LifeKeeper Single Server Protection Technical Documentation v9.2.2

March 2018

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

Introduction

LifeKeeper Single Server Protection (SSP) allows for application monitoring in a single node configuration (i.e., no cluster requirements or restraints). Single node environments may be physical or virtual (vSphere, KVM, Amazon EC2). LifeKeeper SSP is built on the proven and stable architecture of SIOS LifeKeeper. LifeKeeper SSP provides superior application monitoring and can perform recovery of failed applications and system infrastructure items (e.g., NFS share, IP address, File System). If an application cannot be recovered for some reason, LifeKeeper SSP will initiate a restart of the node via a system reboot or via a VMware HA restart for VMware virtual machines configured for VM and Application Monitoring.

Note:Because LifeKeeper SSP is built using the SIOS LifeKeeper technology, you will see references to LifeKeeper throughout the documentation as well as references to information found in the SIOS Protection Suite for Linux documentation for topics common to both products. When referencing these common topics the following subject items do not apply to LifeKeeper SSP:

- Clustering
- Communication Paths
- Shared Storage (requirements, configuration, …)
- Extending / Unextending resource hierarchies
- Storage Kits (DR, DMMP, HDLM, LVM, MD, PPATH and NEC SPS)

  Note: Without the underlying storage kits in LifeKeeper SSP, steps must be taken to ensure the devices required to mount protected file systems are activated during system boot (e.g. if the file system is mounted on a logical volume the volume must be in the active state before LifeKeeper SSP starts)

- Resource/Machine failovers (by default with LifeKeeper SSP these result in a node restart)
- Resource switchovers
- Switchable IP Addresses (with LifeKeeper SSP Switchable IP addresses are required for some protected applications but since there is only a single node no switching actually takes place)

  Note: When operating on Amazon EC2, assign a secondary private IP address to the NIC using the Amazon EC2 Management Console prior to creating the IP resource. Next, create the IP resource as the private IP address on the NIC that is using the LifeKeeper GUI client. An Elastic IP can now be associated with the IP resource

For more information on the SIOS LifeKeeper product, on which LifeKeeper SSP is built, please see the SIOS Protection Suite for Linux documentation for the common release number. This documentation will provide detailed information on resource hierarchy creation, resource types, states and relationships, Graphical User Interface (GUI), as well as common and advanced tasks.

Documentation and Training

A complete reference providing instructions for installing, configuring, administering and troubleshooting SIOS LifeKeeper Single Server Protection for Linux is available in the Documentation section of the SIOS Technology Corp. website. The following sections cover every aspect of SIOS LifeKeeper Single Server Protection for Linux:

Section	Description
Introduction and Installation	Provides useful information for planning and setting up your LifeKeeper Single Server Protection environment, installing and licensing LifeKeeper Single Server Protection and configuring the LifeKeeper graphical user interface (GUI).
Administration	Discusses server-level tasks such as editing server properties and creating resources and resource-level tasks such as editing, extending or deleting resources.
User's Guide	Contains detailed information on the LifeKeeper GUI, including the many tasks that can be performed within the LifeKeeper GUI.
Troubleshooting	Describes known issues and restrictions and suggests solutions to problems that may be encountered during installation, configuration and/or use of SIOS LifeKeeper Single Server Protection for Linux.
Recovery Kits	Contains planning and installation instructions as well as administration, configuration and user information for the Optional Recovery Kits that allow LifeKeeper Single Server Protection to manage and control specific applications.

Training

LifeKeeper Single Server Protection training is available through SIOS Technology Corp. or through your LifeKeeper Single Server Protection provider. Contact your sales representative for more information.

Technical Support

As a SIOS Technology Corp. customer with a valid Support contract, you are entitled to access the new SIOS Technology Corp. Support Self-Service Portal.

The SIOS Technology Corp. Support Self-Service Portal offers you the following capabilities:

- Search our Solution Knowledge Base to find solutions to problems and answers to questions
- Always on 24/7 service with the SIOS Technology Corp. Support team to:
- **Log a Case** to report new incidents
- **View Cases** to see all of your open and closed incidents
- **Review Top Solutions** provides information on the most popular problem resolutions being viewed by our customers.

Contact SIOS Technology Corp. Support at support@us.sios.com to set up and activate your Self-Service Portal account.

You can also contact SIOS Technology Corp. Support at:

1-877-457-5113 (Toll Free)

1-803-808-4270 (International)

Email: support@us.sios.com

LifeKeeper Single Server Protection Core

LifeKeeper Single Server Protection Core is composed of three major components:

- Core Software
- File System, Generic Application and IP Recovery Kit Software
- GUI Software

Note: Because LifeKeeper Single Server Protection is built using the SteelEye LifeKeeper technology, you will see references to LifeKeeper throughout the documentation.

LifeKeeper Single Server Protection Core Software

The LifeKeeper Single Server Protection Core Software consists of the following components:

- LifeKeeper Configuration Database (LCD) - The LCD stores information about the LifeKeeper Single Server Protection-protected resources. This includes information on resource instances, dependencies, shared equivalencies, recovery direction, and LifeKeeper Single Server Protection 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 and for LifeKeeper Single Server Protection inter-process local communication.
- LifeKeeper Single Server Protection Alarm Interface - The LifeKeeper Single Server Protection 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 Single Server Protection-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.

File System, Generic Application and IP Recovery Kit Software

The LifeKeeper Single Server Protection Core provides protection of specific resources on a server. These resources are:

- File Systems - LifeKeeper Single Server Protection allows for the definition of file systems. A LifeKeeper Single Server Protection file system resource. File System Health Monitoring detects disk full and improperly mounted (or unmounted) file system conditions. Depending on the condition detected, the Recovery Kit may log a warning message, attempt a local recovery, or reboot the server.

  Specific help topics related to the File System Recovery Kit include Creating a File System Resource Hierarchy and File System Health Monitoring.

- Generic Applications - The Generic Application Recovery Kit allows protection of a generic or user-defined application that has no predefined Recovery Kit to define the resource type. This kit allows a user to define monitoring and recovery scripts that are customized for a specific application.

  See Creating a Generic Application Resource Hierarchy.

- IP Addresses - The IP Recovery Kit provides a mechanism to recover an IP address from a failed state. Applications under LifeKeeper Single Server Protection protection are associated with the protected IP address.

Refer to the IP Recovery Kit Technical Documentation included with the Recovery Kit for specific product, configuration and administration information.

LifeKeeper GUI Software

The LifeKeeper GUI is a client / server application developed using Java technology that provides a graphical administration interface to LifeKeeper Single Server Protection and its configuration data. The LifeKeeper GUI client is implemented as both a stand-alone Java application and as a Java applet invoked from a web browser.

Integration with VMware HA

As noted in the Introduction Section, LifeKeeper Single Server Protection is designed for use in both physical and virtual environments. When LifeKeeper SSP is installed in a VMware VM the HA features of VMware can be used in conjunction with LifeKeeper SSP to monitor and recover from any protected resource or node failure. To enable these features see Enabling VMware HA Integration with LifeKeeper Single Server Protection. Additionally, LifeKeeper SSP provides an optional component that provides a management interface that integrates with VMware vCenter (see SteelEye Management Console topic).

SteelEye Management Console

The *SteelEye Management Console*, or *SMC*, is an optional piece of LifeKeeper Single Server Protection when running in VMware HA configurations. The SMC is a dedicated system that provides a management interface that is integrated with VMware vCenter.

The following topics will help in understanding the setup, installation and operation of a SteelEye Management Console for use with VMware vCenter Server and SIOS LifeKeeper Single Server Protection. The details are broken down into the following categories:

Installation Overview

The installation of the SteelEye Management Console, or *SMC*, consists of several important steps:

1. A server (virtual or physical) must be identified to host the SMC. This server must be a **dedicated system for running the SMC.** SIOS Technology Corp. does not currently support running the SMC on servers being used for other purposes. This server does not need to be very powerful, thus a small virtual machine should be adequate. See System Requirements for more details on server requirements.
2. The SMC software must be installed by running the `setup` script either from CD media or the `.img` file. This process will make the necessary modifications to the system, install the SMC software components and start the SMC services.
3. The VMware vSphere Client plug-in must be registered with the vCenter server. **Note:** The SMC can only integrate with a single vCenter instance; therefore, if deploying multiple vCenter instances, the SMC software must be installed for each vCenter instance.
4. The SMC must be configured with the credentials required to communicate with each SIOS LifeKeeper Single Server Protection server managed by the vSphere Client (or the subset that will be managed via the SMC). If a certain set of credentials is valid for multiple LifeKeeper Single Server Protection systems, they can be entered once and credentials for the remaining LifeKeeper Single Server Protection nodes will need to be added individually. More details can be found in Configuring Credentials.

System Requirements

The SteelEye Management Console, or *SMC*, must be installed on a dedicated server. The server

can be physical or virtual but must not be used for any other purpose. SIOS Technology Corp. does not currently support running the SMC software on servers that are also being used for other purposes. This server must meet the following minimum requirements:

- It must be an Intel (or AMD) 64-bit system capable of running Red Hat Enterprise Linux/CentOS 6.4 x86_64. This can either be a bare metal system or a virtual machine.
- It must have at least 512MB of RAM.
- It must have at least 8GB of disk space with at least 1GB available for the `/opt` filesystem.
- It must have at least one network adapter.
- It must be able to communicate directly with, via TCP/IP, a VMware vCenter Server (if using the vSphere Client plug-in) and any LifeKeeper Single Server Protection servers that will be viewed/managed by the SMC. There may be network segment/routing/firewall considerations in choosing a server to host the SMC.
- It must be installed with an openssl-devel (its version must coincide with the version of the openssl-devel included in the installing CD).

Setup Prerequisites

Once a system is chosen, the following prerequisites must be set up prior to installing the SMC software.

- The system must be installed with the default, ***base*** software packages and does not require any special package selection.
- The system must have the core operating system **yum repository enabled** and available. This means either the install media repository or a network repository for base system must be enabled in `/etc/yum.repos.d/`.

Once the system is running a supported operating system, the instructions in Running Setup can be followed to install the SMC software components.

Running Setup

The SteelEye Management Console software components can be installed from either a CD/DVD or from the ISO img file containing the CD image. All software procedures from this point must be run as the `root` user.

In either case, the CD or ISO img file must be mounted. The CD can be mounted in the usual way and the img file can be mounted via the loopback device with a command like:

mount -o loop <path-to>/smc.img /mnt

(/mnt can be any location suitable for mounting the image)

Installing the Software

Once the image is mounted, the installation can be started via:

cd /mnt; ./setup

The setup tool will guide the installation process and the following will occur to set up the software components:

- System packages will be upgraded and/or removed to ensure that there are no packages on the system that conflict with the SMC components. This process includes removing the pre-installed webserver and components that depend on it. This process can take several minutes.
- Next, the setup tool will install/upgrade SIOS packages for all the SMC software components. This can also take several minutes and will include other required changes to the system, and in particular, changes to the iptables firewall configuration to allow clients to communicate with the SMC. The iptables configuration will be altered to allow HTTP traffic into the SMC server on TCP/IP port 80 and 443.
- Finally, the setup tool will install the required VMware SDK package. This will require agreeing to the VMware SDK end-user license. The SDK install will prompt for a file path for tool binaries to be installed. SIOS Technology Corp. recommends accepting the default location for these files.

Base vCenter and Credential Configuration

After the software components are installed, the setup tool will configure the vSphere Client plug-in and the default credentials for SIOS LifeKeeper Single Server Protection node communication. The following will happen to conclude the setup process:

- The setup tool will ask for the vCenter server name, user and password. This information will be used to register the vSphere Client plug-in provided by this SMC server with the given vCenter. After this step, the vCenter server should provide an additional tab for the plug-in. The plug-in can be re-registered or unregistered anytime after the initial installation and the details of that process can be found on the Configuring the vSphere Client Plug-in page.
- Last, the setup tool will ask for the ***default*** credentials to use when communicating with SIOS LifeKeeper Single Server Protection nodes. These credentials will be stored on the SMC server and will be used to communicate with LifeKeeper Single Server Protection servers unless credentials specific to the given server have been configured. The default credentials must have admin access to the LifeKeeper Single Server Protection nodes (on a typical installation, the user must be in the lkadmin group in the local `/etc/group` file) to allow the SMC to fully manage the LifeKeeper Single Server Protection systems. In general, the ***default*** credentials would be the *root* user and password of the installed LKSSP node. **Note:** While storage of passwords are base64 encoded they are **not** encrypted in the LifeKeeper credstore database. It is advised to use another LKSSP system account that has membership in the `lkadmin` group.The Configuring Credentials page has more details on how to manage the credentials used by the SMC.

The setup process should now be complete. Please look over the Verifying Installation page for more information on validating that the software was installed and configured correctly.

Credential Considerations

The final step in the setup process above is to store the default credentials for accessing LifeKeeper Single Server Protection systems. Default credentials in this case refers to credentials that will be

used to authenticate with LifeKeeper Single Server Protection systems if there are no per-system credentials configured for a system. The SMC will always fall back to using the default credentials.

For this reason, SIOS Technology Corp. recommends using the same credentials on all LifeKeeper Single Server Protection systems whenever possible to simplify the SMC configuration. This typically means using the root user on the LifeKeeper Single Server Protection systems, but any user that is common across all the systems will suffice as long as the user is in the `lkadmin` group.

SIOS LifeKeeper Single Server Protection vSphere Client Plug-in

The SIOS LifeKeeper Single Server Protection vSphere Client plug-in integrates with the VMware vSphere client to provide application monitoring status for protected virtual machines. The plug-in must be registered with the vCenter Server in order to operate.

The plug-in uses HTTPS secure communications for all transfers. It is normal to receive a Security Warning for the **LK4Linux Valid SMC** SSL certificate when first loading the plug-in in the vSphere client. The SSL certificate can be inspected and installed in the local certificate store, and the Security Warning can safely be ignored.

Plug-in Requirements

- VMware vSphere Version 4 or Version 5
- VMware vSphere Client
- Javascript and cookies enabled on Client system

For more information, see the Configuring the vSphere Client Plug-in and the Addressing vSphere Client Plug-in Security Warnings topics.

Configuring the vSphere Client Plug-in

The vSphere Client plug-in can be re-registered, or the credentials can be changed, at any time after installation. This includes registering the plug-in with a different vCenter server if needed. If the plug-in is going to be installed on a different vCenter server, it should first be unregistered with the current server.

Registering the vSphere Client Plug-in

Registering the plug-in is done via the /opt/LifeKeeper/bin/registerPlugin.pl tool. This tool takes three arguments: the vCenter server, the username and the password. All are required to re-register the vSphere Client plug-in. An example of running this tool would look like (*all on one line*):

```
/opt/LifeKeeper/bin/registerPlugin.pl --server=myvcenter.mydomain.com --username=vcuser
--password=vcpassword
```

Note: Password characters that are also shell characters need to be escaped to avoid shell interpretation.

Unregistering the vSphere Client Plug-in

The plug-ins installed from the SMC server can be listed with the following command:

```
/opt/LifeKeeper/bin/registerPlugin.pl --action=list
```

To remove the plug-in, the following command can be run (*all on one line*):

```
/opt/LifeKeeper/bin/registerPlugin.pl --action=remove --key=com.sios.us.lkssp
```

Note: The key used to unregister a plug-in must be the same as shown by the listaction.

Note: For information on security warnings, see Addressing vSphere Client Plug-in Security Warnings.

vSphere Client User Interface

SIOS LifeKeeper Single Server Protection Tab

When you register the vSphere Client plug-in, the **LifeKeeper Single Server Protection** tab will appear in the vSphere Client User Interface.

Context Menus

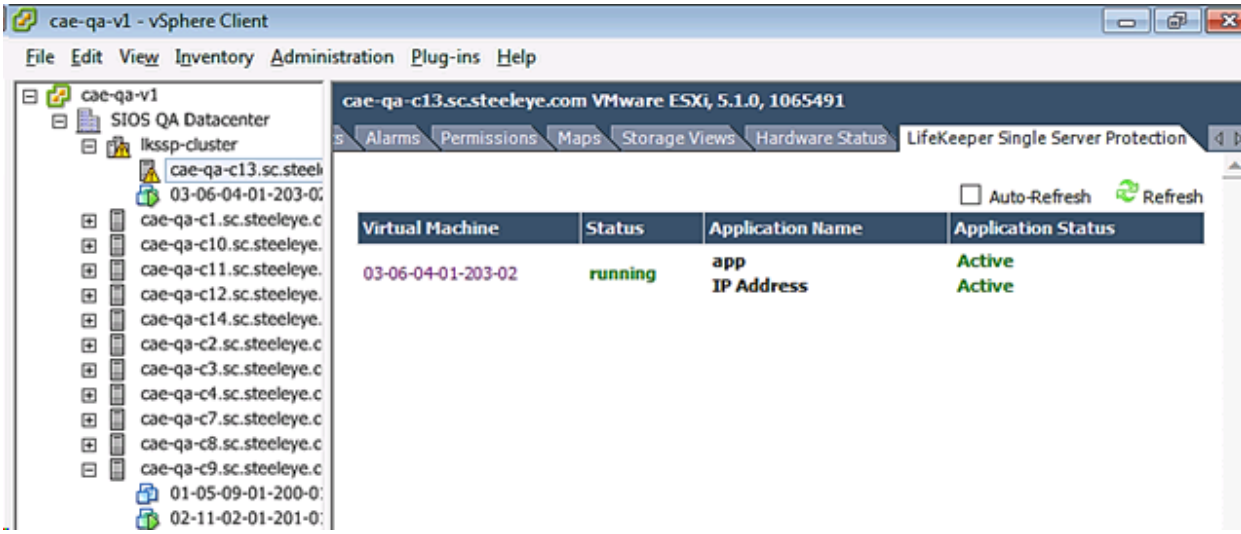
This vSphere Client plug-in offers several context levels for the LifeKeeper Single Server Protection tab depending on where you click in the left-side inventory tree:

Datacenter Level

Displays the virtual machines that are in the datacenter.

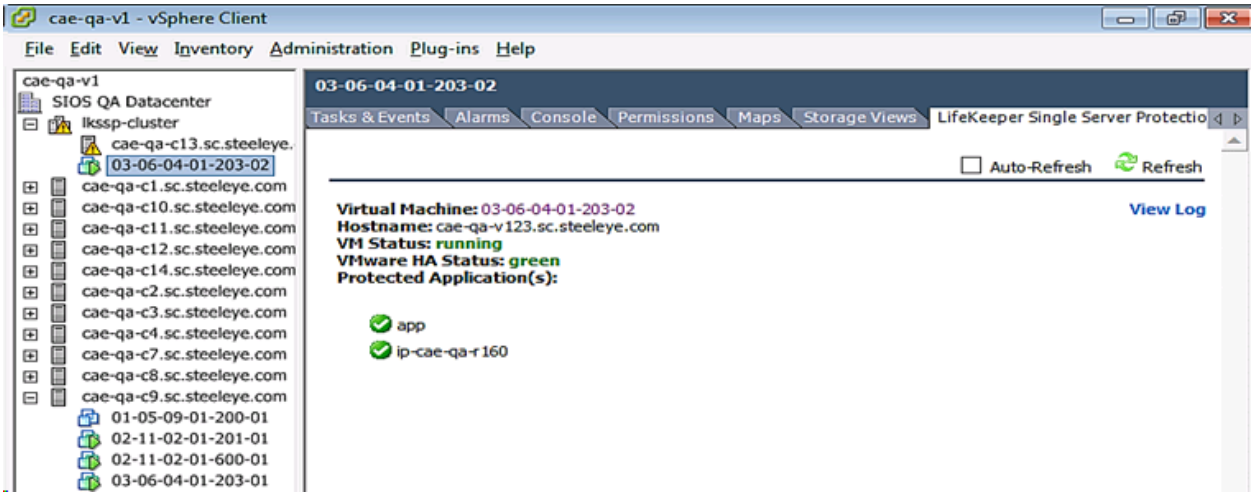
ESX or ESXi Level

Displays the status of the virtual machines running on the particular host.



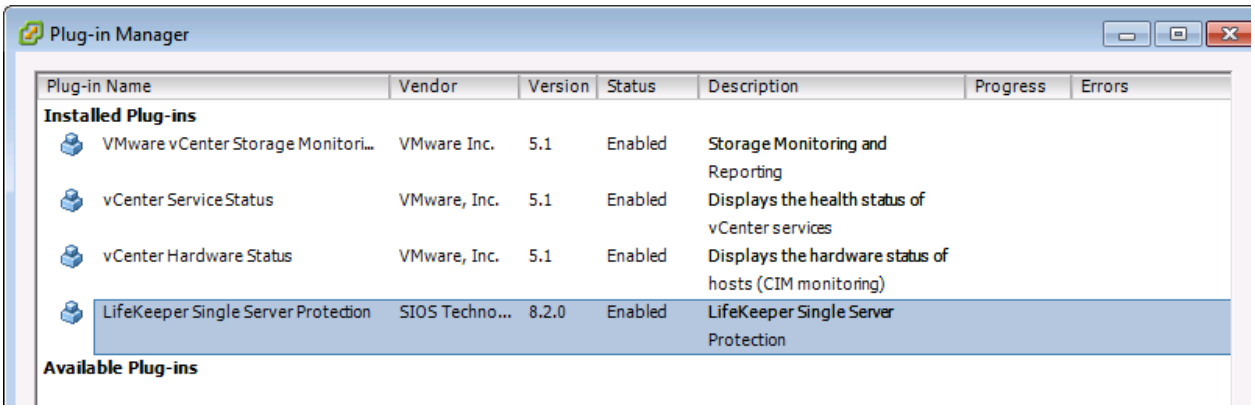
Virtual Machine Level

Displays specific node information: Virtual Machine Name, Hostname, VM Status, VMware HA Status, Protected Applications. There is also a **View Log** link, which allows you to view the LifeKeeper Single Server Protection log.



Manage Plug-Ins

Displays the status of the registered LifeKeeper Single Server Protection vCenter plug-in.



.

Other Views

The LifeKeeper Single Server Protection plug-in is also viewable at the **Datacenter**, **Virtual Application**, and **Resource Pool** levels.

Configuring Credentials

The SMC and LifeKeeper Single Server Protection software manages credentials for communicating with other systems (i.e., vCenter Server or SIOS LifeKeeper Single Server Protection) via a *credential store*. This store is used during plug-in registration (see Configuring the vSphere Client Plug-in) for example. This store can be managed, as needed, by the `/opt/LifeKeeper/bin/credstore` command. This command allows server access credentials

to be set, changed and removed - on a per server basis.

Adding or Changing Credentials

Adding and changing credentials are handled in the same way. A typical example of adding or changing credentials for a server, lkssp-server.mydomain.com, would look like this:

```
/opt/LifeKeeper/bin/credstore -k lkssp-server.mydomain.com myuser
```

In this case, *myuser* is the username used to access lkssp-server.mydomain.com and the password will be asked for via a prompt with confirmation (like *passwd*).

Note: The key name used to store LifeKeeper Single Server Protection server credentials on the SMC must match *exactly* the hostname of the LifeKeeper Single Server Protection server (as displayed in the **Hostname:** field of the vSphere Client Plug-in). If the hostname is an FQDN, then the credential key must also be the FQDN. If the hostname is a short name, then the key must also be the short name.

If following the 'Credential Considerations' suggested in Running Setup, a ***default*** key with an associated username and password will be used for authentication when no specific server keys exist. To add or change the *default* key, run:

```
/opt/LifeKeeper/bin/credstore -k default myuser
```

Listing Stored Credentials

The currently stored credentials can be listed by the following command:

```
/opt/LifeKeeper/bin/credstore -l
```

This will list the *keys* stored in the credential store and, in this case, the *key* indicates the server for which the credentials are used. (This command will not actually list the credentials, only the keys, since the credentials themselves may be sensitive.)

Removing Credentials for a Server

Credentials for a given server can be removed with the following command:

```
/opt/LifeKeeper/bin/credstore -d -k lkssp-server.mydomain.com
```

In this case, the credentials store for the server lkssp-server.mydomain.com will be removed from the store.

Additional Information

More information on the credstore utility can be found by running:

```
/opt/LifeKeeper/bin/credstore --man
```

This will show the entire man/help page for the command.

Verifying Installation

The SteelEye Management Console installation can be verified by connecting, via a web browser, to https://<smcserver>/ which should show a page indicating that the SMC services are available.
Note: It is normal to receive a security warning since the SMC uses a self-signed SSL certificate. This warning can safely be ignored.

If the browser indicates an error showing the page or fails to connect to the newly installed SMC server, please ensure all installation steps were completed without error and that the SMC server is network-accessible.

Troubleshooting

For troubleshooting, please see the SMC Troubleshooting section. Also, for information on security warnings, refer to the topic Addressing vSphere Client Plug-in Security Warnings.

Addressing vSphere Client Plug-in Security Warnings

The LifeKeeper Single Server Protection vSphere Client Plug-in uses a self-signed certificate to enable SSL communications. It is normal to receive a security warning when viewing the plug-in contents. To reduce or eliminate the security warnings, you should install the "LK4Linux Valid SMC" certificate in your vSphere Client system's certificate store. Additionally, you can install the "SIOS Technology, Corp." Certificate Authority (CA) certificate in your system's "Trusted Root Certification Authorities" certificate store.

To install the "LK4Linux Valid SMC" certificate:

1. When a security warning is displayed, choose **View Certificate**.
2. Click on the **Install Certificate** button.
3. Follow the wizard steps to install the certificate.

To install the "SIOS Technology, Corp." certificate authority (CA) certificate in the "Trusted Root Certification Authorities" store:

1. When a security warning is displayed, choose **View Certificate**.
2. Click the **Certification Path** tab.
3. Select the **SIOS Technology, Corp.** certificate.
4. Click **View Certificate** to view the CA cert.
5. Click on the **Install Certificate** button.
6. Click **Next**.
7. On the ***Certificate Store Wizard*** pane, select **Place all certificates in the following store** radio button.

8. The **Browse** button will now be enabled. Click it.
9. Select **Trusted Root Certification Authorities** from the **Select Certificate Store** list.
10. Click **OK**.
11. Click **Next** and complete the wizard.

LifeKeeper API

The LifeKeeper API is used to allow communications between LifeKeeper Single Server Protection servers and the SteelEye Management Console (SMC). Currently, this API is reserved for internal use only but may be opened up to customer and third party usage in a future release.

Network Configuration

Each LifeKeeper Single Server Protection server provides the API via an SSL Connection on port 778. This port may be changed using the configuration variable `API_SSL_PORT` in `/etc/default/LifeKeeper`. This variable is set in `/etc/default/LifeKeeper.local.pl` on the SMC. (**Note:** This setting controls **API client communications to LifeKeeper Single Server Protection servers,** not access to the SMC itself, which is always on port 443). Both LifeKeeper Single Server Protection and the SMC **must** use the same value for `API_SSL_PORT`.

Authentication

The LifeKeeper API uses PAM for authentication. Access to the API is only granted to users that are members of the group *lkadmin*, *lkoper* or *lkguest*. Depending on the PAM configuration of the system, this can be accomplished by using the local system files (i.e. */etc/passwd* and */etc/group*) or by including the user in an LDAP or Active Directory group.

Note: The LifeKeeper API does not use the user database that is managed by the *lkpasswd* utility.

SMC Use of the API

The SMC uses the API to gather information from the LifeKeeper Single Server Protection servers. The SMC uses the *credstore* utility to manage user account info for LifeKeeper Single Server Protection servers. The SMC uses the LifeKeeper Single Server Protection server name as the key in the credential store, so the system name of the LifeKeeper Single Server Protection server should be passed as the *-k* option to the *credstore* utility when specifying credentials for a LifeKeeper Single Server Protection server. The SMC will also check for and use credentials stored in the ***default*** key if it does not find credentials for a specific server.

Using Custom Certificates

LifeKeeper Single Server Protection uses SSL/TLS to communicate between different systems. By default, the product is installed with default certificates that provide some assurance of identity between nodes. This document explains how to replace these default certificates with certificates created by your own Certificate Authority (CA).

How Certificates Are Used

Communication to the SteelEye Management Console (SMC) and LifeKeeper Single Server Protection servers uses SSL/TLS to protect the data being transferred. Both systems provide a certificate to identify themselves, and both systems use a CA certificate to verify the certificate that is presented to them over the SSL connection.

Four certificates are involved:

- /opt/LifeKeeper/etc/certs/LK4LinuxValidNode.pem (LifeKeeper Single Server Protection server certificate)
- /opt/LifeKeeper/etc/certs/LK4LinuxValidSMC.pem (SMC server certificate)
- /opt/LifeKeeper/etc/certs/LK4LinuxClient.pem (LifeKeeper Single Server Protection client certificate, installed on all servers)
- /opt/LifeKeeper/etc/certs/LKCA.pem (certificate authority, installed on all servers)

The first three certificates must be signed by the fourth certificate to satisfy the verification performed by the servers. Note that the common name of the certificates is not verified, only that the certificates are signed by the CA.

Using Your Own Certificates

In some installations, it may be necessary to replace the default certificates with certificates that are created by an organization's internal CA. If this is necessary, replace the four certificates listed above with new certificates *using the same certificate file names*. These certificates are of the PEM type. The LK4LinuxValidNode.pem, LK4LinuxValidSMC.pem and LK4LinuxValidClient.pem each contain both their respective key and certificate. The LK4LinuxValidNode.pem and LK4LinuxValidSMC.pem certificates are *server* type certificates. LK4LinuxValidClient.pem is a *client* type certificate.

If the default certificates are replaced, LifeKeeper Single Server Protection and SMC will need to be restarted to reflect the changes. If the certificates are misconfigured, steeleye-lighttpd daemon will not start successfully and errors will be received in the LifeKeeper Single Server Protection log file. If problems arise, refer to this log file to see the full command that should be run.

Application Recovery Kits

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

Note that the following documentation is LifeKeeper documentation. Therefore, there are several points to keep in mind when reading these documents:

- LifeKeeper Single Server Protection is a single node product, so the following concepts do not apply to LifeKeeper Single Server Protection environments: communication paths, clusters, backup nodes, extending hierarchies to backup nodes, switchover or failover of hierarchies to backup nodes. Any references to these terms or operations should be ignored for the purposes of LifeKeeper Single Server Protection configuration and administration.
- LifeKeeper Single Server Protection does not protect shared storage or perform data replication, so applications can be placed on any available storage and protected with LifeKeeper Single Server Protection, which greatly simplifies installation and configuration. Any references to moving application data to shared disk can be ignored when reading these guides.
- Being a VMware-based product, any discussion of hardware requirements is moot with the exception that CPU, memory and disk space requirements do still apply to virtual machines.
- While IP Address can still be protected (and in some configurations, will need to be protected), any reference to a "switchable IP" should be ignored. With only one system in the configuration, a protected IP is, of course, not switchable, but will still be protected on the single node on which it is defined.
- SAP hierarchies cannot be created using the GUI. For a workaround, see the SAP Known Issue in the Troubleshooting section.
- Oracle hierarchies cannot be created on the root file system. For a workaround, see the Oracle Known Issue in the Troubleshooting section.

Technical documentation for each Recovery Kit provides configuration and administration information specific to that particular software package. Visit the Technical Documentation section of our website for links to the Recovery Kit documentation.

Resource Hierarchies

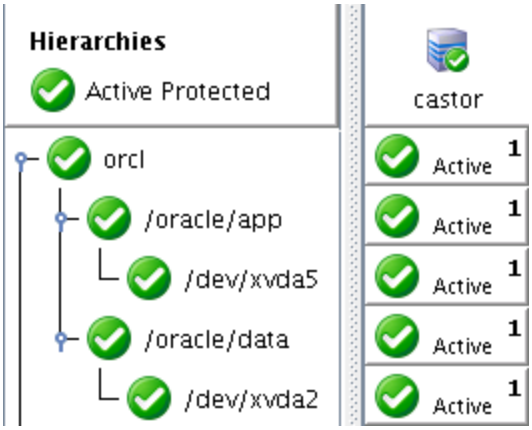
The LifeKeeper GUI enables you to create a resource hierarchy on a server. After you create the resource hierarchy, LifeKeeper Single Server Protection manages the stopping, starting, monitoring and recovery of the resources within the hierarchy. The related topics below provide background for hierarchy definition tasks.

Resource Types

A resource is a protected software entity, such as an application, database, file system, etc. Resources are categorized by type. LifeKeeper Single Server Protection Core supplies File System, IP Address and Generic Application resource types. The various recovery kits provide other database, application and system infrastructure resource types.

For example, a resource hierarchy for a protected Oracle database may include resources of the following types:

- **database/oracle**: Protected Oracle database instance.
- **gen/filesys:** Linux file system resource.
- **device:** Disk partition or virtual device, for example *sdc1.*

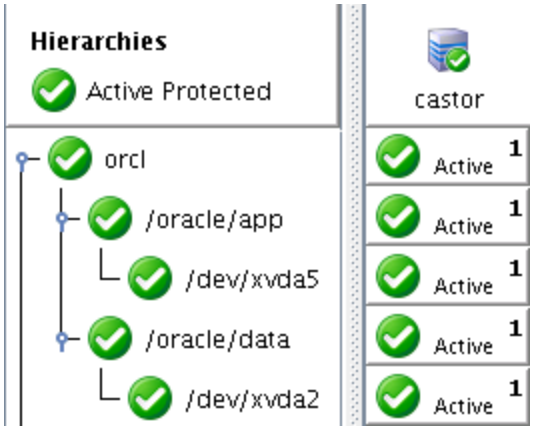


Resource States

State	Meaning
In-Service, Protected (ISP)	Resource is operational. LifeKeeper Single Server Protection application recovery is operating normally.
In-Service, Unprotected (ISU)	Resource is operational but may be in a **warning** state.
Out-of-Service, Failed (OSF)	Resource has gone out-of-service because of a failure. Recovery has not been completed or has failed. LifeKeeper Single Server Protection alarming is not operational for this resource.
Out-of-Service, Unimpaired (OSU)	Resource is out-of-service.
Illegal State (ILLSTATE)	This state appears in situations where no state has been set for a resource instance. Under normal circumstances, this invalid state does not last long: a transition into one of the other states is expected.

Hierarchy Relationships

LifeKeeper Single Server Protection allows you to create relationships between resource instances. The primary relationship is a dependency. The combination of resource instances and dependencies is referred to as a resource hierarchy.



(In the above image, the Oracle database *orcl* depends on two file systems, */oracle/app* and */oracle/data.*)

When one resource *depends* on another to function properly, a dependency relationship is needed. A dependency defines the order in which resources are taken in and out of service. Normally, LifeKeeper Single Server Protection recovery kits will create the proper dependencies when an application is protected. However, there may be cases where custom dependencies need to be created (one common case is with the use of the generic application recovery kit to protect a custom application).

Detailed Status Display

This topic describes the categories of information provided in the detailed status display as shown in the following example of output from the **lcdstatus** command. For instructions on how to display this information, see the LCD(1M) man page. At the command line, you can enter either **man lcdstatus** or **man LCD**. For status information available in the LifeKeeper GUI, see Viewing the Status of a Server or Viewing_the_Status_of_Resources.

Example of detailed status display:

Resource Hierarchy Information

```
Resource hierarchies for machine "castor":

ROOT of RESOURCE HIERARCHY
 orcl: id=orcl app=database type=oracle state=ISP
        initialize=(AUTORES_ISP) automatic restore to IN-SERVICE by
LifeKeeper
        info=/oracle/app/oracle/product/11.2.0/dbhome_1
        reason=restore action has succeeded
        depends on resources: /oracle/data,/oracle/app
 /oracle/data: id=/oracle/data app=gen type=filesys state=ISP
        initialize=(AUTORES_ISP) automatic restore to IN-SERVICE by
LifeKeeper
        info=ext3rw0
        reason=restore action has succeeded
        depends on resources: /dev/xvda2
        these resources are dependent: orcl
 /dev/xvda2: id=/dev/xvda2 app=scsi type=DEVNAME state=ISP
        initialize=(AUTORES_ISP) automatic restore to IN-SERVICE by
LifeKeeper
        reason=restore action has succeeded
        these resources are dependent: /oracle/data
 /oracle/app: id=/oracle/app app=gen type=filesys state=ISP
        initialize=(AUTORES_ISP) automatic restore to IN-SERVICE by
LifeKeeper
        info=ext3rw0
        reason=restore action has succeeded
        depends on resources: /dev/xvda5
        these resources are dependent: orcl
 /dev/xvda5: id=/dev/xvda5 app=scsi type=DEVNAME state=ISP
        initialize=(AUTORES_ISP) automatic restore to IN-SERVICE by
LifeKeeper
        reason=restore action has succeeded
        these resources are dependent: /oracle/app
```

The following LifeKeeper machines are known:
    machine=castor state=ALIVE

Resource Hierarchy Information

LifeKeeper Single Server Protection displays the resource status beginning with the root resource. The display includes information about all resource dependencies.

The first line for each resource description displays the resource tag name followed by a colon (:), for example: */oracle/data*:. The following items describe the resources in the hierarchy:

- **id**. Unique resource identifier used by LifeKeeper Single Server Protection.
- **app**. Identifies the type of application, for example the sample resource is a *database* application.
- **type**. Indicates the resource class type, for example the sample resource is an *oracle* type application.
- **state**. Current state of the resource:
    - ISP—In-service, protected.
    - ISU—In-service, unprotected.
    - OSF—Out-of-service, failed.
    - OSU—Out-of-service, unimpaired.
- **initialize**. Specifies the way the resource is to be initialized.
- **info**. Contains resource-specific information used internally by the recovery kit.
- **reason**. If present, describes the reason the resource is in its current state. For example, an application might be in the OSU state because it has been taken out of service.
- **depends on resources.** If present, lists the tag names of the resources upon which this resource depends.
- **these resources are dependent.** If present, indicates the tag names of all parent resources that are directly dependent on this resource.

Installing the LifeKeeper Single Server Protection Software

Install the LifeKeeper Single Server Protection software on each server in the LifeKeeper Single Server Protection configuration. Each LifeKeeper Single Server Protection server must have the packages necessary to support your configuration requirements, including any optional Recovery Kit packages.

IMPORTANT: Please review the Linux Dependencies topic prior to installing LifeKeeper Single Server Protection .

The LifeKeeper Single Server Protection core package and any optional recovery kits will be installed through the command line using the LifeKeeper Single Server Protection Installation Image File (*lkssp.img*). This image file provides a set of installation scripts designed to perform user-interactive system setup tasks that are necessary when installing LifeKeeper Single Server Protection on your system. The installation image file identifies what Linux distribution you are running and, through a series of questions you answer, installs various packages required to ensure a successful LifeKeeper Single Server Protection installation. A licensing package is also installed providing utilities for obtaining and displaying the Host ID of your server and your Entitlement ID once your licenses have been installed. The Entitlement ID is used to obtain valid licenses for running LifeKeeper Single Server Protection and was provided with your Software.

Note: These installation instructions assume that you are familiar with the Linux operating system installed on your servers.

IMPORTANT:

- LifeKeeper Single Server Protection does not provide shared storage support or I/O fencing. Each server must use local disk storage for application data.
- All LifeKeeper Single Server Protection packages are installed in the directory */opt/LifeKeeper*.
- If you are re-installing the existing version of LifeKeeper, you must remove the old LifeKeeper packages first. A standard LifeKeeper installation requires that you redefine any existing resource hierarchies. If you wish to retain your current resource hierarchy definitions, refer to the .
- If you receive an error message referencing the LifeKeeper Distribution Enabling package when you are installing LifeKeeper Single Server Protection you should run/re-run the **setup** script on the LifeKeeper Single Server Protection Installation Image File.

Installing the LifeKeeper Single Server Protection Software

LifeKeeper Single Server Protection will be installed through the command line regardless of the Linux distribution you are operating under.

1. Mount the `lkssp.img` file using the following command:

   ```
   mount <PATH/IMAGE_NAME> <MOUNT_POINT> -t iso9660 -o
   loop
   ```

   Where PATH is the path to the image
   IMAGE_NAME is the name of the image
   MOUNT_POINT is the path to mount location

2. Change to the `lkssp.img` mounted directory and type the following:

   ```
   ./setup
   ```

3. Text will appear explaining what is going to occur during the installation procedure. You will now be asked a series of questions where you will answer "**y**" for **Yes** or "**n**" for **No**. The type and sequence of the questions are dependent upon your Linux distribution.

   Read each question carefully to ensure a proper response. It is recommended that you answer **Yes** to each question in order to complete all the steps required for a successful LifeKeeper Single Server Protection Installation.

4. Next, the LifeKeeper Single Server Protection Core Packages will be installed.

5. The setup script will then perform the installation of the licensing utilities. See Obtaining and Installing the License for details.

6. After you have answered all the questions posed by the setup script, you will be informed that the installation was successful and then be presented with a list of all LifeKeeper Single Server Protection Recovery Kits available for installation.

   Note: Trace information for execution of the setup scripts is saved in */var/log/LK_install.log*.

7. Select the kits you would like installed by highlighting the kit and pressing the "space" bar. This will place an "i" next to each kit that will be installed. Then press **Enter**.

   Note: To add kits at a later time, simply run setup again followed by -k:

   ```
   ./setup -k
   ```

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

The Installation script installs the Licensing Utilities package which obtains and displays all of the available Host IDs for your server during the initial install of your LifeKeeper Single Server Protection Software. Once your licenses have been installed the utility will return the Entitlement ID if it is available or the Host IDs if it is not.

Note:Host IDs, if displayed will always be based on the MAC address of the NICs.

Any LifeKeeper Single Server Protection licenses obtained from the SIOS Technology Corp. Licensing Operations Portal will contain your Entitlement ID and will not be locked to a specific node in the cluster. The Entitlement ID (Authorization Code) which was provided with your LifeKeeper Single Server Protection Software, is used to obtain the permanent license required to run the LifeKeeper Single Server Protection Software. The process is illustrated below.

Note: Each software package requires a license for each server.

Perform the following steps to obtain and install your license(s) for each server in the LifeKeeper Single Server Protection cluster:

1. **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.
2. **Obtain your licenses from the SIOS Technology Corp. Licensing Operations Portal.**
    a. Using the system that has internet access, log in to the SIOS Technology Corp. Licensing Operations Portal.
    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 **Add New Host** to create a new host.

g. Select **Any** from the Node Locked Host list 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).

3. Install your license(s). On each system, copy the license file(s) to*/var/LifeKeeper/license*, or on each system, run `/opt/LifeKeeper/bin/lkkeyins` and specify the filename (including full path) to the file.

Resource Policy Management

Overview

Resource Policy Management in LifeKeeper Single Server Protection (SSP) provides behavior management of resource local recovery and failover. Resource policies are managed with the **lkpolicy** command line tool (CLI).

LifeKeeper SSP Recovery Behavior

LifeKeeper SSP is designed to monitor individual applications and groups of related applications, periodically performing local recoveries or notifications when protected applications fail. Related applications, by example, are hierarchies where the primary application depends on lower-level storage or network resources. When an application or resource failure occurs, the default behavior is:

1. **Local Recovery**: First, attempt **local** recovery of the resource or application. An attempt will be made to restore the resource or application on the local server without external intervention. If local recovery is successful, then LifeKeeper SSP will not perform any additional action.
2. **Failover**: Second, if a local recovery attempt *fails* to restore the resource or application (or the *recovery kit* monitoring the resource has no support for local recovery), then a ***failover*** will be initiated (see Failover in the Standard Policies section below).

Please see LifeKeeper Single Server Protection Fault Detection and Recovery Scenario for more detailed information about our recovery behavior.

Custom and Maintenance-Mode Behavior via Policies

LifeKeeper SSP supports the ability to set additional policies that modify the default recovery behavior. There are four policies that can be set for individual resources (see the section below about precautions regarding individual resource policies) or for an entire server. **The recommended approach is to alter policies at the server level.**

The available policies are:

Standard Policies

- **Failover** - For LifeKeeper SSP this policy setting can be used to turn on/off resource failover (which results in a reboot).
- **LocalRecovery** - LifeKeeper SSP by default, will attempt to recover protected resources by restarting the individual resource or the entire protected application prior to performing a fail-over (which would be a reboot). This policy setting can be used to turn on/off local recovery.
- **TemporalRecovery** - Normally, LifeKeeper SSP will perform local recovery of a failed resource. If local recovery fails, LifeKeeper SSP will perform a reboot. If the local recovery succeeds, failover (which would be a reboot) will not be performed.

There may be cases where the local recovery succeeds, but due to some irregularity in the server, the local recovery is re-attempted within a short time; resulting in multiple, consecutive local recovery attempts. This may degrade availability for the affected application.

To prevent this repetitive local recovery/failure cycle, you may set a temporal recovery policy. The temporal recovery policy allows an administrator to limit the number of local recovery attempts (successful or not) within a defined time period.

> *Example*: If a user sets the policy definition to limit the resource to three local recovery attempts in a 30-minute time period, LifeKeeper SSP will failover(reboot) when a third local recovery attempt occurs within the 30-minute period.

Defined temporal recovery policies may be turned on or off. When a temporal recovery policy is off, temporal recovery processing will continue to be done and notifications will appear in the log when the policy would have fired; however, no actions will be taken.

Note: It is possible to disable failover and/or local recovery with a temporal recovery policy also in place. This state is illogical as the temporal recovery policy will never be acted upon if failover or local recovery are disabled.

Meta Policies

The "meta" policies are the ones that can affect more than one other policy at the same time. These policies are usually used as shortcuts for getting certain system behaviors that would otherwise require setting multiple standard policies.

- **NotificationOnly** - This mode allows administrators to put LifeKeeper SSP in a "monitoring only" state. **Both** local recovery **and** failover(reboot) **of a resource (or all resources in the case of a server-wide policy) are affected.** The user interface will indicate a **Failure** state if

a failure is detected; but no recovery or failover(reboot) action will be taken. **Note:** The administrator will need to correct the problem that caused the failure manually and then bring the affected resource(s) back in service to continue normal LifeKeeper SSP operations.

Important Considerations for Resource-Level Policies

Resource level policies are policies that apply to a specific resource only, as opposed to an entire resource hierarchy or server.

Example:

app

- IP

- file system

In the above resource hierarchy, app depends on both IP and file system. A policy can be set to disable local recovery or failover of a specific resource. This means that, for example, if the IP resource's local recovery fails and a policy was set to disable failover of the IP resource, then the IP resource will not fail over or cause a failover of the other resources. However, if the file system resource's local recovery fails and the file system resource policy does not have failover disabled, then the entire hierarchy will failover causing a reboot.

Note: It is important to remember that resource level policies apply only to the specific resource for which they are set.

This is a simple example. Complex hierarchies can be configured, so care must be taken when setting resource-level policies.

The lkpolicy Tool

The `lkpolicy` tool is the command-line tool that allows management (querying, setting, removing) of policies on servers running LifeKeeper SSP. `lkpolicy` supports setting/modifying policies, removing policies and viewing all available policies and their current settings. In addition, defined policies can be set on or off, preserving resource/server settings while affecting recovery behavior.

The general usage is :

```
lkpolicy [--list-policies | --get-policies | --set-policy | --remove-
policy] <name value pair data...>
```

The `<name value pair data...>` differ depending on the operation and the policy being manipulated, particularly when setting policies. *For example:* Most on/off type policies only require `-on or --off` switch, but the temporal policy requires additional values to describe the threshold values.

Example lkpolicy Usage

Authenticating With Local and Remote Servers

The `lkpolicy` tool communicates with LifeKeeper SSP servers via an API that the servers expose. This API requires authentication from clients like the `lkpolicy` tool. The first time the `lkpolicy` tool is asked to access a LifeKeeper SSP server, if the credentials for that server are not known, it will ask the user for credentials for that server. These credentials are in the form of a username and password and:

1. Clients must have LifeKeeper SSP admin rights. This means the username must be in the `lkadmin` group according to the operating system's authentication configuration (via pam). It is **not** necessary to run as **root**, but the root user can be used since it is in the appropriate group by default.
2. The credentials will be stored in the *credential store* so they do not have to be entered manually each time the tool is used to access this server.

See Configuring Credentials for SIOS Protection Suite for more information on the credential store and its management with the credstore utility.

An example session with `lkpolicy` might look like this:

```
[root@thor49 ~]# lkpolicy -l -d v6test4
Please enter your credentials for the system 'v6test4'.
Username: root
Password:
Confirm password:
Failover
LocalRecovery
TemporalRecovery
NotificationOnly
[root@thor49 ~]# lkpolicy -l -d v6test4
Failover
LocalRecovery
TemporalRecovery
NotificationOnly
[root@thor49 ~]#
```

Listing Policies

lkpolicy --list-policy-types

Showing Current Policies

lkpolicy --get-policies

lkpolicy --get-policies tag=*

lkpolicy --get-policies --verbose tag=mysql* # all resources starting with mysql

lkpolicy --get-policies tag=mytagonly

Setting Policies

lkpolicy --set-policy Failover --off

lkpolicy --set-policy Failover --on tag=myresource

lkpolicy --set-policy Failover --on tag=*

lkpolicy --set-policy LocalRecovery --off tag=myresource

lkpolicy --set-policy NotificationOnly --on

lkpolicy --set-policy TemporalRecovery --on recoverylimit=5 period=15

lkpolicy --set-policy TemporalRecovery --on --force recoverylimit=5 period=10

Removing Policies

lkpolicy --remove-policy Failover tag=steve

***Note**: NotificationOnly is a policy alias. Enabling NotificationOnly is the equivalent of disabling the corresponding LocalRecovery and Failover policies.*

Verifying LifeKeeper Single Server Protection Installation

You can verify that the LifeKeeper Single Server Protection packages were installed correctly by entering the following at the command line:

```
rpm -V <package name>
```

Note: If the package is installed correctly, no output will be displayed by this command.

To perform a query from the command line, type

```
rpm -qi <package name>
```

Note: The expected output for this command is the package information.

LifeKeeper Single Server Protection Administration Overview

LifeKeeper Single Server Protection does not require administration during operation. LifeKeeper Single Server Protection works automatically to monitor protected resources and to perform the specified recovery actions if a fault should occur. You use the LifeKeeper Single Server Protection GUI in these cases:

- **Resource and hierarchy definition**. LifeKeeper Single Server Protection provides these interface options:
  - LifeKeeper Single Server Protection GUI.
  - LifeKeeper Single Server Protection command line interface.
- **Resource monitoring.** The LifeKeeper Single Server Protection GUI provides access to resource status information and to the LifeKeeper Single Server Protection logs.
- **Manual intervention.** You may need to stop servers or specific resources for maintenance or other administrative actions. The LifeKeeper Single Server Protection GUI provides menu functions that allow you to bring specific resources in and out of service. Once applications have been placed under LifeKeeper Single Server Protection, they should be started and stopped only through these LifeKeeper Single Server Protection interfaces. Starting and stopping LifeKeeper Single Server Protection is done through the command line only.

See Administration Tasks, GUI Tasks, and Maintenance Tasks in the SPS for Linux documentation for detailed instructions on performing administration, configuration, and maintenance operations including the creation of resource hierarchies.

Note: All LifeKeeper executable scripts and programs run via the command line require super user authority.
A super user granted permissions by running the "su" or "sudo" command is able to execute LifeKeeper commands. However, SIOS Technology Corp has tested executing LifeKeeper commands via the root user only.

LifeKeeper Single Server Protection Administration Overview

LifeKeeper Single Server Protection does not require administration during operation. LifeKeeper Single Server Protection works automatically to monitor protected resources and to perform the specified recovery actions if a fault should occur. You use the LifeKeeper Single Server Protection GUI in these cases:

- **Resource and hierarchy definition**. LifeKeeper Single Server Protection provides these interface options:
  - LifeKeeper Single Server Protection GUI.
  - LifeKeeper Single Server Protection command line interface.
- **Resource monitoring.** The LifeKeeper Single Server Protection GUI provides access to resource status information and to the LifeKeeper Single Server Protection logs.
- **Manual intervention.** You may need to stop servers or specific resources for maintenance or other administrative actions. The LifeKeeper Single Server Protection GUI provides menu functions that allow you to bring specific resources in and out of service. Once applications have been placed under LifeKeeper Single Server Protection, they should be started and stopped only through these LifeKeeper Single Server Protection interfaces. Starting and stopping LifeKeeper Single Server Protection is done through the command line only.

See GUI Tasks and Maintenance Tasks for detailed instructions on performing LifeKeeper Single Server Protection administration, configuration and maintenance operations.

Starting LifeKeeper Single Server Protection

All LifeKeeper Single Server Protection software is installed in the directory */opt/LifeKeeper*.

When you have completed all of the verification tasks, you are ready to start LifeKeeper Single Server Protection on both servers. This section provides information for starting the LifeKeeper Single Server Protection server daemon processes. The LifeKeeper Single Server Protection GUI application is launched using a separate command and is described in Configuring the LifeKeeper Single Server Protection GUI. LifeKeeper Single Server Protection provides a command line interface that starts and stops the LifeKeeper Single Server Protection daemon processes. These daemon processes must be running before you start the LifeKeeper Single Server Protection GUI.

Starting LifeKeeper Single Server Protection Processes

If LifeKeeper Single Server Protection is not currently running on your system, type the following command as the user root on all servers:

```
/etc/init.d/lifekeeper start
```

Following the delay of a few seconds, an informational message is displayed.

See the LCD(1M) man page by entering **man LCD** at the command line for details on the **`/etc/init.d/lifekeeper start`** command.

Enabling Automatic LifeKeeper Single Server Protection Restart

While the above command will start LifeKeeper Single Server Protection, it will need to be performed each time the system is re-booted. If you would like LifeKeeper Single Server Protection to start automatically when server boots up, type the following command:

```
chkconfig lifekeeper on
```

See the `chkconfig` man page for further information.

Stopping LifeKeeper Single Server Protection

If you need to stop LifeKeeper Single Server Protection, type the following command as *root* to stop it:

/etc/init.d/lifekeeper stop

This command halts all LifeKeeper Single Server Protection daemon processes on the server being administered if they are currently running. Messages similar to the following are displayed:

```
# /etc/init.d/lifekeeper stop

STOPPING LIFEKEEPER AT: Thu Nov 10 16:56:22 EST 2011
 Skipping remove of OSU resource oracle.
 Skipping remove of OSU resource /oracle/data.
 /opt/LifeKeeper/bin/perform_action -G -t /lv_jailbird1 -a remove -- -m
 Thu Nov 10 16:56:28 EST 2011 remove: BEGIN remove of file system /lv_jailbird1
 LifeKeeper: unmounting file system /lv_jailbird1
     umount /lv_jailbird1
 LifeKeeper: file system /lv_jailbird1 successfully unmounted
 Thu Nov 10 16:56:29 EST 2011 remove: END remove of file system /lv_jailbird1
(err=0)
 /opt/LifeKeeper/bin/perform_action -G -t /dev/mapper/vg_jailbird-lv_jailbird1 -a
remove -- -m
 LIFEKEEPER NOW STOPPED AT: Thu Nov 10 16:56:38 EST 2011
```

See lkstop under the LCD(1M) man page for additional details.

Disabling Automatic LifeKeeper Single Server Protection Restart

If you do not want LifeKeeper Single Server Protection to automatically restart when the system is restarted, type the following command:

chkconfig lifekeeper off

See the chkconfig man page for further information.

Viewing LifeKeeper Single Server Protection Processes

To see a list of all LifeKeeper Single Server Protection core daemon processes currently running, type the following command:

ps -ef | grep LifeKeeper | grep -w bin | grep -v lklogmsg

An example of the output is provided below:

```
root 11663 11662 0 14:03 pts/0 00:00:00 /bin/bash /etc/redhat-lsb/lsb_start_daemon
/opt/LifeKeeper/sbin/runsvdir -P /opt/LifeKeeper/etc/service log: runit just
```

starte
d.....
.....

root 11666 11663 0 14:03 pts/0 00:00:00 /bin/bash -c ulimit -S -c 0 >/dev/null 2> &1 ; /opt/LifeKeeper/sbin/runsvdir -P /opt/LifeKeeper/etc/service log: runit just
starte
d.....
.....

root 11880 11873 0 14:03 ? 00:00:00 /opt/LifeKeeper/bin/lk_logmgr -l/opt/LifeKeeper/out -d/etc/default/LifeKeeper

root 12240 11877 0 14:04 ? 00:00:00 /opt/LifeKeeper/bin/lcm

root 12247 11879 0 14:04 ? 00:00:00 /opt/LifeKeeper/bin/ttymonlcm

root 12250 11876 0 14:04 ? 00:00:00 /opt/LifeKeeper/bin/lcd

root 12307 11874 0 14:04 ? 00:00:00 /opt/LifeKeeper/bin/lkcheck

root 12311 11875 0 14:04 ? 00:00:00 /opt/LifeKeeper/bin/lkscsid

root 12325 11871 0 14:04 ? 00:00:00 /opt/LifeKeeper/bin/lkvmhad

root 12335 12330 0 14:04 ? 00:00:00 /opt/LifeKeeper/bin/perl /opt/LifeKeeper/htdoc/cgi-bin/DoRequest.fcgi

The run state of LifeKeeper Single Server Protection can be determined via the following command:

/opt/LifeKeeper/bin/lktest

If LifeKeeper Single Server Protection is running it will output something similar to the following:

F S UID PID PPID C CLS PRI NI SZ STIME TIME CMD

4 S root 12240 11877 0 TS 39 -20 6209 14:04 00:00:00 lcm

4 S root 12247 11879 0 TS 39 -20 30643 14:04 00:00:00 ttymonlcm

4 S root 12250 11876 0 TS 29 -10 9575 14:04 00:00:00 lcd

If LifeKeeper Single Server Protection is not running, then nothing is output and the command exists with a 1.

Note: There are additional LifeKeeper Single Server Protection processes running that start, stop, and monitor the LifeKeeper Single Server Protection core daemon processed along with those required for the Graphical User Interface (GUI). See Viewing LifeKeeper Single Server Protection Controlling Processes and Viewing LifeKeeper GUI Server Processes for a list of the processes. Additionally, most LifeKeeper Single Server Protection processes have a child lklogmsg to capture and log any unexpected output.

Viewing LifeKeeper Single Server Protection GUI Server Processes

To verify that the LifeKeeper Single Server Protection GUI Server is running, type the following command:

```
ps -ef | grep runGuiSer
```

You should see output similar to the following:

root 2805 1 0 08:24 ? 00:00:00 sh /opt/LifeKeeper/bin/runGuiServer

To see a list of the other GUI Server daemon processes currently running, type the following command:

```
ps -efw | grep S_LK
```

You should see output similar to the following:

root 819 764 0 Oct16 ? 00:00:00 java -Xint -Xss3M -DS_LK=true -Djava.rmi.server.hostname=wake -Dcom.steeleye.LifeKeeper.rmiPort=82 -Dcom.steeleye.LifeKeeper.LKROOT=/opt/LifeKeeper -DGUI_RMI_REGISTRY=internal -DGUI_WEB_PORT=81 com.steeleye.LifeKeeper.beans.S_LK

To verify that the LifeKeeper Single Server Protection GUI Server Administration Web Server is running type the following command:

```
ps -ef|grep steeleye-light | egrep -v "lklogmsg|runsv"
```

You should see output similar to the following:

```
root 12330 11872 0 14:04 ? 00:00:00 /opt/LifeKeeper/sbin/steeleye-lighttpd -D -f/opt/LifeKeeper/etc/lighttpd/lighttpd.conf
```

Viewing LifeKeeper Single Server Protection Controlling Processes

To verify that the LifeKeeper Single Server Protection controlling processes are running, type the following command:

```
ps -ef | grep runsv
```

You should see output similar to the following:

root 11663 11662 0 14:03 pts/0 00:00:00 /bin/bash /etc/redhat-lsb/lsb_start_daemon /opt/LifeKeeper/sbin/runsvdir -P /opt/LifeKeeper/etc/service log: runit just starte
d.....

root 11666 11663 0 14:03 pts/0 00:00:00 /bin/bash -c ulimit -S -c 0 >/dev/null 2>&1 ; /opt/LifeKeeper/sbin/runsvdir -P /opt/LifeKeeper/etc/service log: runit just starte d.....

root 11667 11666 0 14:03 pts/0 00:00:00 /opt/LifeKeeper/sbin/runsvdir -P /opt/LifeKeeper/etc/service log: runit just starte d.....

root 11871 11667 0 14:03 ? 00:00:00 /opt/LifeKeeper/sbin/runsv lkvmhad

root 11872 11667 0 14:03 ? 00:00:00 /opt/LifeKeeper/sbin/runsv steeleye-lighttpd

root 11873 11667 0 14:03 ? 00:00:00 /opt/LifeKeeper/sbin/runsv lk_logmgr

root 11874 11667 0 14:03 ? 00:00:00 /opt/LifeKeeper/sbin/runsv lkcheck

root 11875 11667 0 14:03 ? 00:00:00 /opt/LifeKeeper/sbin/runsv lkscsid

root 11876 11667 0 14:03 ? 00:00:00 /opt/LifeKeeper/sbin/runsv lcd

root 11877 11667 0 14:03 ? 00:00:00 /opt/LifeKeeper/sbin/runsv lcm

root 11878 11667 0 14:03 ? 00:00:00 /opt/LifeKeeper/sbin/runsv lkguiserver

root 11879 11667 0 14:03 ? 00:00:00 /opt/LifeKeeper/sbin/runsv ttymonlcm

These processes start, stop, and monitor LifeKeeper Single Server Protection core daemon processes and must be running to start LifeKeeper Single Server Protection. These processes are configured by default to start when the system boots and this behavior should not be altered.

Enabling VMware HA Integration with LifeKeeper Single Server Protection

By default LifeKeeper Single Server Protection integration with VMware HA is disabled when installed on a VMware VM. To enable integration requires the following steps:

1. Installation of VMware tools in the LifeKeeper Single Server Protection VM.
2. Edit */etc/default/LifeKeeper* and change the VMware HA integration tunable `HA_DISABLE` value form 1 to 0.
3. Restart LifeKeeper Single Server Protection. If LifeKeeper Single Server Protection is currently running, it must be stopped and restarted to pick up the above change in `/etc/default/LifeKeeper`.
4. Installation of the SteelEye Management Console (optional step).

Increasing the Log File Size

When LifeKeeper Single Server Protection starts, it allocates the maximum space required for its log files. The size, or the space allocated for the LifeKeeper Single Server Protection log files, is a tunable parameter located in */etc/default/LifeKeeper*.

```
LOGFILE=log:2048

LOGFILE=TTYLCM:256

LOGFILE=LCM:1024

LOGFILE=LCD:512

LOGFILE=remote_execute:512

LOGFILE=SNMP:512

LOGFILE=NOTIFY:512
```

The numeric argument dictates the size of the log file and represents the number of 512 byte blocks.

CAUTION: The size of the log files must be less than 2 GB.

Enabled VMware HA Fault Detection and Recovery Scenario

The ability to provide detection and alarming for problems within an application is critical to building the best total fault resilient solution. Since every specific application varies on the mechanism and format of failures, no one set of generic mechanisms can be supplied. In general, however, many application configurations can rely on the Core system error detection provided within LifeKeeper Single Server Protection. This topic demonstrates the power of LifeKeeper Single Server Protection's core facilities.

Below is a recovery scenario to demonstrate how LifeKeeper Single Server Protection provides fault detection and recovery when an application fails.

1. LifeKeeper Single Server Protection will first attempt recovery by trying to restart the application.
2. If the recovery succeeds, the application should continue to run normally.
3. If the recovery attempt fails:
    a. If the LifeKeeper Single Server Protection recovery attempt fails, and LifeKeeper Single Server Protection is installed in a VMware guest OS with HA enabled (HA_DISABLE=0 in /etc/default/LifeKeeper), then LifeKeeper Single Server Protection will trigger VMware HA by withholding the heartbeat that LifeKeeper Single Server Protection sends down to the Application Monitoring Interface. VMware HA will then respond by restarting the server.

b. If the LifeKeeper Single Server Protection recovery attempt fails, and LifeKeeper Single Server Protection is not installed in a VMware guest OS or is installed in a VMware guest OS but has HA disabled (HA_DISABLE=1 in /etc/default/LifeKeeper), then a system reboot will be forced.

Optionally, LifeKeeper Single Server Protection can be placed in **Notification Only** mode. In this mode the automatic triggering of a system reboot is disabled (see the section VMware HA and Notification Only Mode below). In **Notification Only** mode you must log into the system and correct the issue that caused failure.

VMware HA and Notification Only Mode

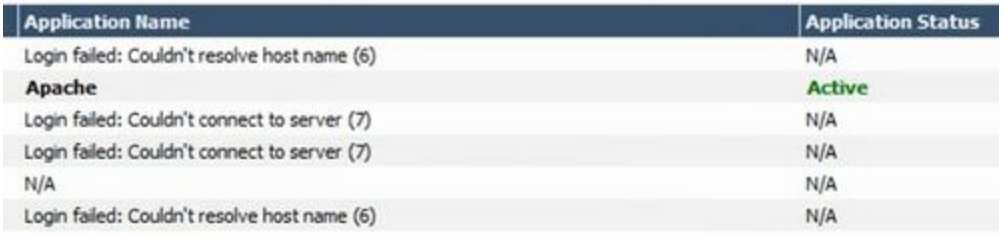
1. In **Notification Only** mode with HA enabled in the VMware guest OS and the LifeKeeper SSP vCenter plugin installed, when a failure is detected, LifeKeeper Single Server Protection will not attempt to restart the application. Instead, the resource will be marked as **Failed**. The vCenter plugin dashboard view status screen will show failure (**Application Status: Failed**).

Application Name	Application Status
Login failed: Couldn't resolve host name (6)	N/A
Apache	**Failed**
Login failed: Couldn't connect to server (7)	N/A
Login failed: Couldn't connect to server (7)	N/A
N/A	N/A
Login failed: Couldn't resolve host name (6)	N/A

2. Log in to the server and correct the issue that caused the failure.
3. Open the **LifeKeeper Admin Console** either through the **CLI** or by clicking on the protected virtual machine within the **vSphere Client User Interface**.

4. Bring the application back in service.
5. Go to the **dashboard** view within the **vSphere Client User Interface**.

6. Click **Refresh. Application status** goes back to **Active**.



Application Name	Application Status
Login failed: Couldn't resolve host name (6)	N/A
Apache	**Active**
Login failed: Couldn't connect to server (7)	N/A
Login failed: Couldn't connect to server (7)	N/A
N/A	N/A
Login failed: Couldn't resolve host name (6)	N/A

LifeKeeper Single Server Protection Heartbeat with VMware HA

The LifeKeeper Single Server Protection heartbeat is the signal sent to VMware HA (every 10 seconds if running in a VMware guest OS and if HA is enabled) indicating that the protected applications are OK. If an application fails, LifeKeeper Single Server Protection will first attempt to recover the application. If recovery fails, LifeKeeper Single Server Protection will withhold the heartbeat, which instructs VMware HA to reboot the VM.

Maintaining a LifeKeeper Single Server Protection Protected System

When performing system or application maintenance on a LifeKeeper Single Server Protection-protected server, you should either stop LifeKeeper Single Server Protection monitoring or place the protected resources into maintenance mode. This will stop LifeKeeper Single Server Protection from interfering with the system and application maintenance tasks by disabling both application recovery and triggering of VMware HA failure events.

To stop and restart LifeKeeper Single Server Protection:

1. **Stop LifeKeeper Single Server Protection**. Use the command **`/etc/init.d/lifekeeper stop-daemons`** to stop LifeKeeper Single Server Protection. The resources will remain running but will no longer be monitored by LifeKeeper Single Server Protection. Any failure will have to be handled manually.
2. **Perform maintenance**. Perform the necessary maintenance.
3. **Start LifeKeeper Single Server Protection.** Use the command **`/etc/init.d/lifekeeper start`** to start LifeKeeper Single Server Protection. Your resources are now protected.

Alternatively, place the resources in maintenance (a.k.a., notification only) mode:

1. **Place all resources in maintenance mode.** Use the command **`/opt/LifeKeeper/bin/lkpolicy -s NotificationOnly --On`**. Resources will not be recovered and VMware HA failure events will not be triggerred.
2. **Perform maintenance**. Perform the necessary maintenance.

3. **Turn off maintenance mode.** Use the command **`/opt/LifeKeeper/bin/lkpolicy -s NotificationOnly --Off`**. Resources are now protected.

Creating Resource Hierarchies

1. There are four ways to begin creating a resource hierarchy.
    - Right-click on a server icon to bring up the server context menu, then click on **Create Resource Hierarchy**.
    - On the global toolbar, click on the **Create Resource Hierarchy** button.
    - On the server context toolbar, if displayed, click on the **Create Resource Hierarchy** button.
    - On the **Edit** menu, select **Server**, then click on **Create Resource Hierarchy**.
2. A dialog entitled **Create Resource Wizard** 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. Select the **Switchback Type** and click **Next**.
4. Select the **Server** and click **Next**. **Note**: If you began from the server context menu, the server will be determined automatically from the server icon that you clicked on, and this step will be skipped.
5. Continue through the succeeding dialogs, entering whatever data is needed for the type of resource hierarchy that you are creating.

LifeKeeper Single Server Protection Application Resource Hierarchies

If you install LifeKeeper Single Server Protection without any recovery kits, the Select Recovery Kit list includes options for File System,Generic Application, and IP 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 File System Resource Hierarchy
- Creating a Generic Application Resource Hierarchy

The IP Recovery Kit is documented in the IP Recovery Kit Administration Guide.

Recovery Kit Options

Each optional recovery kit that you install adds entries to the Select Recovery Kit list; for example, you may see Oracle, Apache, and NFS Recovery Kits. Refer to the Administration Guide that accompanies each recovery kit for directions on creating the required resource hierarchies.

Creating a File System Resource Hierarchy

Use this option to protect a file system only.

1. There are four ways to begin creating a file system resource hierarchy.
    - Right-click on a server icon to bring up the server context menu, then click on **Create Resource Hierarchy**.
    - On the server context toolbar, if displayed, click on the **Create Resource Hierarchy** button.
2. A dialog entitled *Create Resource Wizard* will appear with a **Recovery Kit** list. Select *File System Resource* and click **Next**.
3. Select the **Switchback Type** and click **Next**. Note: *This setting is not relevant for LifeKeeper Single Server Protection.*
4. The *Create gen/filesys Resource* dialog will now appear. Select the **Mount Point** for the file system resource hierarchy and click **Next**.

    Note: In order for a mount point to appear in the choice list, the mount point must be currently mounted.
5. LifeKeeper Single Server Protection creates a default **Root Tag** for the file system resource hierarchy. (This is the label used for this resource in the status display). You can select this root tag or create your own, then click **Next**.
6. Click **Create Instance**. A window will display a message indicating the status of the instance creation.
7. Click **Next**. A window will display a message that the file system hierarchy has been created successfully.
8. At this point, click **Cancel** to return to the GUI.

Creating a Generic Application Resource Hierarchy

The generic application recovery kit is used to protect a custom application that has no specific associated recovery kit. Sample scripts are provided in `/opt/LifeKeeper/lkadm/subsys/gen/app/templates`. Copy these samples to another directory before customizing and testing them.

Note: For applications that depend on other resources such as a file system or IP address, create each of these resources separately and use **Create Dependency** to create the appropriate dependencies.

1. Begin creating a generic application resource hierarchy.
    - Right-click on a server icon to bring up the server context menu, then click on **Create Resource Hierarchy**

- On the server context toolbar, if displayed, click on the **Create Resource Hierarchy** button

2. A dialog entitled **Create Resource Wizard** will appear with a **Recovery Kit** list. Select **Generic Application** and click **Next**.
3. Select the **Switchback Type** and click **Next**. **Note:** This setting is not relevant for LifeKeeper Single Server Protection.
4. On the next dialog, enter the path to the **Restore Script** for the application and click **Next**. This is the command that starts the application. A template restore script, restore.template, is provided in the templates directory. The restore script must not impact applications that are already started.
5. Enter the path to the **Remove Script** for the application and click **Next**. This is the command that stops the application. A template remove script, remove.template, is provided in the templates directory.
6. Enter the path to the **QuickCheck Script** for the application and click **Next**. This is the command that monitors the application. If the script detects a problem, it should exit with a non-zero value in order to trigger a recovery or restart (see Fault Detection and Recovery Scenario).
7. Enter the path to the **Local Recovery Script** for the application and click **Next**. This is the command that attempts to restore a failed application. A template recover script, recover.template, is provided in the templates directory.
8. Enter any **Application Information** and click **Next**. This is optional information about the application that may be used by the restore, remove, quickCheck, and recover scripts.
9. Select either **Yes** or **No** for **Bring Resource In Service**, and click **Next**. Selecting **No** will cause the resource state to be set to **OSU** following the create; selecting **Yes** will cause the previously provided restore script to be executed. For applications depending upon other resources such as a file system or IP address, select **No** if you have not already created the appropriate dependent resources.
10. Enter the **Resource Tag**, which is a unique name for the resource instance. (This is the label you will see for this resource in the status display.)
11. Click **Create Instance** to start the creation process. A window will display a message indicating the status of the instance creation.
12. Click **Next**. A window will display a message that the hierarchy has been created successfully.
13. At this point, click **Cancel** to return to the GUI.

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.

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

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.

Creating 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. Right-click on the icon for the parent resource to which you want to add a child dependency. When the resource context menu appears, click **Create Dependency**
2. Select a **Child Resource Tag** from the drop-down box of existing, 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.
   - Any resource that is not in service, if the parent resource is in service.

   Click **Next** to proceed to the next dialog.
3. The dialog will then confirm that you have selected the appropriate parent and child resource tags for your dependency creation. Click **Create Dependency** to create the dependency on the server.
4. 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.

Deleting a Resource Dependency

1. Right-click on the icon for the parent resource from which you want to delete a child dependency. When the resource context menu appears, click **Delete Dependency**.
2. Select the **Child Resource Tag** from the drop down box. This should be the tag 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 **Delete Dependency** to delete the dependency on the server.

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

1. Right-click on the icon for a resource in the hierarchy that you want to delete. When the resource context menu appears, click **Delete Resource Hierarchy**.
2. The dialog will display a message verifying the hierarchy you have specified for deletion. Click **Delete** to perform the action.
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.

Removing LifeKeeper Single Server Protection

You can uninstall the LifeKeeper Single Server Protection product simply by running the included rmlk utility:

```
/opt/LifeKeeper/bin/rmlk
```

This will remove all SIOS product rpms and remove the */opt/LifeKeeper* directory from the system. **Use with caution.**

Quick Service Protection (QSP) Recovery Kit

Introduction

QSP Recovery Kit provides a simplified method to protect OS service. With the QSP Recover Kit users can easily create a LifeKeeper resource instance to protect an OS service provided that service can be started and stopped by the OS service command. The service can also be protected via the Generic Application Recovery Kit, but the use of that kit requires code development whereas the QSP Recovery Kit does not. Also, by creating a dependency relationship, protected services can be started and stopped in conjunction with the application that requires the service. The application is protected by another LifeKeeper resource instance not via the QSP resource.

However, the QSP Recovery Kit quickCheck only perform simple health (using the “status” action of the service command). So, QSP doesn't guarantee that the service is provided, or process is functioning in fact. If complicated starting and/or stopping is necessary, or more robust health checking operations are necessary, the using a Generic Application is recommended.

Requirements

The service to be protected by the QSP Recovery Kit needs to meet the following requirements.

- It must support start and stop action via the OS service command. Also, it must returns 0 when the start and stop action succeeds.
- To perform heath checking the service must support the status action via the OS service command. If it does not support the status action then quickCheck health check operations must be disabled. Also, it must return 0 when the status action succeeds.
- The name of the service to be protected must not exceed 256 characters in length and contain only alphanumeric characters.

The service to be protected by the QSP resource must be running (started) before attempting a resource create. Please notice that some services which are already supplied with dedicated Recovery Kit are not target of QSP (hereinafter referred as “the Services not targeted by QSP protection”) and cannot be protected by QSP Recovery Kit.

Create QSP Resource Hierarchy

This option is used to protect OS services via the QSP Recovery Kit.

1. There are 4 methods to start the creation of QSP resource instance.
   - Right click the server icon to show the server context menu, and select [Create Resource Hierarchy].
   - Click the [Create Resource Hierarchy] button at on the Global tool bar.
   - Click the [Create Resource Hierarchy] button on the Server context tool bar if displayed.
   - From the [Edit] menu select [Server] then [Create Resource Hierarchy]
2. A dialogue box titled [Create Resource Wizard] is displayed. In the [Recovery Kit] drop down is a list of available resource types to create. Select Quick Service Protection and click [Next].
3. Select [Switchback Type] and click [Next].
4. Select [Server] and click [Next]

Note: If the create was started via the server context menu, this step is skipped because the server is known based on the start context (defaults to name of the server on which the create process started)..

5. The next dialog box contains a drop down of the available services [Service Name] that can be protected.

Select service to be protected and click [Next].
Note: The list may not show the service if it is not running. In this case, click [Cancel] to discontinue the process, and start service. Once the service is running restart the create process. Also, the list will not show the Services not targeted by QSP protection.

6. In the next dialog box the quickCheck action is configured.To enable the quickCheck monitoring function, select [enable]. To disable it, select [disable]. Click [Next] to continue. The quickCheck action can be changed at any time.

Note: If the selected service does not support the “status” action via the OS service command, then set the quickCheck action to “disabled” because the QSP Recovery Kit cannot monitor the service state.

7. Input the [Resource Tag]. This is a unique name for the resource instance.
(This is the label that uniquely identifies the resource instance and is used whenever displaying LifeKeeper protected resource instances in UI.)
8. Click [Create Instance] to start the creation process. The status of the resource instance creation is displayed in the status window.
9. Click [Next] to display the resource extension dialog. Click [Next] to begin the extension process or click [Cancel] to go back to GUI. When [Cancel] is clicked, an alert is displayed that the hierarchy exist on only one server, and protection by LifeKeeper is not available at this time.

Extending QSP Resource Hierarchy

This function, as explained in the section Extending Resource Hierarchies, starts automatically after finishing the Create QSP Resource Hierarchy (URL) process, or, from right clicking on an existing QSP resource and selecting [Extend Resource Hierarchy]. After finishing the pre-extend process, then, complete the following steps.

1. Select [Resource Tag] provided by LifeKeeper, or, input unique tag for the resource hierarchy on the target server.
2. Click [Extend] to start extension process. The status of the extension process is displayed in the dialogue box, and when it is finished it will show a message indicating the hierarchy is correctly extended. If the hierarchy is to be extended to another server click [Next Server], otherwise click [Finish] to complete the extension. If [Next Server] is selected, the extension operation is repeated.
3. When [Finish] is clicked the integrity of the hierarchy is checked. If any problems are detected the extension is reversed. To complete the verification and close the dialog box click [Done].

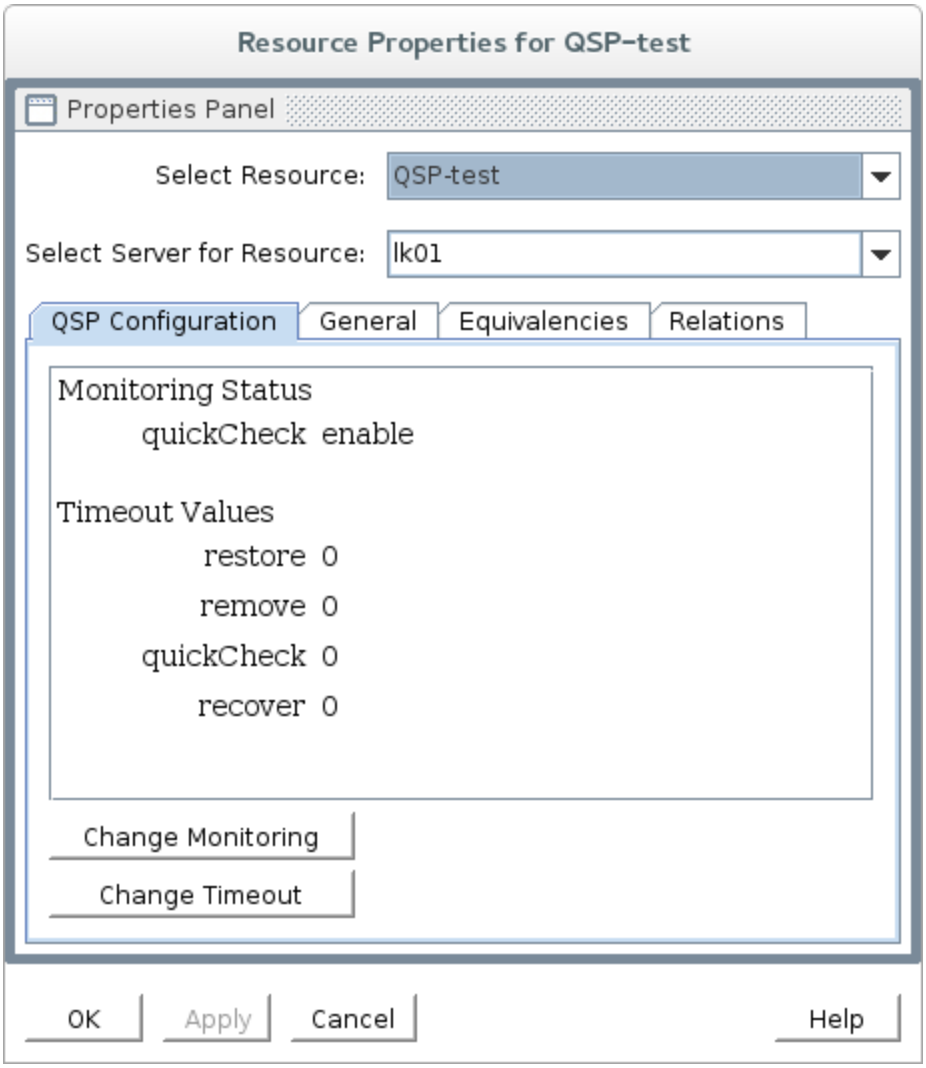
QSP Resource Configuration

The following parameters are unique to each QSP resource instance are available for modification.

Set Up Items		Default Value	Description
Monitoring	quickCheck	Specified when creating the resource	Set to enable the checking of the status of the service or to disable / skip the monitoring function
	restore	0	Specify the restore timeout (unit: second). If set to 0 no timeout occurs when restoring the resource instance.
	remove	0	Specify the remove timeout (unit: second). If set to 0 no timeout occurs when removing the resource instance.
Time Out			Specify the quickCheck timeout (unit: second).
	quickCheck	0	If set to 0 no time out occurs when performing health checking of the resource instance.
	recover	0	Specify the recover timeout (unit: second). If set to 0 no timeout occurs during recovery of the resource instance.

Checking / Changing of the set value is possible from the [QSP Configuration] tab by Display Resource Properties. and must be performed on each node in the hierarchy.

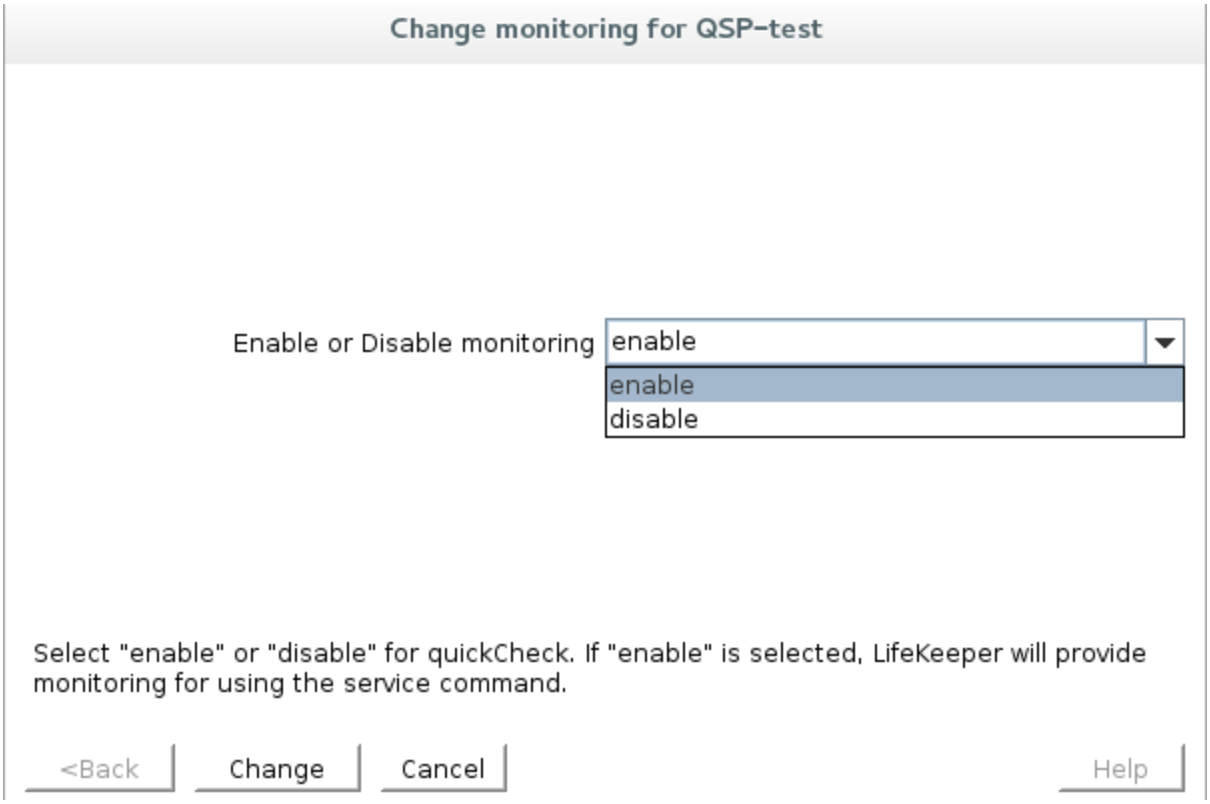
If the quickCheck function is disabled, quickCheck and recover of timeouts are not displayed and thus cannot be changed.



How to change the Monitoring function

1. Display the [QSP Configuration] tab of the resource properties, and click [Change quick-Check]
2. Select [enable] to enable quickCheck, or, [disable] to disable it
3. Clicking [Change] starts the change process, and display change process message.
4. Finish by clicking [Done].

Note: Modification of these values is a per node operation. If the same change is needed on another node, then the process must be repeated on that node.



How to change Timeout Value

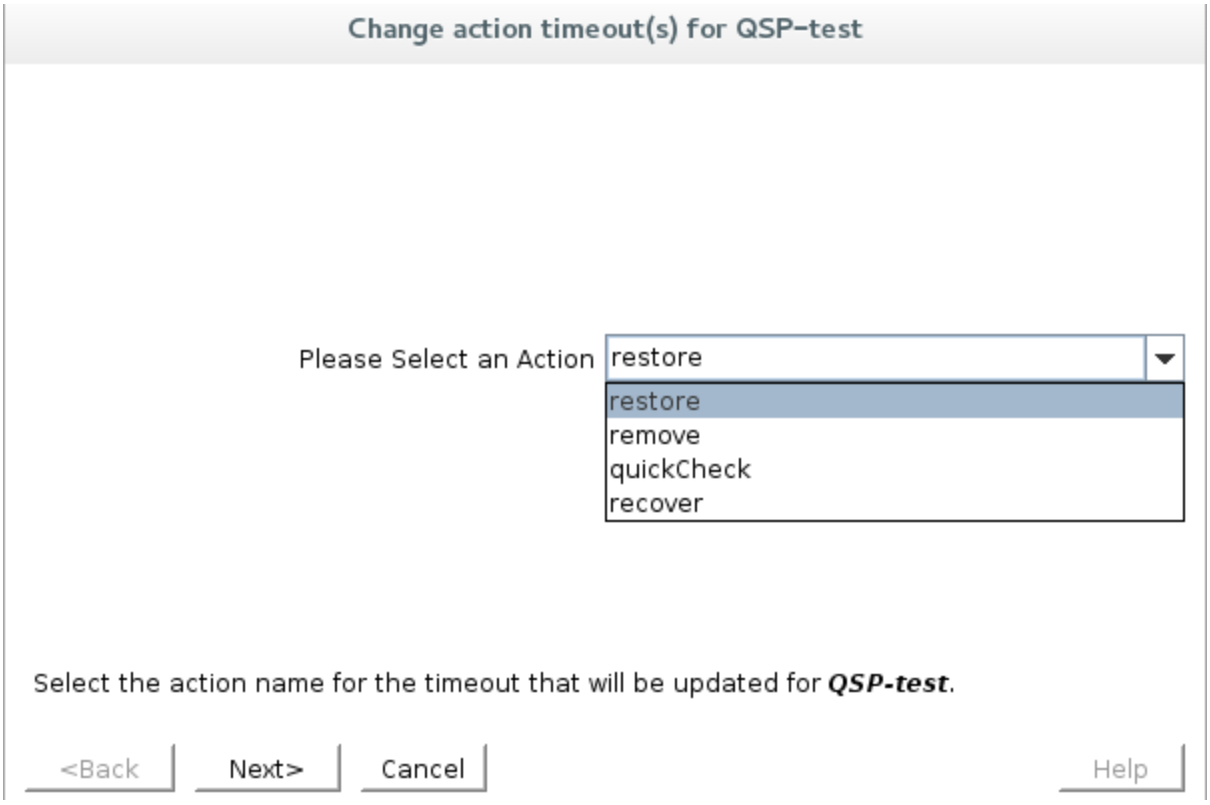
1. Display the [QSP Configuration] tab of resource properties, and click [Change Timeout]
2. Select the timeout action to be changed (restore, remove, quickCheck or recover), and click [Next]

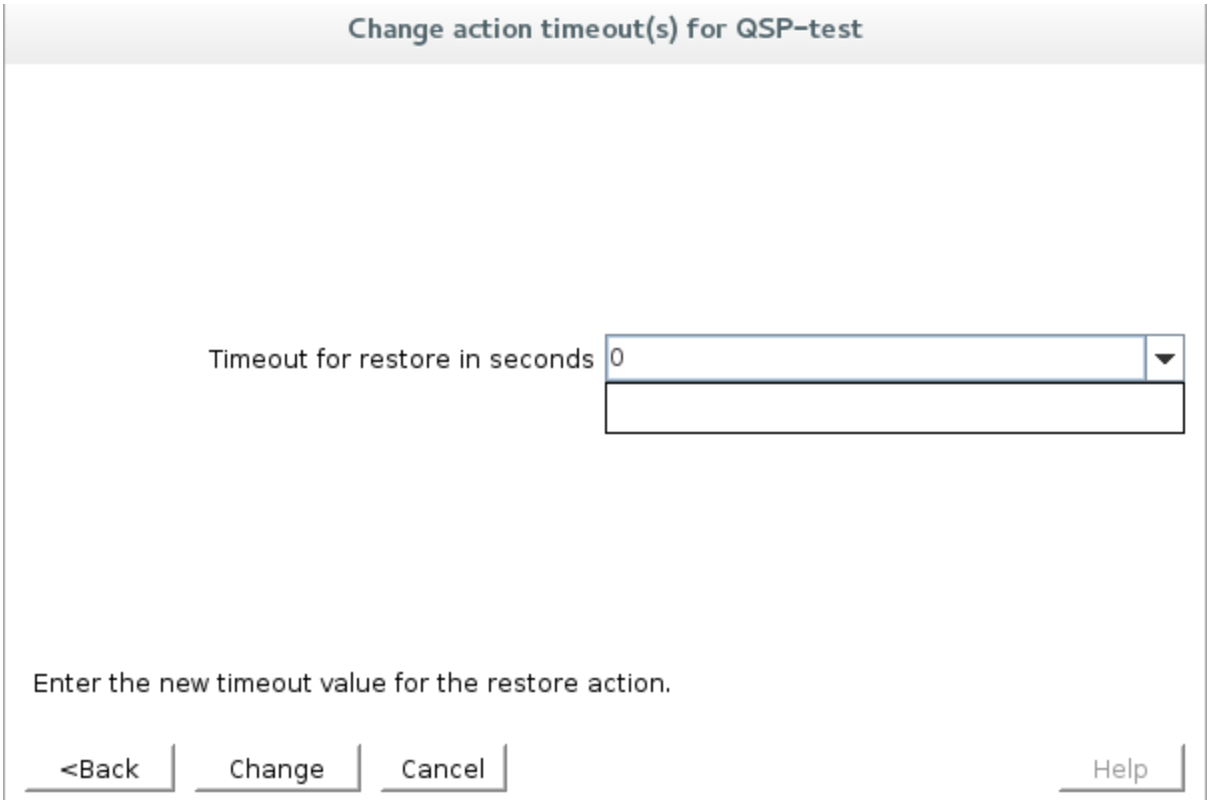
Note: [quickCheck] and [recover] timeouts are not displayed in the select list if the monitoring function is disabled.

3. Input the timeout value seconds.

Note: Input decimal numbers only. Non numerical characters are invalid.

4. Clicking [Change] starts the timeout change process and displays change process messages.
5. Finish by clicking [Done]
6. Note: Modification of these values is a per node operation. If the same change is needed on another node, then the process must be repeated on that node.


Change action timeout(s) for QSP-test
Please Select an Action
restore
restore
remove
quickCheck
recover
Select the action name for the timeout that will be updated for QSP-test.
<Back
Next>
Cancel
Help



LifeKeeper API for Monitoring

1. Introduction

The LifeKeeper API for Monitoring can obtain the operational status of LifeKeeper nodes and their protected resources by making status inquiries to the available nodes in the LifeKeeper cluster.

2.Summary

This document describes the LifeKeeper API for Monitoring (hereinafter referred to as the API) and is targeted for developers who manage the resource protected by LifeKeeper.

By using the API, the information supplied by the lcdstatus command is obtained through CGI script and the lighttpd module. By using this API, users can determine the current status of the LifeKeeper nodes and resources without logging-in to LifeKeeper servers.

The API can supply the following information.

- LifeKeeper node status is the node alive and processing or down
- Communication path status between nodes in the cluster, are communication path(s) up or down
- Status of protected resources

To get the detailed status of any abnormal condition requires logging-in to LifeKeeper GUI or checking the LifeKeeper log as necessary.

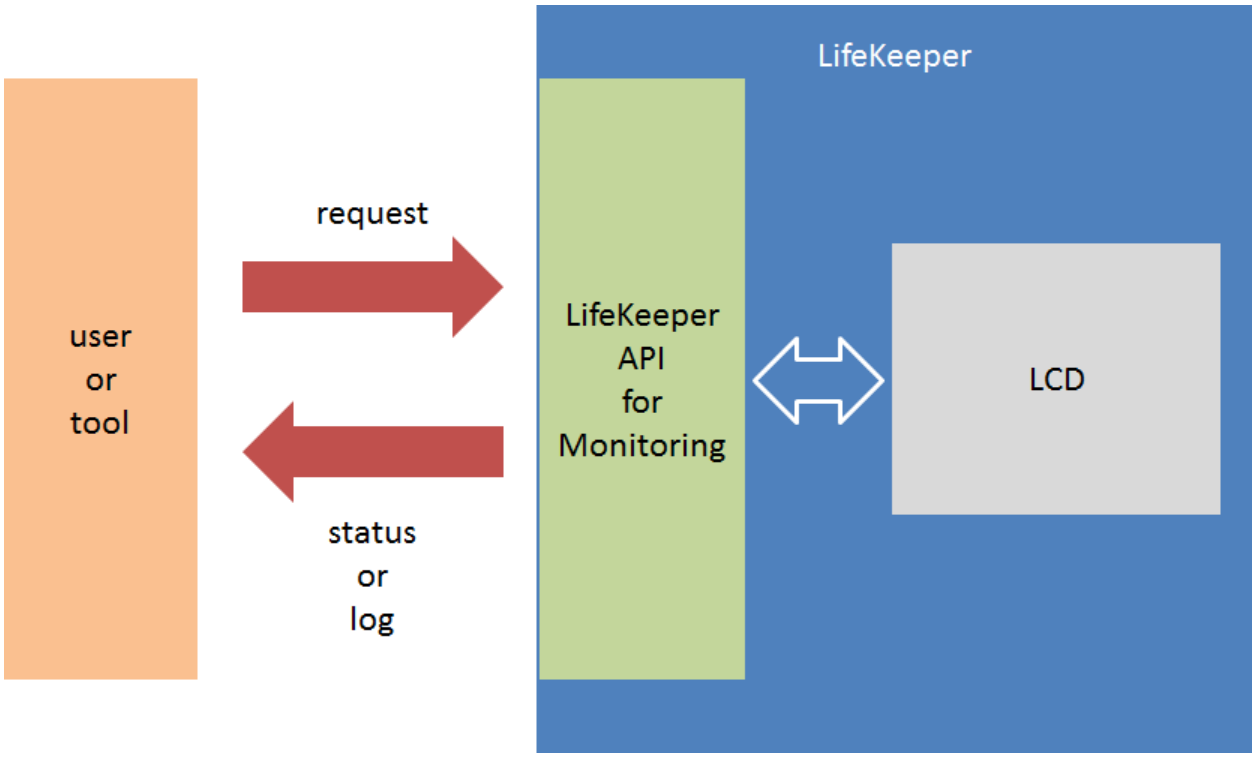


Chart 1. Overview of the LifeKeeper API for Monitoring

3.Information to be supplied with this API

The following information is supplied through this API when the user makes an inquiry to an active LifeKeeper node. The information supplied is about the specific LifeKeeper server to which the inquiry was directed even if the cluster consists of multiple servers.

- Status
- Operating status of each server
- Node name
- Operational status (ALIVE/DEAD)

- Operational status of communication path(s)
- Node name
- Operational status (ALIVE/DEAD)
- Address / device name
- Status of protected resources
- Node Name
- Tag
- Status (ISP, OSU, OSF, …)
- Dependency setting
- Mirror information for Data Replication resources (available only if status is ISP)
- Replication status (75%, 100%, ...)
- Mirror status (Sync/Async, Paused, ...)
- Log
- /var/log/lifekeeper.log
- Up to 1000 lines (when data output format is HTML)
- All (when data output format is plain text)
- Not supported if log file path is changed
- /var/log/lifekeeper.err
- Up to 1000 lines (when data output format is HTML)
- All (when data output format is plain text
- Not supported if log file path is changed

4.Communication format

The API uses HTTP to obtain the requested information. To obtain information, the user sends a HTTP GET request to the CGI scripts via lighttpd on the specific server.

5.Data format

The following 3 data formats are available.

- JSON
- To be used by an external tool to analyze the status information returned
- Status checking is possible
- Log output is not available

- HTML
- To be used to visually check via a browser
- Status checking is possible
- Log information is available up to 1000 lines
- plain text
- Used for regular log checking
- For logging purpose only and not for checking the status
- All contents of /var/log/lifekeeper.log and /var/log/lifekeeper.err are available

Available JSON format from the status in figure 2 is shown in figure 3, and HTML format is in figure 4

Figure 2. Example of active LifeKeeper configuration

```
{
  "resource": [
    {
      "replication": {},
      "child": [
        {
          "tag": "datarep-data"
        }
      ],
      "server": {
        "status": "ISP",
        "name": "lk01"
      },
      "tag": "/data"
    },
    {
      "replication": {
        "percent": "100%",
        "mirror": "Fully Operational"
      },
      "child": [],
      "server": {
        "status": "ISP",
        "name": "lk01"
      },
      "tag": "datarep-data"
    },
    {
      "replication": {},
      "child": [],
      "server": {
        "status": "ISP",
        "name": "lk01"
      },
```

Figure 3. Status output example using the JSON data format

RESOURCEs

tag	lk01
/data	ISP
datarep-data	ISP
ip-10.125.139.118	ISP

DATA REPLICATIONs

tag	nodes	mirror status	replication status
datarep-data	lk01 -> lk02	Fully Operational	100%

COMMUNICATION PATHs

communication path	status
192.168.139.18/192.168.139.19	ALIVE
172.20.139.18/172.20.139.19	ALIVE

Figure 4. Status output using the HTML format

6.How to use

6.1.Activate the API

The API is disabled by default. To activate, requires modification of /etc/default/LifeKeeper set the LKAPI_MONITORING configuration parameter to true (see figure 5). Setting of the configuration parameter only activates the API on that node and therefore must be set on each node on which the API will be used. Setting of this configuration parameter does not require a restart of LifeKeeper..

```
LKAPI_MONITORING=true
```

Figure 5. Enabling the LifeKeeper API for Monitoring

6.2Port number

The API uses port 779 by default. To change the port number, the user needs to set the following in /etc/default/LifeKeeper.

```
LKAPI_WEB_PORT=<port number>
```

Figure 6. Change the port number for LifeKeeper API for Monitoring

6.3.Usage examples

To obtain information a request is made to a server with an active LifeKeeper API configuration. Basic example using curl.

curl http://<IPADDR>:779/Monitoring.cgi

If no arguments are given, the current status is obtained using the JSON data format.

Request for log information using HTML data format.

curl http://<IPADDR>:779/Monitoring.cgi?format=html&show=log

The list of available arguments can be found in the table below.

List 1. Arguments

Name	Explanation	Value	Comments
show	Specify the target information	status, log, log-err	show=status is the default
format	Specify data format	json, html, plain	format=json is the default. If the format is json an error will be displayed if show=log or show=log-err is set.

7.Security

All the users requesting information via the API must be authorized to get LifeKeeper status information.

For this reason, user security settings can limit the users who can get the status by, configuring SSL, and encrypting the information.

7.1.Basic Authentication

To obtain the information via the API, Basic Authentication is required. To setup the authentication requires modification to the lighttpd configuration file (Modify the part in red colored character.) plus a restart of the lighttpd module. See figure 7 for how to configure lightttp.conf.

After modification execute the command ”/opt/LifeKeeper/sbin/sv restart steeleye-lighttpd” and reboot lighttpd to restart lighttpd using the new configuration.

/opt/LifeKeeper/etc/lighttpd/lighttpd.conf

```
server.modules          = (
   :
                        "mod_auth", # uncommenting
```

/opt/LifeKeeper/etc/lighttpd/conf.d/lkapi_user.conf

```
print qq/\$SERVER["socket"] == ":$lkapi_port" {\n/;
print qq/    server.document-root = "\/opt\/LifeKeeper\/api"\n/;
print qq/    auth.backend = "htpasswd"\n/;
print qq/    auth.backend.htpasswd.userfile =
"\/opt\/LifeKeeper\/etc\/lighttpd\/lighttpd.user.htpasswd"\n/;
print qq/    auth.require = ( "\/" =>\n/;
print qq/       (\n/;
print qq/               "method"  => "basic",\n/;
print qq/               "realm"   => "LifeKeeperAPI",\n/;
print qq/               "require" => "valid-user"\n/;
print qq/       )\n/;
print qq/    )\n/;
print qq/ }\n/;
```

Step to create htpasswd file.

```
htpasswd -c /opt/LifeKeeper/etc/lighttpd/lighttpd.user.htpasswd USERNAME
```

Figure 7. Basic Authentication setting example

7.2.SSL/TLS set up

SSL/TLS is available for the communication via this API. The lighttpd modifications for SSL/TLS is shown in the example in Figure 8. After modification execute the command ”/opt/LifeKeeper/sbin/sv restart steeleye-lighttpd” and reboot lighttpd to restart lighttpd with the new configuration.

/opt/LifeKeeper/etc/lighttpd/include_ssl_port.pl

```
configAPI("0.0.0.0", 443);
if(socket($sock, AF_INET6, SOCK_STREAM, 0)) {
   configAPI("[::]", 443);
}
sub configAPI {
   my $addr = shift;
   my $port = shift;

   print qq/\$SERVER["socket"] == "$addr:$port" {\n/;
   print qq/    server.document-root = "\/opt\/LifeKeeper\/api"\n/;
   print qq/    ssl.engine  = "enable"\n/;
   print qq/    ssl.pemfile = "\/opt\/LifeKeeper\/etc\/certs\/LK4LinuxValidNode.pem"\n/;
   print qq/    ssl.use-sslv2 = "disable"\n/;
   print qq/    ssl.use-sslv3 = "disable"\n/;
   print qq/ }\n/;
}
```

Figure 8 SSL/TLS setting example

7.3. Modification to support SSL/TLS + Basic authentication

Using SSL/TLS, modification example to set up Basic authentication is below. After modification, execute the command “/opt/LifeKeeper/sbin/sv restart steeleye-lighttpd“ and restart lighttpd to reflect the modified set up.

/opt/LifeKeeper/etc/lighttpd/lighttpd.conf

```
server.modules            = (
  :
                        "mod_auth", # uncommenting
```

/opt/LifeKeeper/etc/lighttpd/include_ssl_port.pl

```
configAPI("0.0.0.0", 443);
if(socket($sock, AF_INET6, SOCK_STREAM, 0)) {
  configAPI("[::]", 443);
}
sub configAPI {
  my $addr = shift;
  my $port = shift;

  print qq/\$SERVER["socket"] == "$addr:$port" {\n/;
  print qq/   server.document-root = "\/opt\/LifeKeeper\/api"\n/;
  print qq/   ssl.engine  = "enable"\n/;
  print qq/   ssl.pemfile = "\/opt\/LifeKeeper\/etc\/certs\/LK4LinuxValidNode.pem"\n/;
  print qq/   ssl.use-sslv2 = "disable"\n/;
  print qq/   ssl.use-sslv3 = "disable"\n/;
 print qq/   auth.backend = "htpasswd"\n/;
 print qq/   auth.backend.htpasswd.userfile =
"\/opt\/LifeKeeper\/etc\/lighttpd\/lighttpd.user.htpasswd"\n/;
 print qq/   auth.require = ( "\/" =>\n/;
 print qq/      (\n/;
 print qq/               "method"  => "basic",\n/;
 print qq/               "realm"   => "LifeKeeperAPI",\n/;
 print qq/               "require" => "valid-user"\n/;
 print qq/      )\n/;
 print qq/   )\n/;
  print qq/ }\n/;
}
```

Figure 9. SSL/TLS+Basic Authentication set up example

7.4.IP address access limitation

The lightppd configuration can also be setup to limit IP addresses that can be used to access data via the API. The lighttpd configuration to limit access is shown in Figure 9. The example will reject the connections from IP address other than 192.168.10.1. After modification execute the command ”/opt/LifeKeeper/sbin/sv restart steeleye-lighttpd” to restart lighttpd with the new configuration.

```
/opt/LifeKeeper/etc/lighttpd/conf.d/lkapi_user.conf

$HTTP["remoteip"] != "192.168.10.1" {
  url.access-deny = ( "" )
}
```

Figure 10. IP access limit setting example

8.Error

Errors can occur during the usage of this API when enabled. Should this occur, the summary of the error is output. Error example when JSON format is shown below.

HTTP status code returned by lighttpd is not described here.

```
{
"error" : {
    id : -1,
    message : "Failed to get LCD status"
  }
}
```

Figure 11. Error output example

Similar message is output in the case the output format is HTML.

User Guide

The LifeKeeper Single Server Protection User Guide is a complete, searchable resource containing detailed information on the many tasks that can be performed within the LifeKeeper GUI.

LifeKeeper GUI

LifeKeeper Graphical User Interface

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

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

GUI Overview - General

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

Note: It is highly recommended that the GUI client not be run over a WAN or VPN due to firewall constraints and performance issues. If you wish to run the GUI client remotely, it is recommended that a VNC session or Windows Remote Desktop session be used to log into a local system where the GUI client can be run more closely to the servers that are being administered.

GUI Server

The GUI server by default is not initialized on each LifeKeeper Single Server Protection server at system startup. The GUI server communicates with GUI clients using Hypertext Transfer Protocol (HTTP) and Remote Method Invocation (RMI). If you wish to start or stop the GUI server apart from starting and stopping LifeKeeper Single Server Protection, see Starting/Stopping the GUI Server.

GUI Client

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

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.

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 a connected server and its resources. It shows:

- the state of the server in the top row,
- the global 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 (check the **Properties Panel** checkbox in the View menu to enable 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 and selecting **Properties**.

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 **Reset** button queries the server for the current values of all properties, clearing any changes that you may have made. This button is always enabled.
- The **Help** button displays context-sensitive help for the properties panel. This button is always enabled.

To enable and disable this panel, use the **Properties Panel checkbox** on the View Menu.

Output Panel

The output panel collects output from commands issued by the LifeKeeper 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 LifeKeeper 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. Message 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.

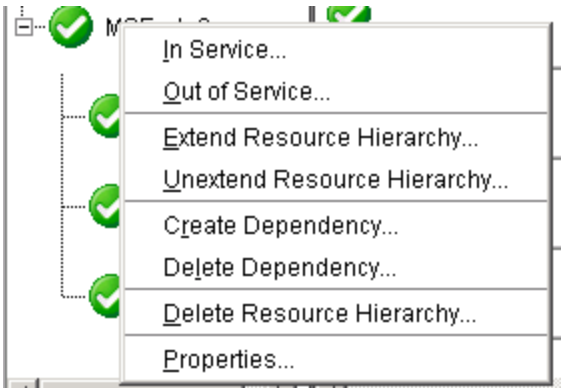
Exiting the GUI

Select **Exit** from the File Menu to disconnect from the server and close the GUI window.

Menus

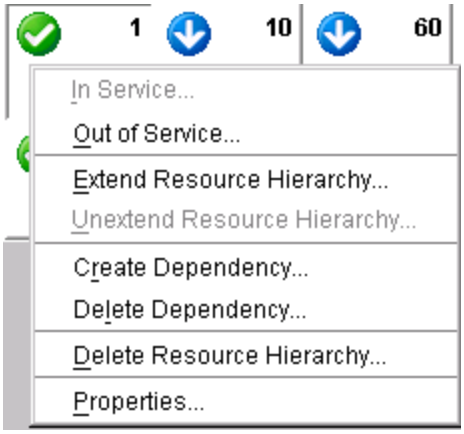
The GUI menus described in this section provide administration functions.

Resource Context Menu



The Resource Context Menu appears when you right-click on a global 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 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.

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

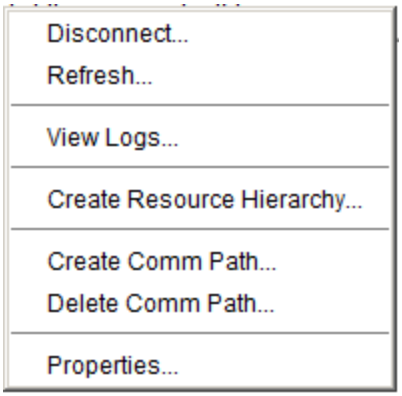
Delete Dependency. Remove a parent/child relationship between two resources.

Delete Resource Hierarchy. Remove a resource hierarchy.

Properties. Display the Resource Properties Dialog

Server Context Menu

The Server Context Menu appears when you right-click on a server icon in the status table. This menu is the same as the Edit Menu's Server submenu except that the actions are always invoked on the server that you initially selected.



Disconnect. Disconnect from a server.

Refresh. Refresh GUI.

View Logs. View LifeKeeper Single Server Protection log messages on connected servers.

Create Resource Hierarchy. Create a resource hierarchy.

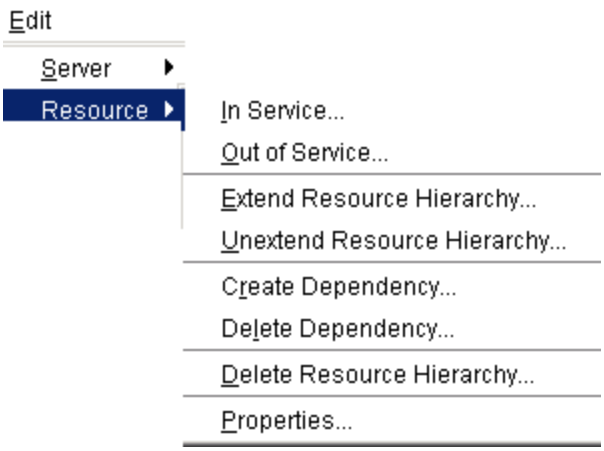
Properties. Display the Server Properties Dialog.

File Menu

Connect. Connect to a LifeKeeper Single Server Protection server. Connection to the LifeKeeper Single Server Protection server requires login authentication.

Exit. Disconnect from server and close the GUI window.

Edit Menu - Resource



In Service. Bring a resource hierarchy into service.

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

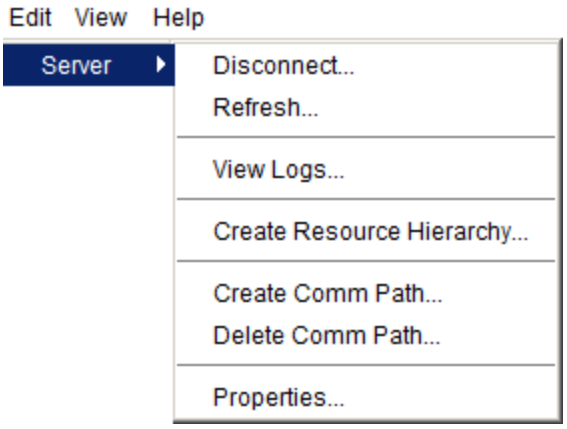
Create Dependency. Create a parent/child relationship between two resources.

Delete Dependency. Remove a parent/child relationship between two resources.

Delete Resource Hierarchy. Remove a resource hierarchy.

Properties. Display the Resource Properties Dialog.

Edit Menu - Server



Disconnect. Disconnect from a server.

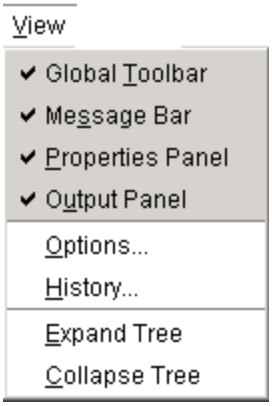
Refresh. Refresh GUI.

View Logs. View LifeKeeper Single Server Protection log messages on connected servers.

Create Resource Hierarchy. Create a resource hierarchy.

Properties. Display the Server Properties Dialog.

View Menu



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.

Options. Edit the display properties of the GUI.

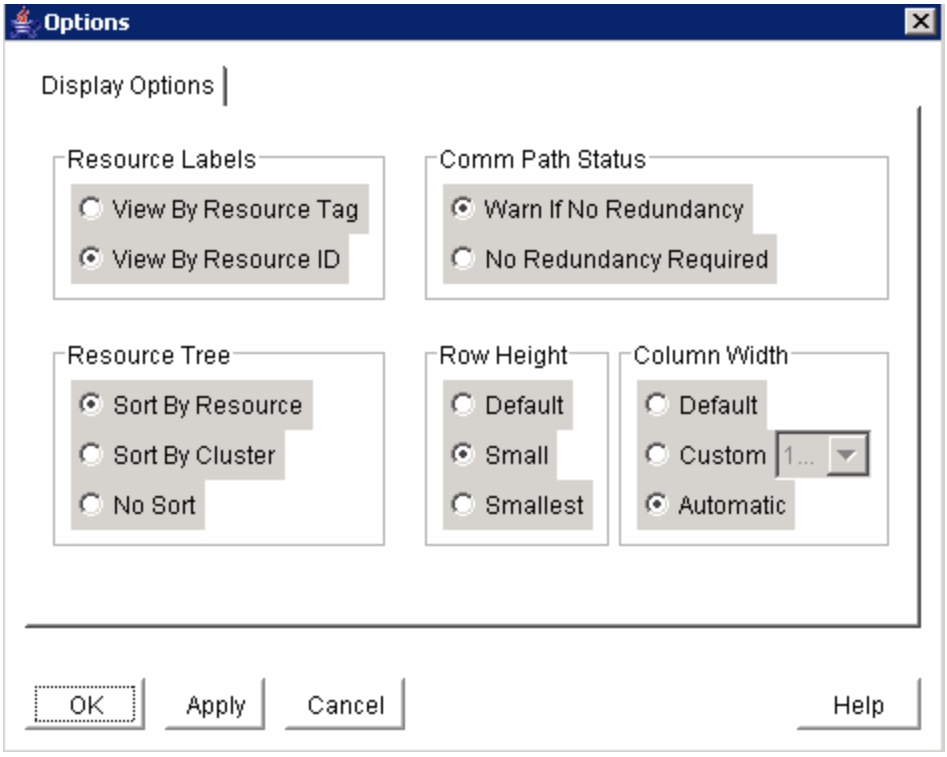
History. Display the newest messages that have appeared in the Message Bar in the LifeKeeper GUI Message History dialog box (up to 1000 lines).

Expand Tree. Expand the entire resource hierarchy tree.

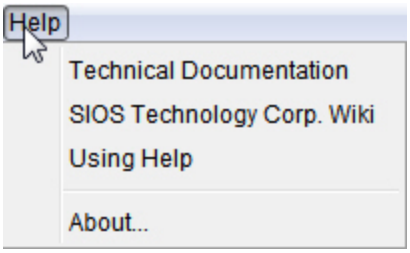
Collapse Tree. Collapse the entire resource hierarchy tree.

View Options Dialog

The **View Options dialog** is available from the **View** menu. The **Display Options** tab allows you to specify various GUI display characteristics.



Help Menu



Technical Documentation. Displays the Technical Documentation landing page with links to all documentation.

Using Help. Displays an overview of the Technical Documentation.

About.... Displays LifeKeeper GUI version information.

Toolbars

The GUI toolbars described in this section provide administration functions.

GUI Toolbar

This toolbar is a combination of the default server and resource context toolbars 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 LifeKeeper Single Server Protection Server.
	Disconnect. Disconnecting a LifeKeeper Single Server Protection server.
	Refresh. Refresh GUI.
	View Logs. View LifeKeeper Single Server Protection log messages on connected server.
	Create Resource Hierarchy. Create a resource hierarchy.
	Delete Resource Hierarchy. Remove a resource hierarchy from the LifeKeeper Single Server Protection server.
	In Service. Bring a resource hierarchy into service.
	Out of Service. Take a resource hierarchy out of service.
	Create Dependency. Create a parent/child relationship between two resources.
	Delete 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 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.
	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 LifeKeeper Single Server Protection server.
	Refresh. Refresh GUI.
	View Logs. View LifeKeeper Single Server Protection log messages on connected server.
	Create Resource Hierarchy. Create a resource hierarchy.
	Delete Resource Hierarchy. Remove a resource hierarchy from a LifeKeeper Single Server Protection server.

LifeKeeper GUI Message History

A history of the messages displayed in the message bar are displayed in the **LifeKeeper GUI Messages History** dialog. 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.

<-- 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:

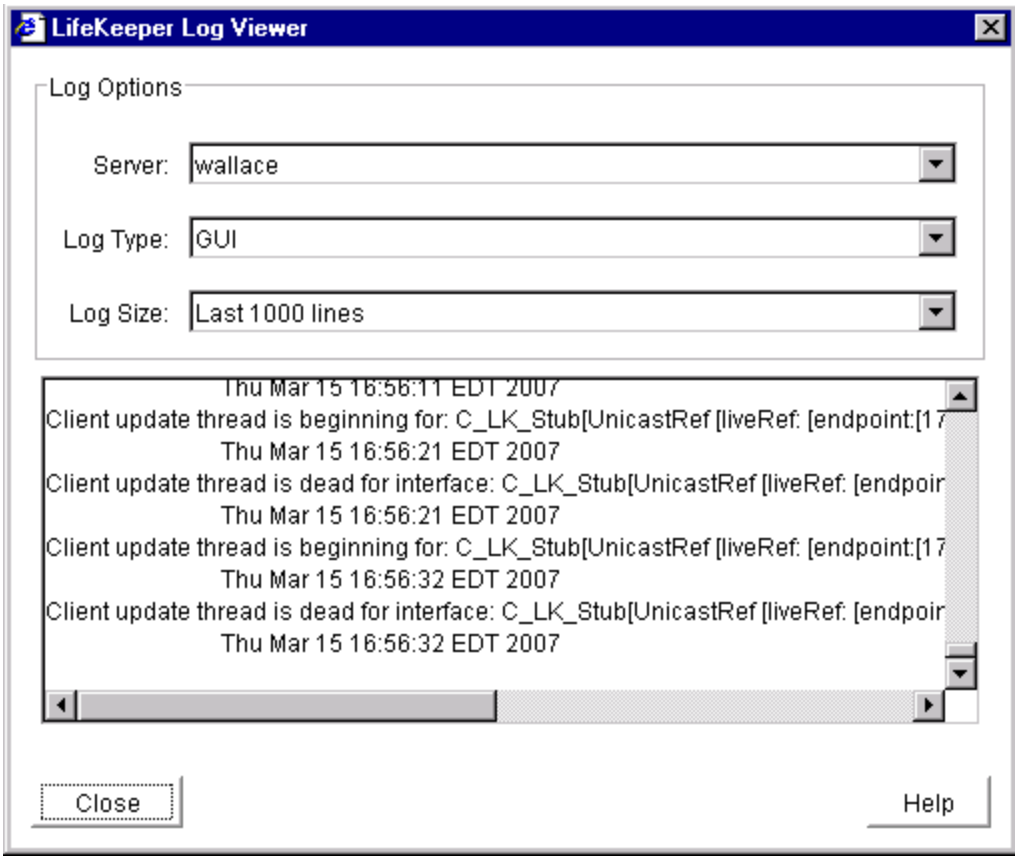
-->"server name":"action"

-->"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.



Preparing to Run the GUI

The topics in this section will help prepare the LifeKeeper Graphical User Interface.

Configuring the LifeKeeper Single Server Protection GUI

Configuring the LifeKeeper Single Server Protection Server for GUI Administration

Perform the following steps for each LifeKeeper Single Server Protection server. Each step contains references or links for more detailed instructions.

1. You must install the Java Runtime Environment (JRE) or Java Software Development Kit (JDK) on each server. See the LifeKeeper Single Server Protection Release Notes for the required Java version and URL to access the required download. Note: You may install the JRE from the LifeKeeper Single Server Protection IMG file by running the setup script on the IMG file, and opting only to install the JRE. (See Using the LifeKeeper Single Server Protection ISO Image for more information.)
2. Start the LifeKeeper GUI Server on each server (see Starting/Stopping the GUI Server). **Note**:

Once the GUI Server has been started following an initial installation, starting and stopping LifeKeeper Single Server Protection will start and stop all LifeKeeper Single Server Protection daemon processes including the GUI Server.

3. If you plan to allow users other than root to use the GUI, then you need to Configure GUI Users.

Running the GUI

You can run the LifeKeeper GUI on the LifeKeeper Single Server Protection server.

See Running the GUI on the LifeKeeper Single Server Protection Server for information on configuring and running the GUI.

GUI Configuration

Item	Description
GUI client and server communication	The LifeKeeper GUI client and server use Java Remote Method Invocation (RMI) to communicate. For RMI to work correctly, the client and server must use resolvable hostnames or IP addresses. If DNS is not implemented (or names are not resolvable using other name lookup mechanisms), edit the */etc/hosts* file on each client and server to include the names and addresses of all other LifeKeeper Single Server Protection servers.
GUI Server Java platform	The LifeKeeper GUI server requires that the Java Runtime Environment (JRE) - Java virtual machine, the Java platform core classes and supporting files - be installed. The JRE 5.0 for Linux is available on the LifeKeeper Single Server Protection IMG file or it can be downloaded directly from http://www.oracle.com/technetwork/java/archive-139210.html. **Note:** By default, the LifeKeeper GUI server expects the JRE on each server to be installed in the directory */usr/java/j2re1.5.0_07*. If this is not found, it will look in the directory */usr/java/j2sdk1.5.0_07* for a Java Software Development Kit (JDK). If you want to use a JRE or JDK in another directory location, you must edit the PATH in the LifeKeeper Single Server Protection default file */etc/default/LifeKeeper* to include the directory containing the java interpreter, java.exe. If LifeKeeper Single Server Protection is running when you edit this file, you should stop and restart the LifeKeeper GUI server to recognize the change. Otherwise, the LifeKeeper GUI will not be able to find the Java command.
Java remote object registry server port	The LifeKeeper GUI server uses port 82 for the Java remote object registry on each LifeKeeper Single Server Protection server. This should allow servers to support RMI calls from clients behind typical firewalls.

LifeKeeper Single Server Protection administration web server	The LifeKeeper GUI server requires an administration web server for client browser communication. Currently, the LifeKeeper GUI server is using a private copy of the lighttpd web server for its administration web server. This web server is installed and configured by the stee-leye-lighttpd package and uses port 81 to avoid a conflict with other web servers.

GUI Limitations

Item	Description
GUI inter-operability restriction	The LifeKeeper Single Server Protection for Linux client may only be used to administer LifeKeeper Single Server Protection on Linux servers. The LifeKeeper for Linux GUI will *not* interoperate with LifeKeeper for Windows.

Configuring GUI Users

There are three classes of GUI users with different permissions for each.

1. Users with ***Administrator*** permission can perform all possible actions through the GUI.
2. Users with ***Operator*** permission on a server can view LifeKeeper Single Server Protection 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 LifeKeeper Single Server Protection configuration and status information on that server.

The GUI server must be invoked as *root*. During installation of the GUI package, an entry for the root login and password is automatically configured in the GUI password file with ***Administrator***permission, allowing *root* to perform all LifeKeeper Single Server Protection tasks on that server via the GUI application or web client. If you plan to allow users other than *root* to use LifeKeeper GUI clients, then you need to configure LifeKeeper GUI users.

User administration is performed through the command line interface, using *lkpasswd*, as described below. Unless otherwise specified, all commands require you to enter the user's password twice. They take effect on the user's next login or when the GUI server is restarted, whichever comes first. Each user has a single permission on a given server. Previous permission entries are deleted if a new permission is specified on that server.

- To grant a user ***Administrator*** permission for the LifeKeeper GUI, type the following command:

  ```
  /opt/LifeKeeper/bin/lkpasswd -administrator <user>
  ```

- To grant a user ***Operator*** permission for the LifeKeeper GUI, type the following command:

  ```
  /opt/LifeKeeper/bin/lkpasswd -operator <user>
  ```

- To grant a user ***Guest*** permission for the LifeKeeper GUI, type the following command:

```
/opt/LifeKeeper/bin/lkpasswd -guest <user>
```

- To change the password for an existing user without changing their permission level, type the following command:

  ```
  /opt/LifeKeeper/bin/lkpasswd <user>
  ```

- To prevent an existing user from using the LifeKeeper GUI, type the following command:

  ```
  /opt/LifeKeeper/bin/lkpasswd -delete <user>
  ```

- This command does not require password entry.

Java Security Policy

The LifeKeeper GUI uses policy-based access control. When the GUI client is loaded, it is assigned permissions based on the security policy currently in effect. The policy, which specifies permissions that are available for code from various signers/locations, is initialized from an externally configurable policy file.

There is, by default, a single system-wide policy file and an optional user policy file. The system policy file, which is meant to grant system-wide code permissions, is loaded first, and then the user policy file is added to it. In addition to these policy files, the LifeKeeper GUI policy file may also be loaded if the LifeKeeper GUI is invoked as an application.

Location of Policy Files

The system policy file is by default at:

<JAVA.HOME>/lib/security/java.policy (Linux)

<JAVA.HOME>\lib\security\java.policy (Windows)

Note: JAVA.HOME refers to the value of the system property named "JAVA.HOME", which specifies the directory into which the JRE or JDK was installed.

The user policy file starts with `.` and is by default at:

<USER.HOME>\.java.policy

Note: USER.HOME refers to the value of the system property named "user.home", which specifies the user's home directory. For example, the home directory on a Windows NT workstation for a user named Paul might be "paul.000".

For Windows systems, the user.home property value defaults to

*C:\WINNT\Profiles\<USER> (**on multi-user Windows NT systems**)*

*C:\WINDOWS\Profiles\<USER> (**on multi-user Windows 95/98 systems**)*

*C:\WINDOWS (**on single-user Windows 95/98 systems**)*

The LifeKeeper GUI policy file is by default at:

*/opt/LifeKeeper/htdoc/java.policy (**Linux**)*

Policy File Creation and Management

By default, the LifeKeeper GUI policy file is used when the LifeKeeper GUI is invoked as an application. If you are running the LifeKeeper GUI as an applet, you will need to create a user policy file in your home directory if one does not already exist. The user policy file should specify the minimum permissions required to run the LifeKeeper GUI, which are provided in the "Sample Policy File" section later in this topic.

A policy file can be created and maintained via a simple text editor, or via the graphical **Policy Tool** utility included with the Java Runtime Environment (JRE) or Java Development Kit (JDK). Using the Policy Tool saves typing and eliminates the need for you to know the required syntax of policy files. For information about using the Policy Tool, see the Policy Tool documentation at http://docs.oracle.com/javase/1.3/docs/tooldocs/tools.html.

The **simplest way to create a user policy file** with the minimum permissions required to run the LifeKeeper GUI is to copy the LifeKeeper GUI policy file located in ***/opt/LifeKeeper/htdoc/java.policy*** to your home directory and rename it *.java.policy* (note the leading dot before the filename which is required). On a Windows system, you can copy the LifeKeeper GUI policy file by opening the file *http://<server name>:81/java.policy* (where <server name> is the host name of a LifeKeeper Single Server Protection server) and saving it as *.java.policy* in your home directory. If you need to determine the correct location for a user policy file, enable the Java Console using the Java Control Panel and start the LifeKeeper GUI as an applet. The home directory path for the user policy file will be displayed in the Java Console.

Granting Permissions in Policy Files

A permission represents access to a system resource. In order for a resource access to be allowed for an applet, the corresponding permission must be explicitly granted to the code attempting the access. A permission typically has a name (referred to as a "target name") and, in some cases, a comma-separated list of one or more actions. For example, the following code creates a FilePermission object representing read access to the file named *abc* in the */tmp* directory:

```
perm = new java.io.FilePermission("/tmp/abc","read");
```

In this, the target name is "*/tmp/abc*" and the action string is "*read*".

A policy file specifies what permissions are allowed for code from specified code sources. An example policy file entry granting code from the */home/sysadmin* directory read access to the file */tmp/abc* is:

```
grant codeBase "file:/home/sysadmin/" {
permissionjava.io.FilePermission "/tmp/abc", "read"; };
```

Sample Policy File

The following sample policy file includes the minimum permissions required to run the LifeKeeper GUI. This policy file is installed in ***/opt/LifeKeeper/htdoc/java.policy*** by the LifeKeeper GUI package.

```
     /*
      * Permissions needed by the LifeKeeper GUI. You may want to
      * restrict this by codebase. However, if you do this, remember
      * that the recovery kits can have an arbitrary jar component
      * with an arbitrary codebase, so you'll need to alter the grant
      * to cover these as well.
      */
      grant {

/*
 * Need to be able to do this to all machines in the
 * LifeKeeper cluster. You may restrict the network
 * specification accordingly.
 */
 permission java.net.SocketPermission"*", "accept,connect,resolve";
 /*
 * We use URLClassLoaders to get remote properties files and
 * jar pieces.
 */
 permission java.lang.RuntimePermission"createClassLoader";
 /*
 * The following are needed only for the GUI to run as an
 * application (the default RMI security manager is more
 * restrictive than the one a browser installs for its
 * applets.
 */
 permission java.util.PropertyPermission "*","read";
 permission java.awt.AWTPermission "*";
 permission java.io.FilePermission "<<ALL FILES>>","read,execute";

};
```

Java Plug-In

Whether you are using Netscape 6/7, Mozilla 1.x, Firefox 1.x or Internet Explorer 6/7, the first time your browser attempts to load the LifeKeeper GUI, it will either automatically download the Java Plug-In software or redirect you to a web page to download and install it. From that point forward, the browser will automatically invoke the Java Plug-in software every time it comes across web pages that support the technology.

Downloading the Java Plug-in

Java Plug-in software is included as part of the Java Runtime Environment (JRE) for Solaris, Linux and Windows. Downloading the JRE typically takes a total of three to ten minutes, depending on your network and system configuration size. The download web page provides more documentation and installation instructions for the JRE and Java Plug-in software.

Note 1: You should close and restart your browser after installing the plug-in and whenever plug-in properties are changed.

Note 2: Only Java Plug-in version 1.3.x or later are supported with LifeKeeper Single Server Protection.

Java Plug-in Troubleshooting

If you are using Netscape 6/7, Mozilla 1.x or Firefox 1.x, you may need to create a symbolic link in the Netscape, Mozilla or Firefox plugins directory to the path of the *libjavaplugin_oji.so* file under the *$JAVAHOME* directory.

For example (Firefox 1.5 and jre 1.5):

```
cd /usr/lib/mozilla//plugins
ln -s/usr/java/jre1.5.0_07/plugin/i386/ns7/libjavaplugin_oji.so
```

For Netscape 4, if your browser doesn't find the Java Plug-in even though you've installed the Java Runtime Environment, including the Java Plug-in, be sure that the *NPX_PLUGIN_PATH* environment variable is set to the location of the Java Plug-in (the directory in which the *javaplugin.so* file is located) e.g. export *NPX_PLUGIN_PATH=$JAVAHOME/jre/plugin/i386/ns4* where *$JAVAHOME* is the top-level directory of your Java Runtime Environment installation.

The Java Plug-in supports the Java 2 SDK, Standard Edition v1.3 security model. All applets are run under the standard applet security manager. See the Java Security FAQ or Setting Browser Security Parameters for the GUI Applet for more information.

In some platform/browser combinations, the Java Plug-in software will affect the appearance and behavior of Java components in the LifeKeeper GUI such as the ScrollBar, ToolBar or Menu components. In many of these situations, if you resize or iconify/de-iconify the window (i.e. force a repaint) as a workaround, the problem will go away.

Running the GUI on a Remote System

You may administer LifeKeeper Single Server Protection from a Linux, Unix or Windows system outside the LifeKeeper Single Server Protection server by running the GUI as a Java applet. Configuring and running the LifeKeeper GUI remotely is described below.

Configuring the GUI on a Remote System

In order to run the LifeKeeper GUI on a remote Unix system or Windows system, your browser must provide full JDK 1.4 support. Refer to the LifeKeeper Single Server Protection Release Notes for information on the supported platforms and browsers for the LifeKeeper GUI.

1. If you are running the LifeKeeper GUI as an applet, you need to create a Java user policy file in your home directory if one does not already exist. The user policy file should specify the minimum permissions required to run the LifeKeeper GUI.
    - The **simplest way to create a user policy file** with the minimum permissions required to run the LifeKeeper GUI is to copy the LifeKeeper GUI policy file located in ***/opt/LifeKeeper/htdoc/java.policy*** to your home directory and rename it ***.java.policy*** (note there is a leading dot in the file name that is required). On a Windows system, you can copy the LifeKeeper GUI policy file

by opening the file *http://<server name>:81/java.policy* (where <*servername*> is the host name of a LifeKeeper Single Server Protection server) and saving it as *.java.policy* in your home directory. If you need to determine the correct location for a user policy file, enable the Java Console using the Java Control Panel and start the LifeKeeper GUI as an applet. The home directory path for the user policy file will be displayed in the Java Console.

- If you already have a user policy file, you can add the required entries specified in *$LKROOT/ htdoc/java.policy* on a LifeKeeper Single Server Protection server into the existing file using a simple text editor. See the Java Security Policy topic for further information.

2. You must set your browser security parameters to low. This generally includes enabling of the Java and Java applets. Since there are several different browsers and versions, the instructions for setting browser security parameters are covered in the topic Setting Browser Security Parameters for the GUI Applet. **Note**: It is important to use caution in visiting external sites with low security settings.
3. When you run the GUI for the first time, if you are using Netscape or Internet Explorer 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. See the LifeKeeper Single Server Protection Release Notes for the required Java Plug-in version and URL to access the download.

Running the GUI on a Remote System

After you have completed the tasks described above, you are ready to run the LifeKeeper GUI as an applet on a remote system.

1. Open the URL, *http://<server name>:81*, for the LifeKeeper GUI webpage (where <*server name*> is the name of the LifeKeeper Single Server Protection server). The webpage contains the LifeKeeper Single Server Protection splash screen and applet. When the web page is opened, the following actions take place:
   - 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.

   If everything loads properly, a **Start** button should appear in the applet area. If the splash screen does not display a **Start** button or you suspect that the applet failed to load and initialize, refer to the Applet Troubleshooting section below or see Network-Related Troubleshooting.

2. When prompted, click **Start**. The LifeKeeper GUI appears and the Server Connect dialog is automatically displayed. Once a Server has been entered and connection established, the GUI window displays a visual representation and status of the resources protected by the connected server. The GUI menus and toolbar buttons provide LifeKeeper Single Server Protection administration functions.

Note: Some browsers add "**Warning: Applet Window**" to windows and dialogs created by an applet. This is normal and should be ignored.

Applet Troubleshooting

If you suspect that the applet failed to load and initialize, try the following:

1. Verify that 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. Verify that you have installed the Java Plug-in. If your problem appears to be Java Plug-in related, refer to the Java Plug-in topic.
3. Verify that you have met the browser configuration requirements, especially the security settings. Refer to Setting Browser Security Parameters for the GUI Applet for more information. If you don't find anything obviously wrong with your configuration, continue with the next steps.
4. 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**.
5. Reopen the URL, *http://<server name>:81,* to start the GUI applet. If you've modified the Java Plug-In Control Panel, restart your browser.
6. 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.

Common Tasks

This section contains basic tasks that can be performed by any user.

Connecting to a Server

This task connects your GUI client to your LifeKeeper Single Server Protection server.

1. There are two possible ways to begin.
    - On the global toolbar, click the **Connect** button.
    - On the File Menu, click **Connect**.
2. In the **Server Name** field, enter the name of a server 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 add the server to the status display.

Note: If the initial login name and password fails to authenticate the client on the server, the user is prompted to enter another login name and password for that server. If "**Cancel**" is selected from the Password dialog, connection to the server is aborted.

Disconnecting From a Server

This task disconnects your GUI client from your LifeKeeper Single Server Protection server.

1. There are three possible ways to begin.
    - On the Global Toolbar, click the **Disconnect** button.
    - On the Edit Menu, select **Server** and then click **Disconnect**.
    - On the Server Context Toolbar, if displayed, click the **Disconnect** button.
2. Select the name of the server from which you want to disconnect.
3. Click **OK**. A **Confirmation** dialog listing the server is displayed.
4. Click **OK** in the **Confirmation** dialog to confirm that you want to disconnect from the server.

After disconnecting from the server, it is removed from the GUI status display.

Viewing Connected Servers

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. 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.
UNKNOWN		Network connection was lost. Last known LifeKeeper Single Server Protection state is ALIVE.

Viewing Server Log Files

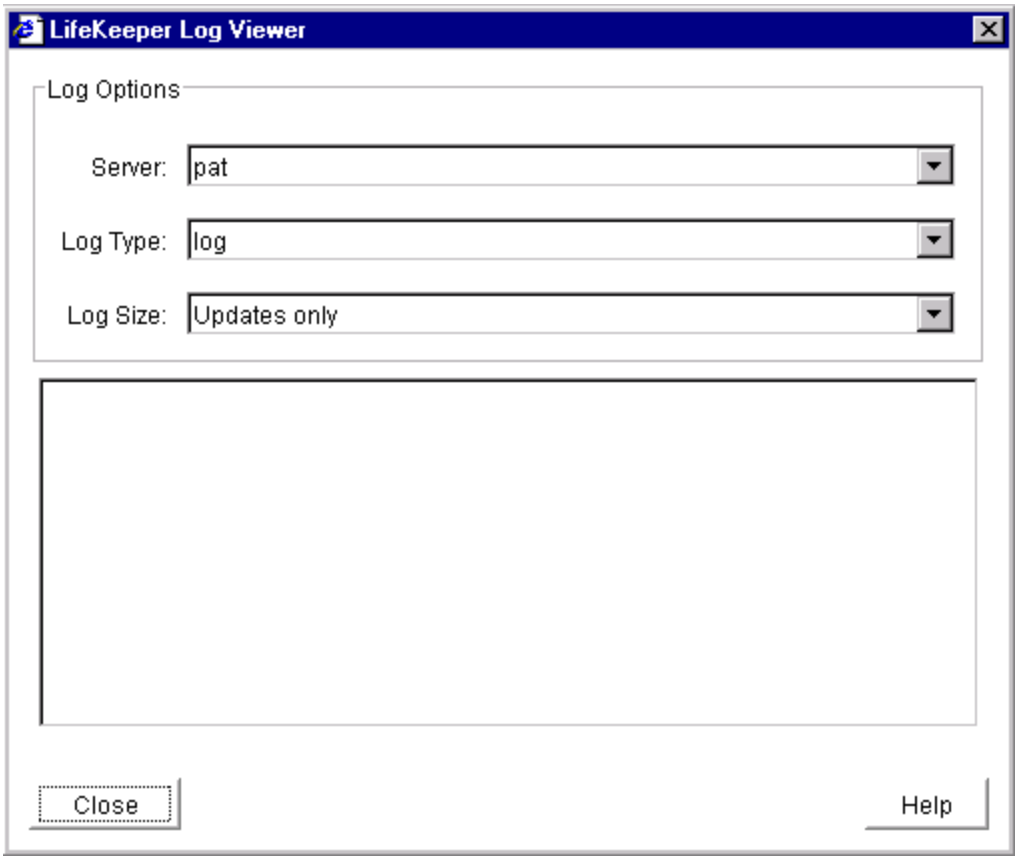
1. There are four ways to begin.
    - Right-click on a server icon to display the Server Context Menu, then click **View Log** to bring up the LifeKeeper Single Server Protection Log Viewer Dialog.

- On the Global Toolbar, click the **View Log** button, then select the server that you want to view from the Server list in the LifeKeeper Single Server Protection Log Viewer Dialog.
- On the Server Context Toolbar, if displayed, click the **View Log** button.
- On the Edit Menu, point to **Server**, click **View Log**, then select the server that you want to view from the Server list in the **LifeKeeper Single Server Protection Log Viewer Dialog**.

2. If you started from the **Global Toolbar** or the **Edit Menu** and you want to view logs for a different server, select that server from the **Server** list in the LifeKeeper Single Server Protection Log Viewer Dialog. This feature is not available if you selected **View Logs** from the **Server Context Menu** or **Server Context Toolbar**.
3. When you are finished, click **OK** to close the **Log Viewer** dialog.

Log Viewer Dialog

The **Log Viewer Dialog** may be accessed from the server context menu, the global toolbar or the server context toolbar if the properties panel is enabled and server properties are displayed. This dialog displays a limited view of the log files maintained by LifeKeeper Single Server Protection. When accessed from the toolbar or the **View** menu, you can look at log files for any server by changing the selected server in the Server list.



Server. A drop-down list of servers connected to the cluster. Select the server whose log files you want to view. This list is not available if the dialog is invoked from the server context menu.

Log Type. A drop-down list of log files available on the selected server. Select one of the options to indicate which log files you want to see:

- log
- TTYLCM
- LCM
- LCD
- remote_execute
- GUI
- SNMP
- NOTIFY

Log Size. A drop-down list containing four options. Select one of the options to indicate how much of the log file you want to see:

- Updates Only
- Last 100 lines
- Last 500 lines
- Last 1000 lines

Viewing Server Properties

1. There are two possible ways to begin.
    - Right-click on the icon for the server for which you want to view the properties. When the Server Context Menu appears, click **Properties**. Server properties will also be displayed in the Properties Panel if it is enabled when clicking on the server.
    - On the Edit Menu, point to **Server** and then click **Properties**. When the dialog comes up, select the server for which you want to view the properties from the Server list.
2. If you want to view properties for a different server, select that server from the dialog's **Server** list.
3. When you are finished, click **OK** to close the window.

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 = ipdnet0-153.98.87.73, Resource ID = IP-153.98.87.73
```

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: **Hierarchy Resource Status** (left pane) shows the resource dependencies in a tree fashion, and the **Server Resource Status Table** shows a table of individual resource statuses for each server.

Server Resource Status Table

The following figure shows servers with resource statuses.

Server Resource State	Visual State	What it Means
Active		Resource is operational and protected. (ISP)
Warning		Resource is operational but there is a warning. (ISU)
StandBy		Resource is out of service. (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 Single Server Protection is not running on this server.
	Empty panel	Server does not have the resource defined.

Viewing Resource Properties

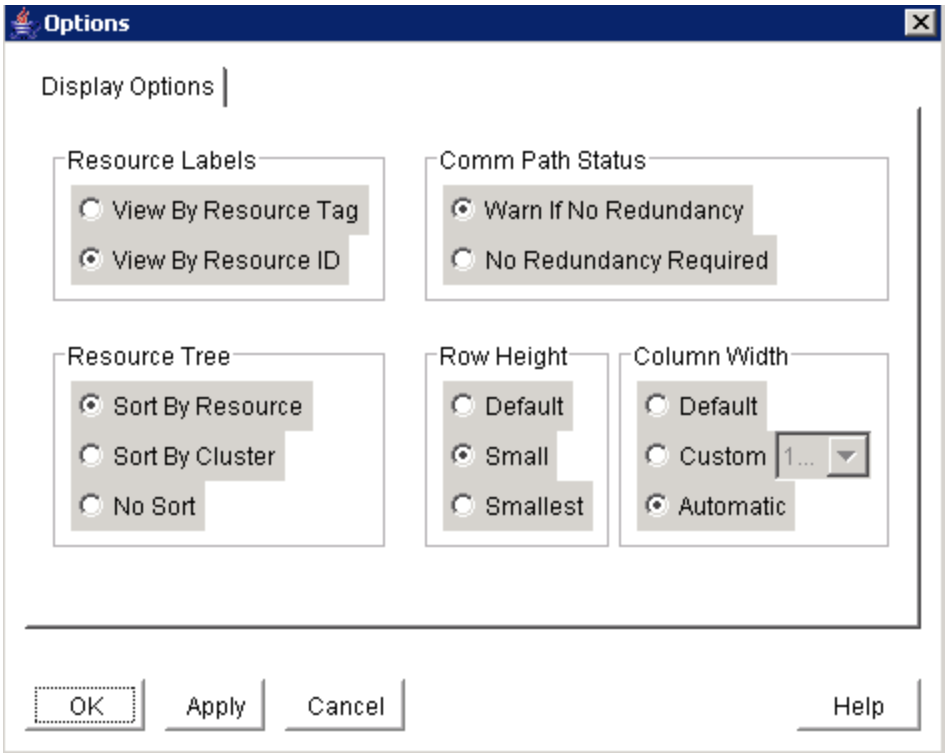
There are two ways to view resource properties:

- Enable the Properties Panel and click the resource for which you wish to view properties.
- Right-click on the icon for the resource for which you want to view the properties. When the Resource Context Menu appears, click **Properties**. Resource properties will also be displayed in the Properties Panel if it is enabled.

Setting View Options for the Status Window

The **Options** Dialog is available from the **View** menu. This allows you to specify various LifeKeeper Single Server Protection display characteristics. These settings, along with all checkbox menu item settings and the various window sizes, are stored between sessions in the file *.lkGUIpreferences* in your home folder on the client machine. This file is used by both the web and application clients. The preference settings on each client machine are independent of those on other machines. If you want to synchronize preference settings between two machines, you may do so permanently by sharing the preference files or temporarily by moving copies between the machines.

1. On the View Menu, click **Options**. The View Options Dialog is displayed.



2. To arrange the display of resources in the status window, click the **Display Options** tab and then select the option group you would like to modify. See the detailed explanation of the option groups below.
3. Click **OK** to save your settings and return to the status window.

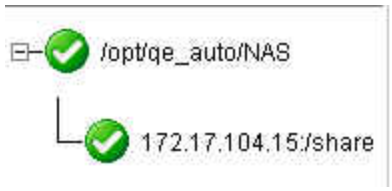
Resource Labels

This option group allows you to specify whether resources are viewed in the resource hierarchy tree by their tag name or ID.

Note: The resource tag/ID shown in the resource hierarchy tree 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.

By tag name:

By ID:



Resource Tree

This option group allows you to specify the sorting order of the resources in the resource hierarchy tree.

- **Sort By Resource** will sort resources by resource label only.
- **No Sort** will disable sorting such that the resources are displayed in the order in which they are discovered by the GUI.

Top level resources in the resource hierarchy tree may be sorted manually by left-clicking the resource in the tree and "dragging" it to a new position. The order depends on what resource is moved and the location in the tree to which it has been moved.

Note: The 0 (zero) and 9 (nine) keys are defined as hot/accelerator keys to facilitate quickly expanding or collapsing the resource hierarchy tree. The mouse can be used to expand or collapse the complete tree by clicking on the title area of the resource hierarchy tree; double-click to expand and single-click to collapse.

Row Height

This option group allows you to control the row height of the resources in the table. The choices are **Default**, **Small** and **Smallest**.

Note: The "+" and "-" keys are defined as hot/accelerator keys to facilitate quickly resizing resources in the resource hierarchy tree and table.

Column Width

This option group allows you to control the column width of the servers and resources in the table. The choices are:

- **Default**: Standard width.
- **Custom**: Allows you to select a width (in pixels) from a drop-down list.
- **Automatic**: Automatically resizes all columns to fill available space.

Note: The 7 (seven) and 8 (eight) keys are defined as hot/accelerator keys to facilitate quickly resizing the column size of resources in the resource hierarchy table.

Viewing Message History

1. On the View Menu, click **History**. The LifeKeeper GUI Message History dialog is displayed.
2. If you want to clear all messages from the history, click **Clear**.
3. Click **OK** to close the dialog.

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

	In this segment of the tree, the resource *file_system_2* is expanded and the resource *nfs-/opt/qe_auto/NFS/export1* is collapsed. appears to the left of a resource icon if it is expanded. appears if it is collapsed.

To **expand** a resource hierarchy tree,

- Click the or
- Double-click the resource icon to the right of a .

To **expand all** resource hierarchy trees,

- On the **View Menu**, click **Expand Tree** or
- Double-click the **Resource Hierarchy Tree** button in the column header in the left pane of the **Status** window.

Note: The resource tag/ID shown in the resource hierarchy tree 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 tree,

- click the or
- double-click the resource icon to the right of a .

To collapse all resource hierarchy trees,

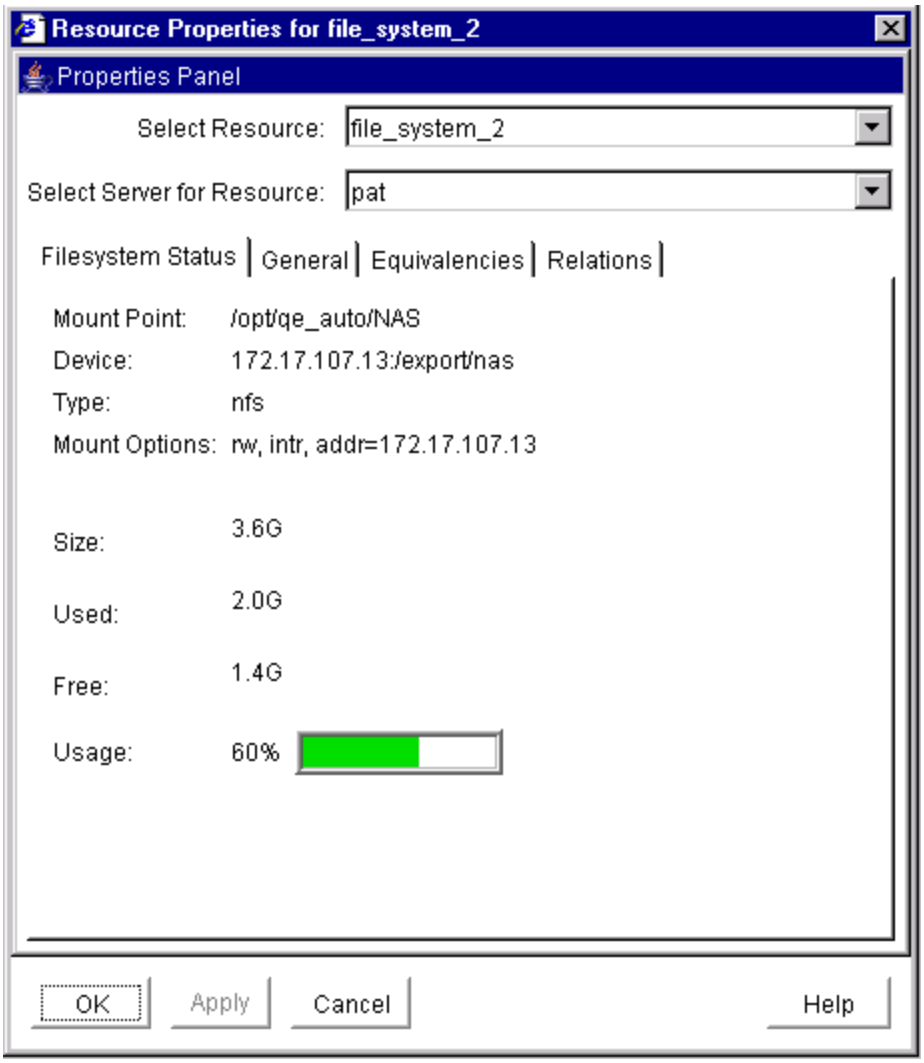
- On the **View Menu**, click **Collapse Tree** or
- Double-click the **Resource Hierarchy Tree** button in the column header in the left pane of the Status window

Note: The "9" and "0" keys are defined as hot/accelerator keys to facilitate quickly expanding or collapsing all resource hierarchy trees.

Resource Properties Dialog

The Resource Properties dialog is available from the Edit menu or from a resource context menu. This dialog displays the properties for a particular resource on a server. When accessed from the Edit menu, you can select the resource and the server. When accessed from a resource context menu, you can select the server.

General Tab

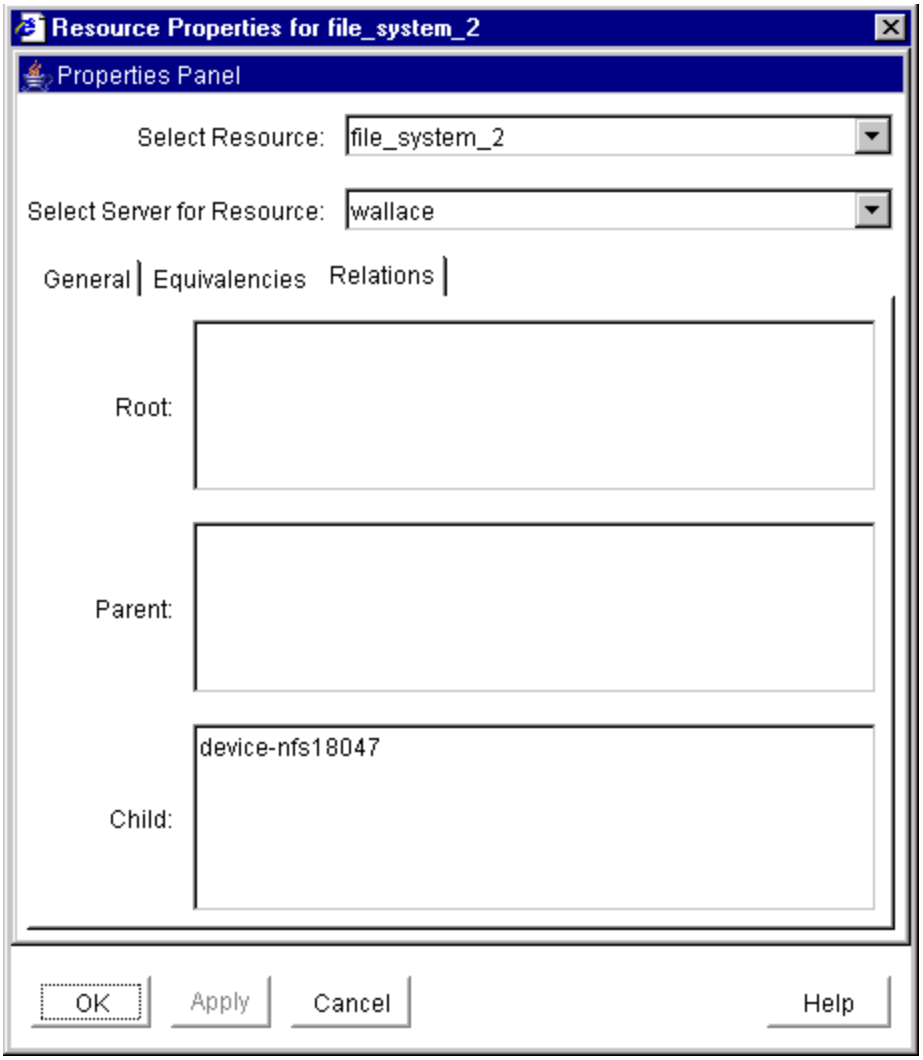


- **Tag**. The name of a resource instance, unique to a system, that identifies the resource to an administrator.
- **ID**. A character string associated with a resource instance, unique among all instances of the

resource type, that identifies some internal characteristics of the resource instance to the application software associated with it.

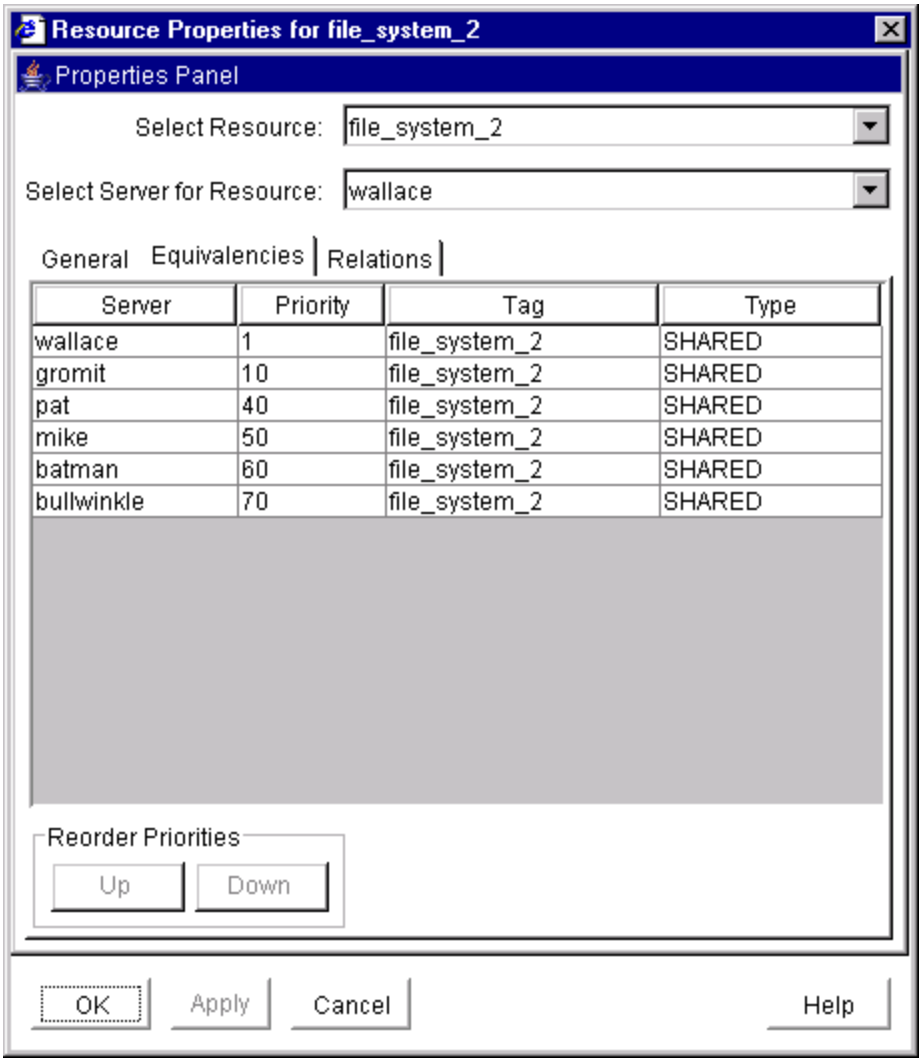
- **State**. Current state of the resource instance:
  - ***Active*** - In-service locally and protected.
  - ***Warning*** - In-service locally, but local recovery will not be attempted.
  - ***Failed*** - Out-of-service, failed.
  - ***Standby***- Out-of-service, unimpaired.
  - ***ILLSTATE***- A resource state has not been initialized properly by the resource initialization process which is run as part of the LifeKeeper Single Server Protection startup sequence. Resources in this state are not under LifeKeeper Single Server Protection protection.
  - ***UNKNOWN***- Resource state could not be determined. The GUI server may not be available.
- **Reason**. If present, describes the reason the resource is in its current state, that is, the reason for the last state change. For example the application on galahad is in the OSU state because the shared primary resource *ordbfsaa-on-tristan* on tristan is in ISP or ISU state. Shared resources can be active on only one of the grouped systems at a time.
- **Initialization**. The setting that determines resource initialization behavior at boot time, for example, `AUTORES_ISP`, `INIT_ISP`, or `INIT_OSU`.

Relations Tab



- **Parent**. Identifies the tag names of the resources that are directly dependent on this resource.
- **Child**. Identifies the tag names of all resources on which this resource depends.
- **Root**. Tag name of the resource in this resource hierarchy that has no parent.

Equivalencies Tab



- **Server**. The name of the server on which the resource has a defined equivalency.
- **Tag**. The tag name of this resource on the equivalent server.
- **Type**. The type of equivalency (`SHARED`, `COMMON`, `COMPOSITE`).
- **Reorder Priorities** (available if the user has Administrator permission). Up/Down buttons let you to re-order the priority of the selected equivalency.

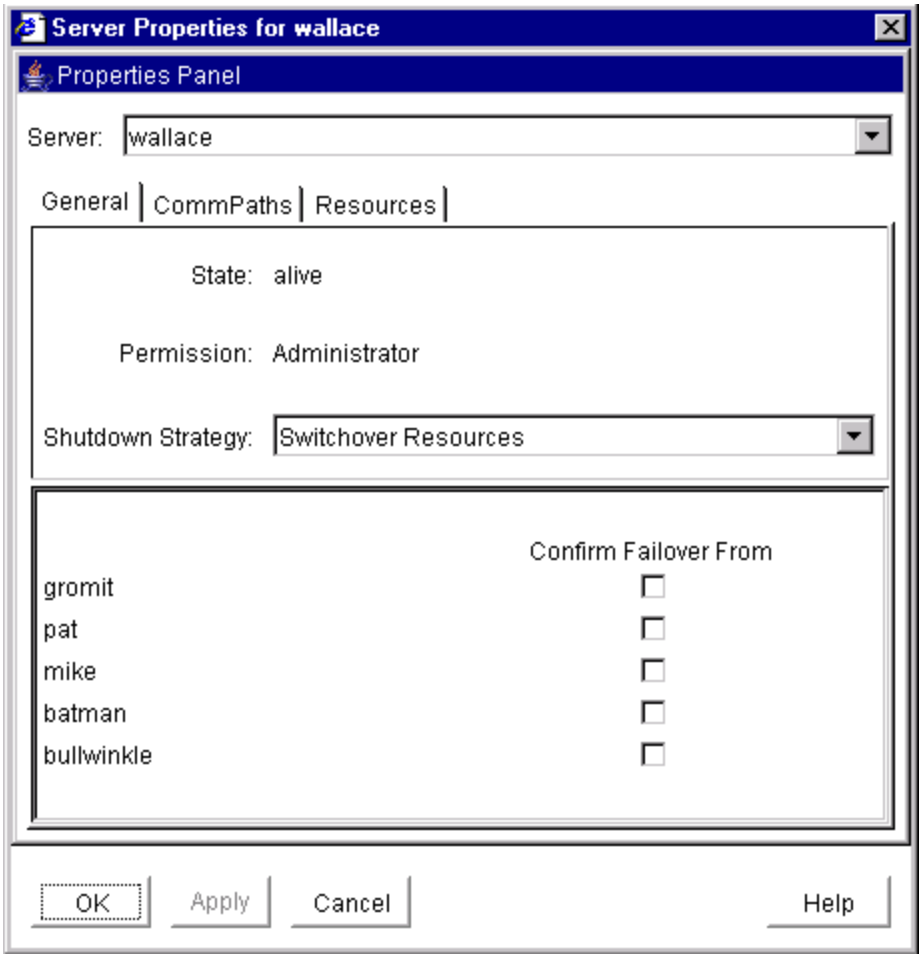
The **OK** button applies any changes that have been made and then closes the window. The **Apply** button applies any changes that have been made. The **Cancel** button closes the window without saving any changes made since Apply was last clicked.

Server Properties Dialog

The Server Properties dialog is available from a server context menu or from the Edit menu. This dialog displays the properties for a particular server. The properties for the server will also be displayed in the properties panel if it is enabled.

The three tabs of this dialog are described below. 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

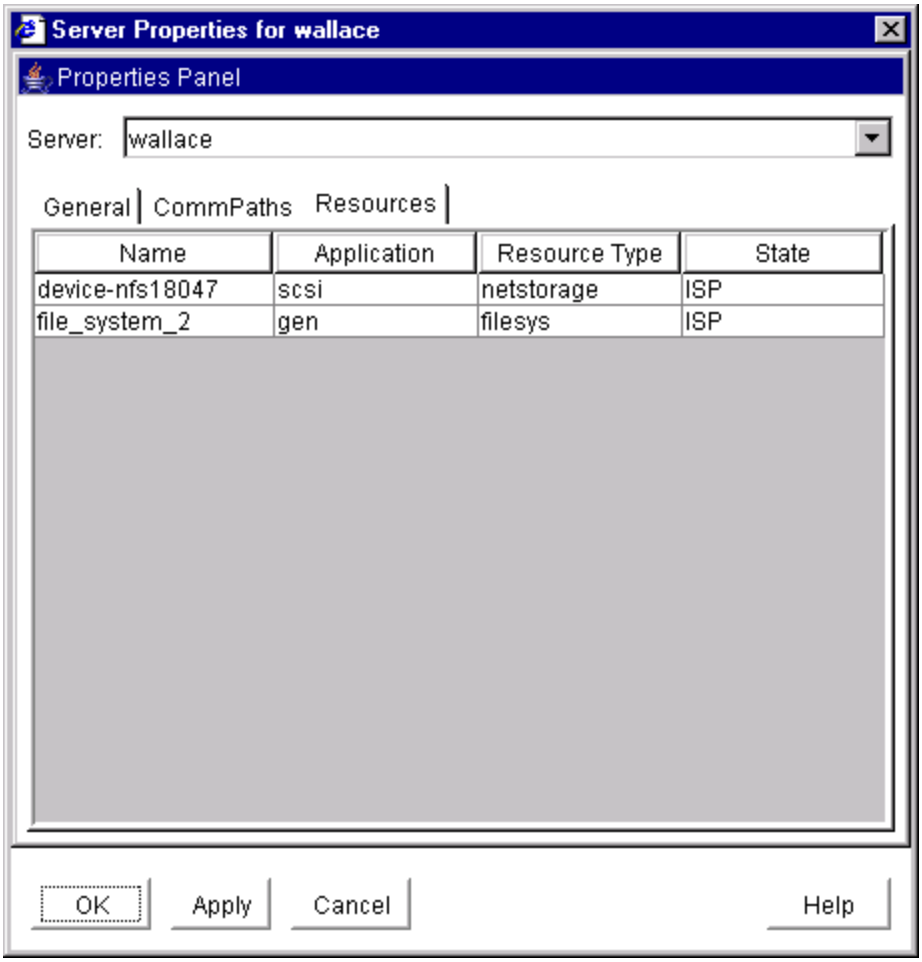


- **Name**. Name of the selected server.
- **State**. Current state of the server. These are the possible server state values:

- *ALIVE* - server is available.
- *DEAD* - server is unavailable.
- *UNKNOWN* - state could not be determined. The GUI server may not be available.

- **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 Single Server Protection task.
  - *Operator* - the user can monitor LifeKeeper Single Server Protection resource and server status, and can bring resources in service and take them out of service.
  - *Guest* - the user can monitor LifeKeeper Single Server Protection resource and server status.

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, app, filesys, nfs, device, disk,...)
- **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 Single Server Protection startup sequence. Resources in this state are not under LifeKeeper Single Server Protection protection.
    - *UNKNOWN* - Resource state could not be determined. The GUI server may not be available.

Operator Tasks

The following topics are more advanced tasks that require Operator permission.

- Bringing a Resource In Service
- Taking a Resource Out of Service

Bringing a Resource In Service

1. There are five possible ways to begin.
    - Right-click on the icon for the resource/server combination that you want to bring into service. When the Resource Context Menu appears, click **In Service**.
    - Right-click on the icon for the global resource that you want to bring into service. When the **Resource Context Menu** appears, click **In Service**. When the dialog comes up, select the server on which to perform the In Service from the **Server** list and click **Next**.
    - On the Global Toolbar, click the **In Service** button. When the dialog comes up, select the server on which to perform the In Service from the **Server** list and click **Next**. On the next dialog, select one or more resources that you want to bring into service from the **Resource**(s) list and click **Next** again.
    - On the Resource Context Toolbar, if displayed, click the **In Service** button.
    - On the Edit Menu, point to **Resource** and then click **In Service**. When the dialog comes up, select the server on which to perform the **In Service** from the **Server** list,

and click **Next**. On the next dialog, select one or more resources that you want to bring into service from the Resource(s) list and click **Next** again.

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. Any additional dependent (child) resources that were brought into service are noted in the dialog or **output panel**.
4. Errors that occur while bringing a resource in service are logged in both the LifeKeeper Single Server Protection log and the GUI log of the server on which you want to bring the resource into service.

Taking a Resource Out of Service

1. There are four possible ways to begin.
   - Right-click on the icon for the global resource or resource/server combination that you want to take out of service. When the Resource Context Menu appears, click **Out of Service**.
   - On the Global Toolbar, click the **Out of Service** button. When the Out of Service dialog comes up, select one or more resources that you want to take out of service from the Resource(s) list, and click **Next**.
   - On the Resource Context Toolbar, if displayed, click the **Out of Service** button.
   - On the Edit Menu, point to **Resource** and then click **Out of Service**. When the **Out of Service** dialog comes up, select one or more resources that you want to take out of service from the **Resource**(s) list, and click **Next**.
2. An **Out of Service** 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 proceed to the next dialog box.
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 Single Server Protection log and the GUI log of the server on which you want to take the resource out of service.

Advanced Tasks

LCD

LifeKeeper Configuration Database

The LifeKeeper Configuration Database (LCD) maintains the object-oriented resource hierarchy information and stores recovery direction information for all resource types known to LifeKeeper Single Server Protection. 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 **Related** and **Other** topics for information on the LCD directory structure, types of data stored, resource types available and use of application scripts.

LCD Directory Structure

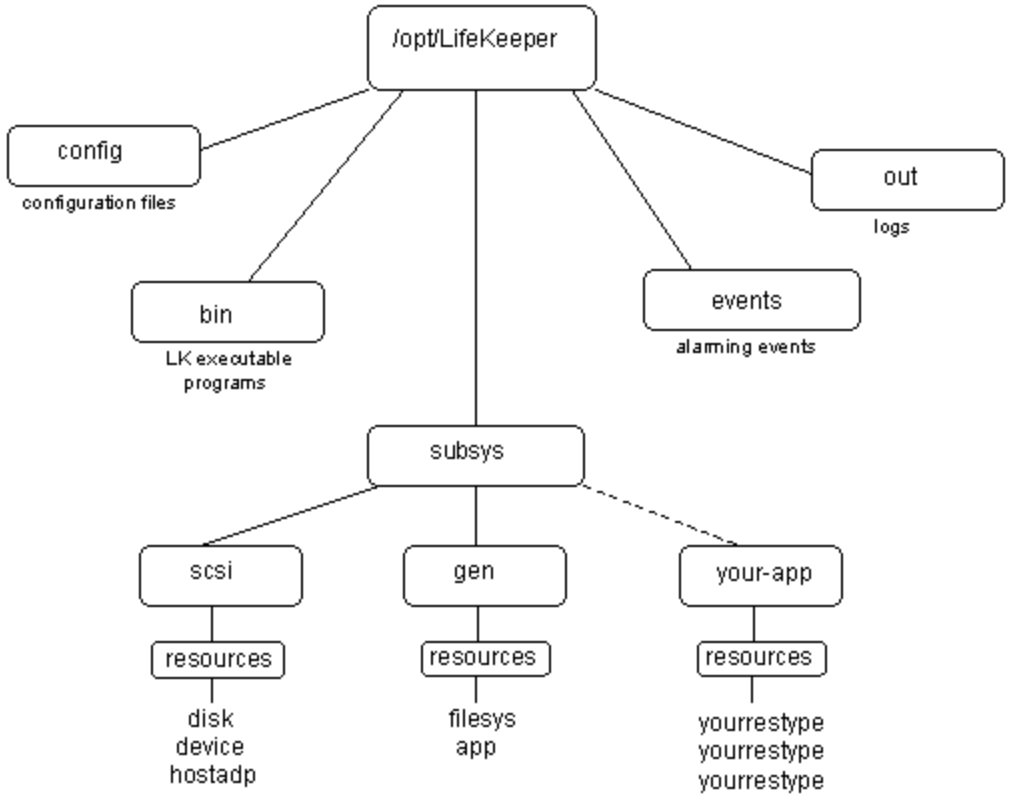
Major subdirectories under */opt/LifeKeeper*:

- **config**. LifeKeeper Single Server Protection configuration files, including shared equivalencies.
- **bin**. LifeKeeper Single Server Protection executable programs, such as is_recoverable. See Fault Detection and Recovery Scenarios for descriptions.
- **subsys**. Resources and types. LifeKeeper Single Server Protection provides resource and type definitions for the generic application menu functions in gen. When you define an application interface, you create directories under subsys.
- **events**. Alarming events. See LifeKeeper Single Server Protection Alarming and Recovery for further information.
- **out**. LifeKeeper Single Server Protection logs. LifeKeeper Single Server Protection sends a variety of error and status messages to several different logs in this directory. See the lk_log (8) manual page.

The structure of the LCD directory in */opt/LifeKeeper* is shown in the topic Structure of LCD Directory in */opt/LifeKeeper*.

Structure of LCD Directory in /opt/LifeKeeper

The following diagram shows the directory structure of ***/opt/LifeKeeper***.



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 Single Server Protection maintains a list of dependencies and a list of dependents (resources depending on a resource.) For information, see the LCDI_relationship (1M) and LCDI_instances (1M) manual pages.

Resource Status Information

LCD maintains status information in memory for each resource instance. The resource states recognized by LCD are **ISP**, **ISU**, **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.

LCD Resource Types

The LCD is maintained in both shared memory and in the */opt/LifeKeeper* directory. As highlighted on the directory structure diagram, subsys contains an application resource set you can use to define your application interface:

- gen - generic application and file system information

This subdirectory is discussed in Resources Subdirectories.

Resources Subdirectories

The ***gen*** directory contains a resources subdirectory. The content of this directory provides a list of the resource types provided by LifeKeeper Single Server Protection:

gen resource types. You find these resource types in the */opt/LifeKeeper/subsys/gen/resources* directory:

- **filesys**—file systems
- **app**—generic or user-defined applications that may depend upon additional resources

Each resource type directory contains one or more of the following:

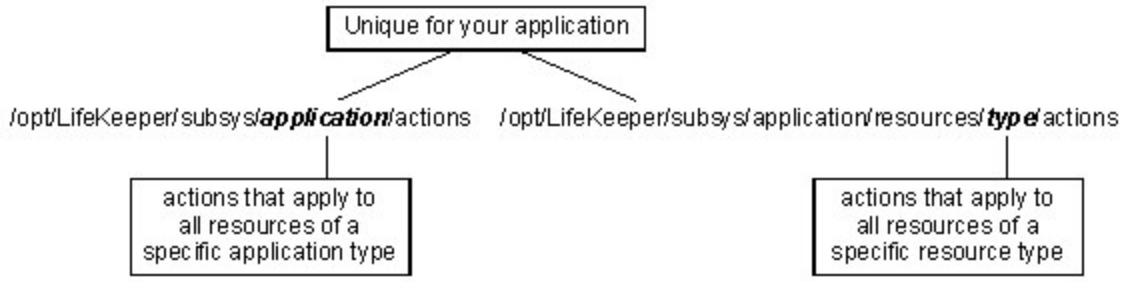
- **instances**. This file reflects the permanent information saved in the LCD about resource instances. It contains descriptive information for the resource instances associated with this resource type.

WARNING: Do not modify the instances file (or any LCD file) directly. To create or manipulate resource instances, use only the LifeKeeper GUI functions or the LifeKeeper Single Server Protection LCDI_instances commands: ins_create, ins_remove, ins_gettag, ins_setas, ins_setinfo, ins_setinit, ins_setstate and ins_list. Refer to the LCDI_instances manual pages for explanations of these commands.

- ***recovery***. This optional directory contains the programs used to attempt the local recovery of a resource for which a failure has been detected. The recovery directory contains directories that correspond to event classes passed to sendevent. The names of the directories must match the class parameter (-C) passed to the sendevent program. (See LifeKeeper Single Server Protection Alarming and Recovery.)

In each subdirectory, the application can place recovery programs that service event types of the corresponding event class. The name of these programs must match the string passed to sendevent with the -E parameter. This optional directory may not exist for many applications.

- ***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 Single Server Protection Recovery Action and Control Interface (LRACI) **perform_action** shell program described in the LRACI-perform_action (1M) manual page.

For additional discussion, see Recovery Scripts.

LifeKeeper Single Server Protection Flags

Near the end of the detailed status display, LifeKeeper Single Server Protection 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.
```

An example of a general LCD lock flag:

- **`!action!02833!701236710!<servername>:filesys`**. The creation of a filesystem hierarchy produces a flag in this format in the status display. The *filesys* designation can be a different resource type for other application resource hierarchies or *app* for generic or user-defined applications.

LCDI Commands

LifeKeeper Single Server Protection provides two mechanisms for defining an application resource hierarchy:

- LifeKeeper GUI
- LifeKeeper Configuration Database Interface (LCDI) commands

The LCDI is a set of commands provided by LifeKeeper Single Server Protection that you can use to create custom resource hierarchy configurations to meet your application needs. You can use the command line interface to configure.

For a description of the commands, see the LCDI manual pages. This topic provides a configuration scenario that demonstrates the way you can use both the GUI and command line to create a resource hierarchy.

Hierarchy Definition

These are the tasks required to construct an application hierarchy:

1. **Create file system resources**. The LifeKeeper GUI provides menus to create file system resources. See Creating File System Resource Hierarchies.

   At the end of this definition task, the LCD has two filesys resources defined as follows. Run lcdstatus -q to view the resource status:

TAG	ID	STATE
my-fs-1	/mnt/fs1	ISP
my-fs-2	/mnt/fs2	ISP

   Note: LifeKeeper Single Server Protection does not place any significance on the tag names used; they are simply labels. The tag names shown are the LifeKeeper Single Server Protection defaults.

2. **Define an application resource.** Create a generic application (gen/app) resource named *my-app*. See Creating a Generic Application Resource Hierarchy.

3. **Define dependencies**. To make the application dependent upon the filesystem resources, use the dep_create command to create resource dependencies, as follows:

   ```
   dep_create -p my-app -c my-fs-1

   dep_create -p my-app -c my-fs-2
   ```

4. **Bring resources into service**. Access the LifeKeeper GUI, right-click the application resource, then select **In-Service** to bring the resources into service.

LCM

The LifeKeeper Communications Manager (LCM) provides reliable communication between processes on a LifeKeeper Single Server Protection server. This process can use redundant communication paths so that failure of a single communication path does not cause failure of LifeKeeper Single Server Protection or its protected resources. The LCM supports a variety of communication alternatives including RS-232 (TTY) and TCP/IP connections.

The LCM provides the following:

- **LifeKeeper Single Server Protection Heartbeat.** Periodic communication with VMware HA to determine if the system is still functioning. LifeKeeper Single Server Protection withholds the heartbeat when it detects an unrecoverable failure to notify VMware HA to reboot the

server.

- **Administration Services.** The administration functions of LifeKeeper Single Server Protection 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.

In addition to the LifeKeeper Single Server Protection services provided by the LCM, inter-system application communication is possible through a set of shell commands for reliable communication. These commands include snd_msg, rcv_msg, and can_talk. These commands are described in the LCMI_mailboxes (1M) manual pages. The LCM runs as a real-time process on the system assuring that critical communications such as system heartbeat will be transmitted.

LifeKeeper Single Server Protection Alarming and Recovery

LifeKeeper Single Server Protection error detection and notification is based on the event alarming mechanism, sendevent. The key concept of the **sendevent** mechanism is that independent applications can register to receive alarms for critical components. Neither the alarm initiation component nor the receiving application(s) need to be modified to know the existence of the other applications. Application-specific errors can trigger LifeKeeper Single Server Protection recovery mechanisms via the **sendevent** facility.

This section discusses topics related to alarming including alarm classes, alarm processing and alarm directory layout and then provides a processing scenario that demonstrates the alarming concepts.

Alarm Classes

The */opt/LifeKeeper/events* directory lists a set of alarm classes. These classes correspond to particular sub-components of the system that produces events (for example, *filesys*). For each alarm class, subdirectories contain the set of potential alarms (for example, *badmount* and *diskfull*). You can register an application to receive these alarms by placing shell scripts or programs in the appropriate directories.

LifeKeeper Single Server Protection uses a basic alarming notification facility. With this alarming functionality, all applications registered for an event have their handling programs executed asynchronously by sendevent when the appropriate alarm occurs. With LifeKeeper Single Server Protection present, the **sendevent** process first determines if the LifeKeeper Single Server Protection resource objects can handle the class and event. If LifeKeeper Single Server Protection finds a class/event match, it executes the appropriate recover scenario.

Defining additional scripts for the **sendevent** alarming functionality is optional. When you define LifeKeeper Single Server Protection resources, LifeKeeper Single Server Protection provides the basic alarming functionality described in the processing scenarios later in this chapter.

Note: Local recovery for a resource instance is the attempt by an application under control of LifeKeeper Single Server Protection to return interrupted resource services to the end-user on the same system that generated the event.

Alarm Processing

Applications or processes that detect an event which may require LifeKeeper Single Server Protection attention can report the event by executing the **sendevent** program, passing the following arguments: respective error class, error name and failing instance. Refer to the sendevent manual pages for required specifics and optional parameters and syntax.

Alarm Directory Layout

The */opt/LifeKeeper/events* directory has two types of content:

- **LifeKeeper Single Server Protection supplied classes**. LifeKeeper Single Server Protection provides two alarm classes listed under the *events* directory: *lifekeeper* and *filesys*. Examples of the alarm events include *noaccess* and *diskfull*. The alarm classes correspond to the strings that are passed with the -C option to the **sendevent** command and the alarm events correspond to the strings that are passed with the -E option. The LifeKeeper Single Server Protection alarm class is used internally by LifeKeeper Single Server Protection for event reporting within the LifeKeeper Single Server Protection subsystem.
- **Application-specific classes**. The other subdirectories in the *events* directory are added when specific applications require alarm class definitions. Applications register to receive these alarms by placing shell scripts or binary programs in the directories. These programs are named after the application package to which they belong.

Recovery Scripts

You can define an application interface to LifeKeeper Single Server Protection through the LifeKeeper GUI, through the command interface or a combination of these as demonstrated by the example in the topic LCDI Command Interface. Whatever method you use, you must at least provide the recovery scripts to start and stop the resource.

To help you define the scripts you need for your application interface, see Types of Scripts and Script Parameters. These topics discuss in more detail four types of scripts: remove, restore, delete and local recovery. The discussions of the remove and restore types include script examples.

Application Interface Levels

LifeKeeper Single Server Protection provides a foundation for event detection and recovery control as well as an overall environment and tool set for ensuring application availability. This topic, along with Interface Issues for Common Application Types and Interface Definition Tasks, describe:

- the application interface levels to consider for assuring availability,
- the actions required to provide interfaces for common application types and
- the steps for creating an interface between your application and LifeKeeper Single Server Protection.

Interface Levels

Your involvement in ensuring high availability for your application begins with the proper choice and configuration of the application. For instance, if you need high availability for applications involving database transactions, you must select or develop an application that provides features such as logging, archiving, roll forward/rollback facilities and controllable internal recovery techniques.

Next, you should decide how to implement the three levels of application interfaces supported by LifeKeeper Single Server Protection :

- dependency definition
- error detection and handling
- recovery actions

Dependency Definition

In order to ensure availability, you must determine the application's dependency upon other system resources and determine how to discover and handle errors in those resources. LifeKeeper Single Server Protection lets you define dependency relationships (hierarchies) using either the LifeKeeper GUI or the command line interface.

Error Detection and Handling

Providing detection and alarming for problems within an application is critical to building the best total fault resilient solution. Since every application varies on the mechanism and format of failures, no one set of generic mechanisms can be supplied. In general, however, many application configurations can rely on the core system error detection provided within LifeKeeper Single Server Protection. The Fault Detection and Recovery Scenario topic describes three common fault situations to demonstrate the power of LifeKeeper Single Server Protection's core facilities.

LifeKeeper Single Server Protection also provides a complete environment for defining errors, alarms and events that can trigger recovery procedures. This interfacing usually requires pattern match definitions for the system error log (*/var/log/messages*) or custom-built application-specific monitor processes.

Recovery Actions

Fault resilience depends upon the definition of recovery actions for the various anticipated fault conditions. LifeKeeper Single Server Protection uses shell scripts to specify recovery actions for protected resources. For LifeKeeper Single Server Protection, the two required recovery scripts for

any application resource are restore and remove, used whenever LifeKeeper Single Server Protection needs to place an application in service or out-of-service, respectively.

The Recovery Scripts topic lists the various types of scripts that LifeKeeper Single Server Protection recognizes and provides descriptions and examples that can help you create customized scripts.

Interface Issues For Common Application Types

The actions you perform to provide an interface between your application and LifeKeeper Single Server Protection depend upon the type and complexity of your application. For example:

File System Service. The LifeKeeper GUI provides menu-based configuration of file system hierarchies.

DBMS Application. The LifeKeeper Single Server Protection product family provides optional recovery kit packages for the RDBMS applications such as MySQL, Oracle, DB2 and Informix. These recovery packages deliver menu options in the LifeKeeper GUI for configuring a database instance into LifeKeeper Single Server Protection protection. These packages also provide default restore and remove scripts that use the associated database initialization commands.

Simple Application. For configurations that depend upon a single file system or disk device, the LifeKeeper GUI provides menus for hierarchy creation functions. You need only provide remove and restore scripts for the application itself.

Complex Application. For more complex configurations where no associated LifeKeeper Single Server Protection application recovery kit package exists, you can use the LifeKeeper Configuration Database Interface (LCDI) commands. You would use the commands, for example, to provide an interface for a custom relational database and for applications that depend upon more than one file system or disk partition. You would also need to provide the necessary restore and remove scripts. For further discussion, see LCDI Command Interface.

Interface Definition Tasks

Before you begin constructing an interface between your application and LifeKeeper Single Server Protection, you need to be sure that you have the appropriate hardware and software (including your application and LifeKeeper Single Server Protection) installed and working properly.

After you are assured that your components are configured and working properly, you should perform the following steps to create the interface with LifeKeeper Single Server Protection:

1. **Create application scripts (or programs)**. LifeKeeper Single Server Protection uses a set of scripts (or programs) to start or stop an application. The two scripts generally required for any application under LifeKeeper Single Server Protection are restore and remove. You can also provide other action scripts to determine how the application or the system reacts to specific operational conditions.
2. **Define the LifeKeeper Single Server Protection resource hierarchy**. In order for LifeKeeper Single Server Protection to control an application in the event of system failures or administrative actions, you must define a resource hierarchy. This definition implies creating resource instances and relationships for the applications.

There are three ways to define the LifeKeeper Single Server Protection resource hierarchy:

1. **Using the LifeKeeper GUI.**
2. **Using the LifeKeeper configuration database interface (LCDI) command**s (for applications dependent upon multiple disk partitions and/or file systems or any arbitrary application type).
3. **Using a combination of GUI and command functions**. The LCDI Command Interface topic provides a creation example using both the GUI and the LCDI commands.
4. **Establishing any optional application fault detection**. The ability to provide detection and alarming for problems within an application is valuable for building the best fault resilient solution. Because every specific application varies on the mechanism and format of failures, no one set of generic mechanisms meets all needs. LifeKeeper Single Server Protection does, however, provide a complete environment for defining errors, alarms and events that can trigger recovery procedures. For further information, see the LifeKeeper Single Server Protection Alarming and Recovery topic.

Types of Scripts

LifeKeeper Single Server Protection uses three principle kinds of recovery scripts: remove, restore and delete. After you develop these scripts, you place them in the appropriate directory so that they are available to LifeKeeper Single Server Protection in case of a failure situation: ***/opt/LifeKeeper/subsys/application/resources/type/actions***.

If you use the LifeKeeper GUI to define your application resource hierarchies, LifeKeeper Single Server Protection automatically places the scripts in the appropriate directories.

In addition to the **restore**, **remove** and **delete** scripts specific to each resource type, LifeKeeper Single Server Protection also interacts with recovery scripts that can supply global actions, actions performed either before or after **restore**, **remove** or **delete**:

- **preremove**. Actions required before the running of any remove scripts. Place preremove scripts in the directory ***/opt/LifeKeeper/subsys/application/actions***.
- **postremove**. Actions required after the running of any remove script. Place postremove scripts in the directory ***/opt/LifeKeeper/subsys/application/actions***.
- **prerestore**. Actions required before running any restore script. Place prerestore scripts in the directory ***/opt/LifeKeeper/subsys/application/actions***.
- **postrestore**. Actions required after running any restore script. Place postrestore scripts in the directory ***/opt/LifeKeeper/subsys/application/actions***.
- **predelete**. Actions required before running any delete scripts. Place predelete scripts in the directory ***/opt/LifeKeeper/subsys/application/actions***.
- **postdelete**. Actions required after running any delete scripts. Place postdelete scripts in the directory ***/opt/LifeKeeper/subsys/application/actions***.

Note that these scripts are run regardless of what application or resource type is having the operation performed. For example, if you supply a postremove script to be run after an out-of-service operation,

and a comm application is brought out of service, this script also runs after the comm remove script is run.

Script Parameters

When you write remove, restore and delete scripts, assume that LifeKeeper Single Server Protection will pass parameters of the following types when the scripts are executed:

- **-t tag.** Tag name of resource on which operation should be performed.
- **-i id.** ID name of resource on which operation should be performed.
- **[-R].** An optional flag passed for use by the error logging functions.

Restore Scripts

LifeKeeper Single Server Protection executes the restore script for a particular resource type whenever it needs to put an instance of that type into service. The restore script for the file system type resources is found in the directory ***/opt/LifeKeeper/subsys/gen/resources/filesys/actions***.

Application action scripts often have standard components. For example, the file system restore script has the same first three content components as the remove script. You can copy parts of those scripts to customize scripts for your applications.

The LRACI program that runs the restore script places the resource instance into the ISP state if it completes successfully (exit code 0) or the OSF state if it fails. The restore script must also successfully handle "restoring" already operational applications or resources.

The sample file system restore script illustrates the functional processing section of the file system restore script. Note these items of interest in the script code:

- Lines 101-108 check to see if the file system is already mounted. If it is, it is not treated as an error since the desired result is achieved. This situation could arise when restarting LifeKeeper Single Server Protection after lkstop -f. Including an analogous check at the beginning of your restore scripts facilitates later software upgrades.
- Line 110 retrieves the instance information.
- Lines 112-126 retrieve underlying device names and file system information.
- Lines 128-141, 181-188 deal with the inappropriate situation of the filesystem being root. A characteristic of a good shell script is the ability to anticipate the possible error situations and deal with them.
- Lines 143-151 make a mount point, if necessary.
- Lines 153-158 try to mount the file system and exit if successful.
- Lines 160-179 clean up the file system. These are normally executed for a system failure recovery because the file system was not unmounted properly.
- Lines 189-195 try to mount the file system again, and if this fails, try these other options:

- Lines 196-214 try to move the mount point.
- Line 215 tries to mount one last time.
- Lines 149, 207, 213, 223, 229 exit failing. LifeKeeper Single Server Protection marks the filesystem instance OSF and aborts the process of bringing this hierarchy into service.

Sample Restore Script

The following example provides the functional processing section of a file system restore script:

```
70 getchildinfo() {
71 OKAPP=$1
72 PTAG=$2
73 ret=1
74 if $LKROOT/bin/dep_list -c $PTAG | sed "s//_/g; s//_/g" >/tmp/DL$$
75 then
76 for i in `cat /tmp/DL$$`
77 do
78 l=`echo "$i" | sed "s/_/ /g; s/_/ /g"`
79 CHDTAG=`echo "$l" | cut -d_ -f2`
80 if CHDDATA=`$LKROOT/bin/ins_list -t$CHDTAG`
81 then
82 APP=`echo "$CHDDATA" | cut -d_ -f2`
83 if [ "$OKAPP" != "$APP" ]
84 then
85 continue
86 fi
87 ret=0
88 break
89 fi
90 done
91 fi
92 if [ "$ret" != 0 ]
93 then
```

```
94 pl "LifeKeeper: *ERROR* getchildinfo found no $OKAPP child for $PTAG"
95 fi
96 return $ret
97 }
98
99 pl "LifeKeeper: mounting file system $FSNAME"
100
101 # Check if the filesystemfile system exists
102 f=`sed -n "\?^.* $FSNAME ?p" /etc/mnttab`
103 if [ "$f" != "" ]
104 then
105 pl "LifeKeeper: file system $FSNAME already mounted"
106 err=0
107 exit 0
108 fi
109
110 if v=`$LKROOT/bin/ins_list -t"$TAG"`
111 then
112 if getchildinfo scsi $TAG
113 then
114 CTAG=$CHDTAG
115 else
116 exit 1
117 fi
118
119 # Get the mount information
120 INFO=`echo "$v" | cut -d_ -f6`
121 FSTYPE=`echo "$INFO" | cut -d_ -f1`
122 FSPERM=`echo "$INFO" | cut -d_ -f2`
123
```

```
124 fp=`echo "$CHDDATA" | cut -d_ -f5`
125 FPName=/dev/dsk/$fp
126 FPRawName=/dev/rdsk/$fp
127
128 # Test for root filesystemfile system
129 test $FSNAME = /
130 if [ "$?" -eq 0 ]
131 then
132 ROOT=1
133 else
134 ROOT=0
135 fi
136
137 # If root filesystemfile system: return code 0 means it is OK and mounted
138 if [ "$ROOT" -eq 1 ]
139 then
140 err=0
141 exit 0
142 else # If other than root filesystemfile system, mount it
143 if [ ! -d $FSNAME ]
144 then
145 mkdir $FSNAME
146 if [ $? != 0 ]
147 then
148 pl "LifeKeeper: can't make file system $FSNAME mount point"
149 exit 1
150 fi
151 fi
```

Remove Scripts

LifeKeeper Single Server Protection executes the remove script for a particular resource type whenever it needs to take a particular instance of that type out of service. This section discusses the remove script for file system type resources found on a LifeKeeper Single Server Protection system in the following directory: ***/opt/LifeKeeper/subsys/gen/resources/filesys/actions***.

The LRACI program that runs the remove script places the resource instance into the OSU state if it completes successfully (exit code 0) and leaves the state unchanged if it fails.

The topic Sections Common to Remove and Restore Scripts highlights the first three sections of the script that are the same in both the remove and the restore scripts. You are encouraged to copy these sections as a starting point for your own script development. The topic Sample Remove Script, providing the functional processing section of a file system remove script, displays the rest of the **remove** script.

These are the five basic parts of the recovery script:

- **Initialization**. All LifeKeeper Single Server Protection scripts source ***/etc/default/LifeKeeper*** in order to set certain required environment variables, most especially **PATH** and **LKROOT**.
- **Calling parameter parsing**. LifeKeeper Single Server Protection calls every recovery script with at least two options:
  - **-t** instance_tagname
  - **-i** instance_id

  The script may use either or both of these parameters to determine which specific instance to remove. In this example, the file system **remove** script uses only the instance_id (the file system mount point). If the instance is a resource type you created, then you would determine the format and meaning of the instance_id. Although the **-R** parameter is optional, you should always include lines 37-39 in recovery scripts because prfuncs uses the variable **RCVARG**.
- **Error logging**. Line 56 reads into the script the contents of the **prfuncs** file. This file contains a number of error logging utilities needed throughout the script. The **prfuncs** manual page describes the functions **log**, **pl** and **pt**.
- **Termination processing**. It is good shell-programming practice to include signal trapping so that the script can clean up after itself when it terminates, whether normally or otherwise. In the **remove** example, the trap function removes any temporary files and sends a completion message to the LifeKeeper Single Server Protection log.
- **Functional processing**. The part of the remove script shown in the topic Sample Remove Script actually removes the file system instance from service. The script specifies these actions:
  - **Checks that the file system is mounted**. The script first performs some checks to verify that the specified file system is mounted. If it is not mounted, the process does not need to unmount it but ends successfully because the intended result is achieved.
  - **Kills processes using the file system**. Before unmounting the file system, the script

kills off any processes using the file system so that it can be unmounted.

- **Unmounts the file system**. If the process cannot immediately unmount the file system, the script provides a wait time to be sure there has been enough time for user processes to die. The script sets **err** to **0** only after a successful unmount.

Sample Remove Script

The following example provides the functional processing section of a file system remove script:

```
72 if f=`sed -n "\?^.* $FSNAME ?p" /etc/mnttab`
73 then
74 if [ "$f" != "" ]
75 then
76 # Check if the file system is being used by a process;
77 # if so, kill the process
78 pl "LifeKeeper: must kill off any processes accessing file system $FSNAME before it can be unmounted."
79
80 # A timing window exists here -- additional users could
81 # access the FS after fuser issues the kill. In this
82 # case the umount would fail. So, we try 3 times or
83 # until fuser reports no processes
84 for try in 1 2 3
85 do
86 if [ `fuser -k -c "${FSNAME}" 2>/dev/null | wc -c` -eq 0 ]
87 then
88 # No more processes... (we have to be quick now!) Unmount
89 # the file system. The umount is done immediately -- this
90 # might mess up the log files if an error occurs, but if
91 # we issue the messages first, it just lengthens the timing
92 # window.
93 if umount "${FSNAME}" 2>/tmp/UM$$
94 then
95 # umount worked... issue the messages now (better late
```

```
96 # than not at all!)
97 pl "LifeKeeper: unmounting file system $FSNAME"
98 pl "\tumount ${FSNAME}"
99 pl "LifeKeeper: file system $FSNAME successfully unmounted"
100 err=0
101 exit 0
102 else
103 pl "LifeKeeper: unmounting file system $FSNAME"
104 pl "\tumount ${FSNAME}"
105 cat /tmp/UM$$ >&2
106 pl "LifeKeeper: *ERROR* file system $FSNAME failed unmount; will try again"
107 fi
108 fi
109 sleep 3 # processes might be slow to die, so wait a bit
110 done
111 else
112 pl "LifeKeeper: File system ${FSNAME} is not mounted."
113 err=0
114 exit 0
115 fi
116 else
117 exit 1
118 fi
119
120 #
121 # If we get here, the fuser above failed to kill all the active processes.
122 # The unmount of the file system will most likely fail, but what can we
123 # do?
124
125 sleep 5
```

```
126
127 pl "LifeKeeper: unmounting file system $FSNAME"
128 pl "\tumount ${FSNAME}"
129 if umount "${FSNAME}"
130 then
131 pl "LifeKeeper: file system $FSNAME successfully unmounted"
132 err=0
133 exit 0
134 else
135 pl "LifeKeeper: *ERROR* file system $FSNAME failed unmount"
136 exit 1
137 fi
```

Sections Common to Remove and Restore Scripts

```
1 #!/usr/bin/ksh
2 #ident "@(#)remove 1.1.1.2"
3 # Copyright 2000 SIOS Technology Corp., Mountain View, CA
4 #
5 # usage: remove -t tagname -i fsname (full path)
6 #
7
8 DEFAULT_FILE=/etc/default/LifeKeeper
9 if [ -z "$LKROOT" ]
10 then
11  PATH=
12  . $DEFAULT_FILE
13  LKROOT=${LKROOT:=/opt/LifeKeeper}
14  PATH=${PATH:=$LKROOT/bin:/usr/bin:/usr/sbin:/bin:/sbin}
15  export LKROOT PATH
16 fi
17
```

```
18 TAG=
19 FSNAME=
20 RCVARG=
21
22 while [ $# != 0 ]
23 do
24  case "$1" in
25  -t*)
26  if TAG=`$LKROOT/subsys/actions/testflag -t "$1""$2"`
27  then
28   shift
29  fi
30  ;;
31  -i*)
32  if FSNAME=`$LKROOT/subsys/actions/testflag -i "$1""$2"`
33  then
34   shift
35  fi
36  ;;
37  -R)
38  RCVARG=-R
39  ;;
40  esac
41  shift
42 done
43
44 if [ "$TAG" = "" ]
45 then
46  echo "$0: -t flag not specified"
47  exit 1
```

```
48 fi
49
50 if [ "$FSNAME" = "" ]
51 then
52  echo "$0: -i flag not specified"
53  exit 1
54 fi
55
56 . $LKROOT/subsys/actions/prfuncs
57
58 log "LifeKeeper: REMOVE FILE SYSTEM $FSNAME STARTAT:\n\t`date`"
59
60 err=1
61
62 trapfunc() {
63  rm -f /tmp/??$$
64  log "LifeKeeper: REMOVE FILE SYSTEM $FSNAME END err=$errAT:\n\t`date`"
65  exit $err
66 }
67
68 trap "trapfunc" 0 1 2 3 4 6 7 8 10 12 13 15 19
```

Delete Scripts

You need to create delete scripts only if there are some tasks inappropriate for a remove script that must be completed before an instance can be successfully deleted. The **file system** resource type includes a **delete** script, for example, because the process of creating a **file system** instance modified */etc/fstab* on both systems. When the instance is removed, the **fstab** files must be restored to their original state.

If you create your resources using the LifeKeeper GUI functions, LifeKeeper Single Server Protection will automatically manage the delete functions appropriately. If you have some similar kind of action that must be undone when LifeKeeper Single Server Protection removes an instance and you want LifeKeeper Single Server Protection to perform the action automatically, you can use the **delete** script mechanism.

Take care when creating **delete** scripts, however, because the delete protocol in LifeKeeper Single Server Protection is complex. You can use the file system delete script as an example. It is located in the directory ***/opt/LifeKeeper/subsys/gen/resource/filesys/actions***.

Note: Delete scripts should never be run manually but run only as part of running the ins_remove program.

Sample Notify Script

This sample script is called when a disk full condition is detected and is responsible for reporting the situation to the system administrator. Administrative intervention is required to correct this condition. The script provides an example for sending email to an administrator. Edit this script, located at ***/opt/LifeKeeper/events/filesys/diskfull/notify***, as needed to enable these notification features.

Local Recovery Scripts

You can also develop scripts that specify local recovery actions that precede LifeKeeper Single Server Protection's inter-server recovery functions. You can find an example of a local recovery script in the directory ***/opt/LifeKeeper/subsys/gen/resources/filesys/recovery/filesys***.

Maintenance Tasks

This section includes topics for maintaining LifeKeeper Single Server Protection.

File System Health Monitoring

The File System Health Monitoring feature detects conditions that could cause LifeKeeper Single Server Protection protected applications that depend on the file system to fail. Monitoring occurs on active/in-service resources (i.e. file systems) only. The two conditions that are monitored are:

- A full (or almost full) file system, and
- An improperly mounted (or unmounted) file system.

When either of these two conditions is detected, one of several actions might be taken.

- A warning message can be logged and email sent to a system administrator.
- Local recovery of the resource can be attempted.
- A reboot of the server can be done.

Condition Definitions

Full or Almost Full File System

A "disk full" condition can be detected, but cannot be resolved - administrator intervention is required. A message will be logged by default. Additional notification functionality is available. For example, an

email can be sent to a system administrator, or another application can be invoked to send a warning message by some other means. To enable this notification functionality, edit the sample notify script.

In addition to a "disk full" condition, a "disk almost full" condition can be detected and a warning message logged in the LifeKeeper Single Server Protection log.

The "disk full" threshold is:

```
FILESYSFULLERROR=95
```

The "disk almost full" threshold is:

```
FILESYSFULLWARN=90
```

The default values are 90% and 95% as shown, but are configurable via tunables in the */etc/default/LifeKeeper* file. The meanings of these two thresholds are as follows:

`FILESYSFULLWARN` - When a file system reaches this percentage full, a message will be displayed in the LifeKeeper Single Server Protection log.

`FILESYSFULLERROR` - When a file system reaches this percentage full, a message will be displayed in the LifeKeeper Single Server Protection log as well as the system log. The file system notify script will also be called.

Unmounted or Improperly Mounted File System

LifeKeeper Single Server Protection checks the */etc/mtab* file to determine whether a LifeKeeper Single Server Protection protected file system that is in service is actually mounted. In addition, the mount options are checked against the stored mount options in the *filesys* resource information field to ensure that they match the original mount options used at the time the hierarchy was created.

If an unmounted or improperly mounted file system is detected, local recovery is invoked and will attempt to remount the file system with the correct mount options.

If the remount fails, a reboot will be performed to resolve the condition. The following is a list of common causes for remount failure which would lead to a reboot:

- corrupted file system (fsck failure)
- failure to create mount point directory
- mount point is busy
- mount failure
- LifeKeeper Single Server Protection internal error

Log File Messages

Following are examples of LifeKeeper Single Server Protection log messages related to file system problems. Items in ***bold-italic*** are variables and will change in the output messages.

This warning is logged when an improperly mounted or unmounted file system is detected:

```
quickCheck: WARNING: time
```

```
        LifeKeeper Single Server Protection protected filesystem tag
        (id) is in service but not mounted (properly) (ro instead of
        rw)

    quickCheck: Attempting Local Recovery of resource tag
```

This warning is logged when a full (almost full) file system is detected:

```
    quickCheck: WARNING: time

        LifeKeeper Single Server Protection protected filesystem tag
        (id) is percent%full (blocksfree freeblocks).
```

One of these messages is logged when the local recovery of an unmounted/improperly mounted file system is finished:

```
    time:

    LOCAL RECOVERY OF RESOURCE tag HAS SUCCEEDED (error)

    LOCAL RECOVERY OF RESOURCE tag HAS FAILED (error)
```

Maintaining a Resource Hierarchy

You can perform maintenance on a resource hierarchy while maintaining LifeKeeper Single Server Protection 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.

Changing LifeKeeper Single Server Protection Configuration Values

There are a number of values in LifeKeeper Single Server Protection that may need to be changed after LifeKeeper Single Server Protection has been configured and set up. Examples of values that may be modified include the uname of LifeKeeper Single Server Protection servers, ip resource addresses and tag names. To change these values, carefully follow the instructions below.

1. Stop LifeKeeper Single Server Protection on your server using the command:

```
        LKROOT/bin/lkstop
```

2. If you are changing the uname of a LifeKeeper Single Server Protection server, change the server's hostname using the Linux **hostname(1)** command.
3. If more than one LifeKeeper Single Server Protection value is to be changed, old and new values should be specified in the following format:

   ```
   old_value1=new_value1
   ....
   old_value9=new_value9
   ```

4. Verify that the changes to be made do not have any unexpected side effects by examining the output of running the **lk_chg_value** command. If there is more than one value to change, run the command:

   ```
   $LKROOT/bin/lk_chg_value -Mvf file_name
   ```

   where *file_name* is the name of the file created in Step 4.

   If there is only one value to change, run the command:

   ```
   $LKROOT/bin/lk_chg_value -Mvo old_value -n new_value
   ```

   The **-M** option specifies that no modifications should be made to any LifeKeeper Single Server Protection files.

5. Modify LifeKeeper Single Server Protection files by running the **lk_chg_value** command without the **-M** option on all servers in the cluster. If there is more than one value to change, run the command:

   ```
   $LKROOT/bin/lk_chg_value -vf file_name
   ```

   where ***file_name*** is the name of the file created in Step 4.

   If there is only one value to change, run the command:

   ```
   $LKROOT/bin/lk_chg_value -vo old_value -n new_value
   ```

6. Restart LifeKeeper Single Server Protection using the command:

   ```
   $LKROOT/bin/lkstart
   ```

   Note: It may be necessary to close and restart the GUI.

Notes:

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

Running LifeKeeper Single Server Protection With a Firewall

LifeKeeper Single Server Protection for Linux can work with a firewall in place on the same server if

you address the following network access requirements.

Note: If you wish to simply disable your firewall, see Disabling a Firewall below.

LifeKeeper GUI Connections

The LifeKeeper GUI uses a number of specific TCP ports, including Ports 81 and 82 as the default initial connection ports. The GUI also uses Remote Method Invocation (RMI), which uses Ports 1024 and above to send and receive objects. All of these ports must be open in the firewall on each LifeKeeper Single Server Protection server to at least those external systems on which the GUI client will be run.

LifeKeeper Single Server Protection IP Address Resources

The firewall should be configured to allow access to any IP address resources in your LifeKeeper Single Server Protection hierarchies from those client systems that need to access the application associated with the IP address.

LifeKeeper Single Server Protection also uses a broadcast ping test to periodically check the health of an IP address resource. This test involves sending a broadcast ping packet from the virtual IP address and waiting for the first response from any other system on the local subnet. To prevent this test from failing, the firewall on each LifeKeeper Single Server Protection server should be configured to allow the following types of network activity.

- Outgoing Internet Control Message Protocol (ICMP) packets from the virtual IP address (so that the active LifeKeeper Single Server Protection server can send broadcast pings)
- Incoming ICMP packets from the virtual IP address (so that other LifeKeeper Single Server Protection servers can receive broadcast pings)
- Outgoing ICMP reply packets from any local address (so that other LifeKeeper Single Server Protection servers can respond to broadcast pings)
- Incoming ICMP reply packets to the virtual IP address (so that the active LifeKeeper Single Server Protection server can receive broadcast ping replies)

Disabling a Firewall

If you wish to disable your firewall, then do the following:

1. Stop the firewall using one of the following commands, depending upon your firewall package:

   ```
   /etc/init.d/ipchains stop or

   /etc/init.d/iptables stop
   ```

   If operating in an IPv6 environment, be sure to account for `ip6tables`

   ```
   /etc/init.d/ip6tables stop
   ```

   If running SuSE Linux Enterprise Server

   ```
   /etc/init.d/SuSEfirewall2_init stop
   ```

```
/etc/init.d/SuSEfirewall2_setup stop
```

2. Either remove the package (using **rpm -e**) or disable its startup using one of the following commands, depending upon your firewall package:

   ```
   /sbin/chkconfig --del ipchains or

   /sbin/chkconfig --del iptables

   /sbin/chkconfig --del ip6tables
   ```

   If running SuSE Linux Enterprise Server, you must manage `SuSEfirewall2` configuration settings.

Running the LifeKeeper GUI Through a Firewall

In some situations, a LifeKeeper cluster is placed behind a corporate firewall and administrators wish to run the LifeKeeper GUI from a remote system outside the firewall.

LifeKeeper uses Remote Method Invocation (RMI) to communicate between the GUI server and client. The RMI client must to be able to make connections in each direction. Because the RMI client uses dynamic ports, you can not use preferential ports for the client.

One solution is to use ssh to tunnel through the firewall as follows:

1. Make sure your IT department has opened the secure shell port on the corporate firewall sufficiently to allow you to get behind the firewall. Often the machine IT allows you to get to is not actually a machine in your cluster but an intermediate one from which you can get into the cluster. This machine must be a Unix or Linux machine.
2. Make sure both the intermediate machine and the LifeKeeper server are running sshd (the secure shell daemon) and that X11 port forwarding is enabled (this is usually the line `X11Forwarding yes' in */etc/ssh/sshd_config*, but if you are unsure, have your IT do this for you.
3. From your Unix client in X, tunnel to the intermediate machine using:

   ```
   ssh -X -C <intermediate machine>
   ```

   The **-C** means `compress the traffic' and is often useful when coming in over slower internet links.
4. From the intermediate machine, tunnel to the LifeKeeper server using:

   ```
   ssh -X <LifeKeeper server>
   ```

   You should not need to compress this time since the intermediate machine should have a reasonably high bandwidth connection to the LifeKeeper server.
5. If all has gone well, when you issue the command:

   ```
   echo $DISPLAY
   ```

   it should be set to something like `*localhost:10.0*'. If it is not set, it is likely that X11 forwarding is disabled in one of the sshd config files.

6. Verify that you can pop up a simple *xterm* from the LifeKeeper server by issuing the command:

   ```
   /usr/X11R6/bin/xterm
   ```

7. If the *xterm* appears, you're ready to run **lkGUIapp** on the LifeKeeper server using the following command:

   ```
   /opt/LifeKeeper/bin/lkGUIapp
   ```

8. Wait (and wait some more). Java uses a lot of graphics operations which take time to propagate over a slow link (even with compression), but the GUI console should eventually appear.

Removing LifeKeeper Single Server Protection

You can uninstall the LifeKeeper Single Server Protection product simply by running the included rmlk utility:

/opt/LifeKeeper/bin/rmlk

This will remove all SIOS product rpms and remove the */opt/LifeKeeper* directory from the system. **Use with caution.**

FAQs

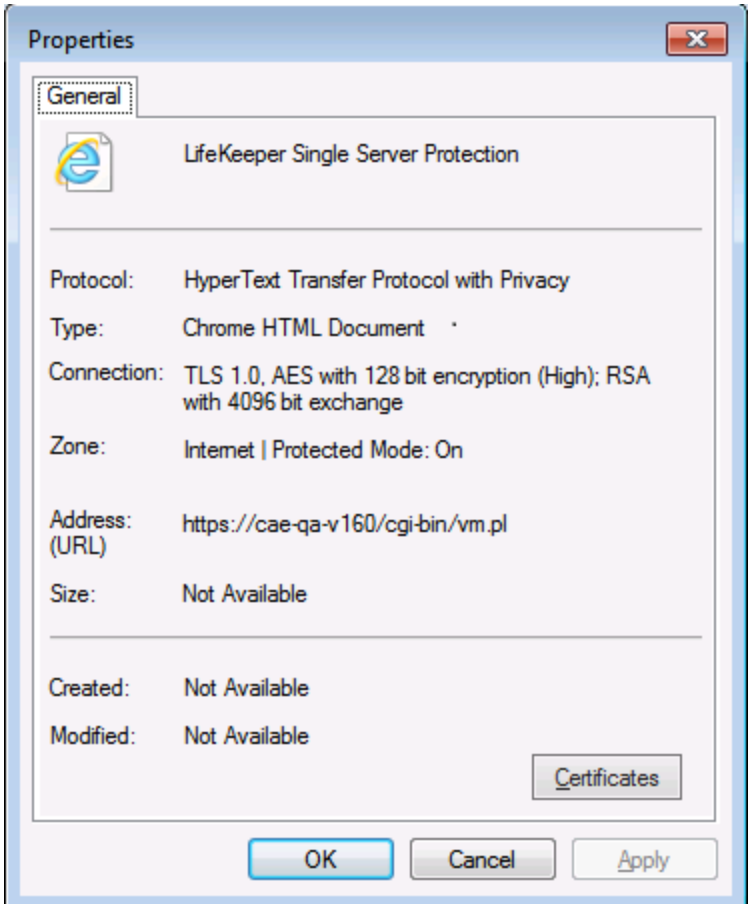
SMC

Question

Is there any way (from the plug-in) to tell which SMC I am using?

Answer

Right-click and view **Properties** of the plug-in web page.



Answer

Troubleshooting

This section contains restrictions and/or known issues open against LifeKeeper Single Server Protection as well as SMC troubleshooting hints and tips.

For more troubleshooting information, see the LifeKeeper Technical Notes and Troubleshooting topics in the LifeKeeper Technical Documentation.

Known Issues and Workarounds

Included below are the restrictions and/or known issues open against LifeKeeper Single Server Protection.

Core

Bug 2257

Access to LifeKeeper Single Server Protection and SIOS Protection Suite nodes via

credstore requires proper credstore key

Solution: When storing credentials for a LifeKeeper Single Server Protection or SIOS Protection Suite node using `credstore`, you must use the proper form of the hostname for the credstore credentials key (i.e. `credstore -k <hostname>`):

For the LifeKeeper Single Server Protection plugin, `credstore` should be run using the hostname of the system as reported in the **Hostname:** field of the LifeKeeper Single Server Protection plugin display.

For SIOS Protection Suite, the hostname used to store credentials must be the same as the one you plan to use in the command line tool's (e.g., `lkpolicy`) -d argument. For example, if you want to run **`lkpolicy -d mynode1`**, then you must store credentials using **`credstore -k mynode1`**. You cannot store credentials using the FQDN in this case. If you do, you must run **`lkpolicy -d`** *FQDN*.

Workaround: If you've stored a default credential set (i.e., **`credstore -k default`**) that works for all your LifeKeeper Single Server Protection and/or SIOS Protection Suite nodes, then you will not be affected by this issue.

Bug 2408

HA hearbeat incorrectly enabled

lkvmhad incorrectly enables the HA hearbeat after second resource failure

Workaround: Set LKCHECKINTERVAL in */etc/default/LifeKeeper* greater than the VMware HA, VM Monitoring Failure Interval. **Note:** The LKCHECKINTERVAL default is 120 seconds. This is also the default 'low' monitoring sensitivity for VMware HA, VM Monitoring.

Cannot install the SMC when an openssl-devel has not been installed

An openssl-devel must be installed in advance when installing SMC v8.3.2. If the openssl-devel has not been installed, the install of SMC will fail outputting the message as follows:

ld -shared -o ./lib/Crypt-SSLeay-0.55-0.9.8/lib/auto/Crypt/SSLeay/SSLeay.so
./lib/Crypt-SSLeay-0.55-0.9.8/lib/auto/Crypt/SSLeay/SSLeay.o -lcrypto -lssl
ld: cannot find -lcrypto
Unable to link the Crypt::SSLeay Perl module. Secured connections will be unavailable until you install the Crypt::SSLeay module.
So required libcrypt.a in system library.

Workaround: When the message above is shown in executing the setup script, you must execute the setup script again after installing the openssl-devel included in the OS installer.

GUI

Refresh problem with LifeKeeper Single Server Protection GUI

The GUI may occasionally scramble the resource tree (i.e., resource dependencies may not be shown correctly).

Workaround: Perform a refresh of the GUI.

Apache

Apache resource creation fails

Example of Error message:

Error: valid_http_root: Since "/usr/sbin/httpd" is shareable on "/usr", "/etc/httpd" must be also

Cause:

Due to a defect, files in mount point "/"(root) cannot be detected appropriately.

For example, if "/etc/httpd" is in a same filesystem as the mount point "/", a resource creation will fail.

Workaround:

Mount one of the below workarounds to avoid this issue.

(a) Transfer such as "/etc/httpd" under the other mount point.

(b) Mount " /etc" to such as" /dev/sdb1".

Oracle

Bug 2387

Cannot create an Oracle hierarchy on root file system in LifeKeeper Single Server Protection environment

Workaround: Using the following procedure, copy Oracle to a new file system.

Create a new disk large enough for Oracle data (e.g. /dev/sdb). (Note: You can size up /oracle directory to get an idea how big this should be; multiply by at least 50% to allow for logs)

Using fdisk, create a new partition on that disk.

```
fdisk /dev/sdb
```

Make a file system.

mkfs -t ext3 /dev/sdb1

Mount this file system (example using /mnt/oracle).

mkdir /mnt/oracle

mount /dev/sdb1 /mnt/oracle

Stop Oracle, Listener.

Copy Oracle to new file system.

cd /oracle

cp -a * /mnt/oracle

(**Note**: This step may take some time based on the amount of data)

Unmount the new file system.

umount /mnt/oracle

Mount the new file system over /oracle.

mount /dev/sdb1 /oracle

Start Listener and then Oracle.

The Oracle Recovery Kit does not support Oracle Database Standard Edition 2 (SE2) on AWS EC2 system

During Oracle Database Standard Edition 2 (SE2) test, an unknown behavior was reported on AWS EC2 system. However, we confirmed that other services except EC2 did not have the same issue, meaning that Oracle Recovery Kit supports SE2 on AWS EC2 system except EC2.

SAP

Bug 2388

For SAP, hierarchies cannot be created using the GUI

Workaround: Use the command line option to create hierarchies. However, at the end of the command line, specify the number 76 as follows:

> $LKROOT/lkadm/subsys/appsuite/sap/bin/create <primary sys> <tag> <SAP SID> <SAP Instance> <switchback type> <IP Tag> <Protection Level> <Recovery Level> <Additional SAP Dependents> 76

See "Setting Up SAP from the Command Line" for further command line information.

Also, refer to Know Issues and Restrictions at SIOS Protection Suite for Linux.

SMC Troubleshooting

See below for troubleshooting hints and tips.

LifeKeeper Single Server Protection monitored guest virtual machines must have VMware Tools installed.

LifeKeeper Single Server Protection is unable to completely connect to virtual machines that do not have working installations of VMware tools. SteelEye Management Console will not be able to connect to these machines to get complete status of the machine and its resources. The LifeKeeper Single Server Protection plugiin in the vSphere Client will show error messages for these guests (where the tools are not installed).

Solution: Make sure the VMware Tools are installed on the guest machines that will be protected by LifeKeeper Single Server Protection.