status:

```
Command:   net helpmsg {status}
Example:   net helpmsg 51
```

Error Status	Description	Recommended Action
5	Permission issues on the current SOURCE are not allowing the mirror to be deleted.	Check both systems for permission differences that may exclude the Local System Account from accessing the mirrored volume.
46	Mismatched user/password combination between systems. This will probably only occur if running DELETEMIRROR from the command prompt.	Use a domain account or make sure the local user accounts you are signing on with have the same password.
51	Windows cannot find the network path.	Make sure all network cards in both systems have File & Print Sharing for Microsoft Networks checked.
53	Cannot access the IP Address specified.	Verify your network configuration (including HOSTS files and DNS are all resolving the IP address consistently.
207	The ring 2 stack is in use	Make sure all the network cards in both systems have Client for Microsoft Networks checked.

Restore and Health Check Account Failures

Symptom

Some recovery kits monitor protected resources by performing query operations that simulate user and/or client activity. This provides LifeKeeper with accurate status information about a protected application or service. It also requires that a valid user account ID and password with login privileges be provided during resource object creation. If the user account does not have login privileges on a particular system, the following error message will be recorded in the **Windows Application Event Log**:

Error Number 1385 - "Logon failure: the user has not been granted the requested logon type at this computer.

Solution

Have the domain administrator provide login privileges for the user account. Also, most recovery kits that require an ID and password have a resource properties or configuration tab available for administrators to change the account information for the resource. Right-click on the resource object and select the appropriate properties or configuration tab. If the resource does not have an account update feature, the resource object must be deleted and a new one created with updated account information.

SQL 2005 and SQL 2008

Symptom

When using SQL 2005 and SQL 2008, after a switchover or a failover, the variable @@servername still points to the primary system.

Solution

You can use “select SERVERPROPERTY(‘ServerName’)” instead of using the variable @@servername. This query will return the correct name of the machine after a switchover or failover.
or

1. Execute the following commands on the new backup server:

```
sp_dropserver @server=’sys-A’
 sp_addserver @server=’sys-B’, @local=’LOCAL’
```

2. Restart the service.

SQL Server Reporting Services (MSSQLSERVER)

Symptom

When protecting SQL Server 2008 R2 services, the "SQL Server Reporting Services (MSSQLSERVER)" may be selected as an optional protected service. However, if the time required to start this service exceeds the default Windows service timeout, you may get Error 1053, the service may fail to start and the LifeKeeper resource in-service operation will fail.

Solution

This problem may be related to system performance and configuration issues. The recommended action is to not protect this service. However, if it must be protected, the following registry setting will extend the time available for services to start. In the "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control" Registry key, add a "ServicesPipeTimeout" value as a DWORD, and set the value to 60000 (decimal, = 60 secs.).

Two-Server Cluster Issue

Symptom

In a two-server cluster, when the primary server fails or is shut down which causes the hierarchies to fail over to the backup server, and the backup server also fails or is shut down before the hierarchies are entirely failed over to the backup server, the following behavior has been detected:

When both servers are rebooted, some of the resources in the hierarchies will be in service on one server and some will be in service on the other server. Some of the higher-level parent resources may not be in service on either server.

Solution

After both servers have been restarted and have completed LifeKeeper initialization, select the parent resource in a hierarchy that did not come in-service from the Hierarchy Administration interface and bring it in-service manually. Repeat this task until all hierarchies are in-service.

Unknown User Name or Bad Password

Access Denied: Unknown User Name or Bad Password

Symptom

If a LifeKeeper client tries to communicate with a server that is in the process of shutting down, the server may abort the validation process by refusing to allow the client to log on. In that case, the client will display a message stating “Access Denied: unknown user name or bad password. Only members of the LifeKeeper-authorized security groups can use LifeKeeper. Would you like to re-enter the authentication data?”

Solution

Click **Yes** to input new credentials, and then click either **Cancel** or re-enter the credentials and click **OK**. **Note**: If you click **No** initially, the LifeKeeper GUI will disconnect from that server and will not reconnect automatically.

Web Client Troubleshooting

Background

The web client does not display the Cluster Connect dialog.

Answer

If the web client does not display the Cluster Connect dialog, try the following:

1. Check whether the applet failed. Usually a message is printed somewhere in the browser window specifying the state of the applet. In Netscape and Internet Explorer, an icon may appear instead of the applet in addition to some text status. Clicking this icon may bring up a description of the failure.

2. Open the **Java Console**.

- For **FireFox**, **Netscape** and **older versions of Internet Explorer**, run the Java Plug-In applet from your machine's Control Panel and select the option to show the console. Restart your browser.
- For **recent versions of Internet Explorer**, select **Tools > Sun Java Console**. If you do not see the Sun Java Console menu item, select **Tools > Manage Add-Ons** and enable the console, after which you may need to restart your browser before the console will appear.
- For **Mozilla**, select **Tools > Web Development > Sun Java Console**.

If the web client is not open, reopen the URL, **`http://<server name>:81,`** to start it.

Check the console for any messages. The messages should help you resolve the problem. If the problem appears to be network related, refer to the Network-Related Troubleshooting section.

Win2008 - IIS Resource Hierarchy Creation Error

Symptom

On a Win2008 system when creating an IIS Resource Hierarchy, you receive the following message: `No qualified sites were found...` and the create fails.

BACKGROUND / TROUBLESHOOTING:

Run the following command:

```
C:\LK\Admin\kit\webapp\bin>enumiis query all
```

Look for the following error message:

```
ERROR: CoCreateInstance Failed! Error: -2147221164 (80040154)

ERROR: W3Service Com Object Failed to initialize
```

Solution

With Win2008, the IIS 6 Management Compatibility (Role Service) is required for the LifeKeeper IIS Kit. You should install all of this option (Metabase Compatibility, WMI Compatibility, Scripting Tools, Management Console).