SIOS

LifeKeeper® for Linux

IP Recovery Kit v7.3 Administration Guide

February 2011

It is the policy of SIOS Technology Corp. (previously known as SteelEye Technology, Inc.) to improve products as new technology, components, software, and firmware become available. SIOS Technology Corp., therefore, reserves the right to change specifications without prior notice.

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

Email correspondence to:
ip@us.sios.com

Table of Contents

IP Recovery Kit Administration Guide

Introduction

The LifeKeeper® for Linux Internet Protocol (IP) Recovery Kit provides a mechanism to recover an IP address from a failed primary server to a backup server in a LifeKeeper environment. The IP Recovery Kit can define an IP address that can be used to connect to a LifeKeeper-protected application. As with other LifeKeeper resources, IP resource switchovers can be initiated automatically as a result of a failure or manually by an administrative action.

The IP Recovery Kit supports the implementation of the TCP/IP protocol suite using the logical interface feature, allowing it to provide switchover and failover of IP addresses without requiring extra standby network interface cards (unless you are utilizing the local recovery feature) or *dummy* IP addresses.

Document Contents

This guide explains the following topics:

- **LifeKeeper Documentation.** A list of all the LifeKeeper for Linux documentation and where the information is available.
- **Requirements.** Before you can install and set up the recovery software, your server must meet certain hardware and software requirements. You should refer to the *LifeKeeper for Linux Planning and Installation Guide* for specific instructions on how to install or remove the LifeKeeper IP Recovery Kit.
- **Configuring Your Recovery Kit.** To ensure that your LifeKeeper configuration provides the protection and flexibility you require, you need to be aware of the configuration rules. To appropriately plan your configuration, you must understand your network configuration, interface selection, user system setup, hierarchy options and the IP configuration tasks. In addition to planning your configuration, this section also includes configuration examples and the specific tasks required to configure your recovery kit.
- **Troubleshooting.** This section provides a list of informational and error messages with recommended solutions.

LifeKeeper Documentation

The following LifeKeeper product documentation is available from SIOS Technology Corp.:

- *LifeKeeper for Linux Release Notes*
- *LifeKeeper for Linux Online Product Manual* (available from the Help menu within the LifeKeeper GUI)
- *LifeKeeper for Linux Planning and Installation Guide*

This documentation, along with documentation associated with optional LifeKeeper recovery kits, is available on the SIOS Technology Corp. website at:

http://us.sios.com/support

The following is a reference document associated with TCP/IP and the LifeKeeper IP Recovery Kit:

- RFC826 (Address Resolution Protocol) Document

Requirements

Before attempting to install or remove the IP Recovery Kit, you must understand the hardware and software requirements for the package and the installation and removal procedures.

Kit Hardware and Software Requirements

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

- **Servers.** The recovery kit requires two or more *supported* computers configured in accordance with LifeKeeper requirements described in the *LifeKeeper Online Product Manual* and the *LifeKeeper Release Notes*, which are shipped with the product media.
- **LifeKeeper software.** You must install the same version of LifeKeeper software and any patches on each server. Please refer to the *LifeKeeper Release Notes* and *Online Product Manual* for specific LifeKeeper requirements.
- **LifeKeeper IP Recovery Kit.** You must have the same version of this recovery kit on each server.
- **IP network interface.** Each server requires at least one Ethernet TCP/IP-supported network interface. In order for IP switchover to work properly, user systems connected to the local network should conform to standard TCP/IP specifications.

  Note: Even though each server requires only a single network interface, you should use multiple interfaces for a number of reasons; for example, heterogeneous media requirements, throughput requirements, elimination of single points of failure, network segmentation, and local recovery support.
- **TCP/IP software.** Each server also requires the TCP/IP software.

Consult the *LifeKeeper Release Notes* or your sales representative for the latest release compatibility and ordering information.

You should refer to the *LifeKeeper for Linux Planning and Installation Guide* for specific instructions on how to install or remove the LifeKeeper IP Recovery Kit.

Principles of Operation

LifeKeeper brings an IP resource into service by creating an IP *alias* address, or logical interface, on one of the physical network interfaces on the primary server. Users connect to the node using this alias address.

The IP software performs checks to help ensure that the selected address, network mask, and interface can function properly. The software verifies the following elements:

- **Unused resource.** The new IP address is not already assigned to any other IP resource in the LifeKeeper cluster.
- **Unique address.** The address cannot be *currently* active on the network. In addition to checking during creation, the software also performs the uniqueness check immediately before bringing the resource into service. If the software detects a duplicate address on the net, it does *not* bring the resource into service.

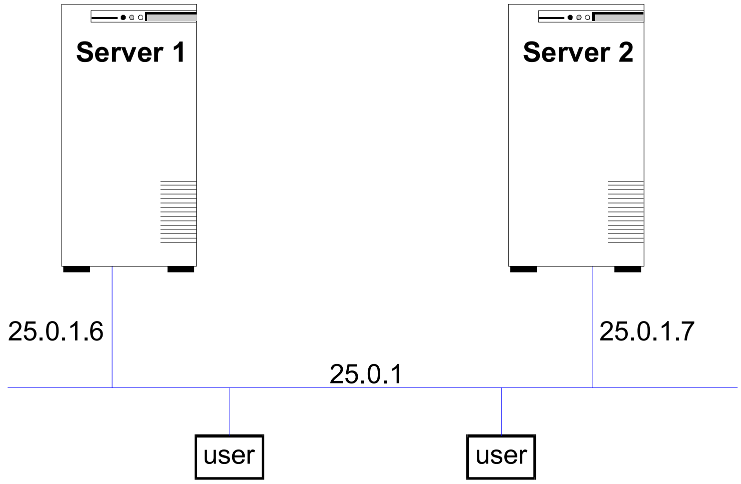
When the primary server fails, the IP Recovery Kit brings the IP resource into service on a backup server by creating a logical interface on one of that server's physical network interfaces.

Since session context is lost following recovery, after the recovery, IP users must reconnect using exactly the same procedures they used to connect originally.

In a manual switchover, the IP Recovery Kit removes the address from service on the active server before adding it to the backup server.

To clarify the administration and operation of the IP Recovery Kit, consider the scenario shown in Figure 1. This example configuration contains two servers, Server1 and Server 2. Each server has a single LAN interface, *eth0*, connected to subnet 25.0.1. The user systems are also on this subnet. The LAN interfaces on Server 1 and Server 2 have addresses 25.0.1.6 and 25.0.1.7, respectively.

Figure 1. Administration and Operation Scenario



The system administrator decides to use 25.0.1.10 as the address for an IP resource, to be called *ipname*. The administrator creates entries in the */etc/hosts* files (and in the DNS, if used), similar to the following:

25.0.1.6	server1
25.0.1.7	server2
25.0.1.10	ipname

Assuming that Server 1 is the primary server for the resource, the administrator creates the IP resource hierarchy for *ipname* on Server 1 using the wizard described in the section entitled *Creating an IP Resource Hierarchy*. The software finds the address associated with *ipname* (25.0.1.10) from */etc/hosts*, verifies that it is available, and brings it into service by creating a logical interface with that address on *eth0* on Server 1. *eth0* on Server 1 now responds to both Server 1 and *ipname*.

The new logical interface in this case would likely be called *eth0:1*, where the *:1* suffix indicates that this is a logical interface associated with the physical interface *eth0*. The IP Recovery Kit chooses the first available logical interface number at the time that the IP resource is being brought into service, so the chosen number could vary depending on what other logical interfaces are defined. This can be checked using the **ifconfig** command.

Users can then connect to Server 1 by entering, for example, **telnet ipname**. If Server 1 crashes, LifeKeeper automatically creates a logical interface with the *ipname* address using *eth0* on Server 2. The user sessions on Server 1 terminate. When users re-run **telnet ipname**, they are routed to Server 2.

Regardless of where *ipname* is actively *in service*, addresses *server1* and *server2* are active and usable, though not protected by LifeKeeper recovery. The addresses could be used for any cases that require connection to a specific server by name rather than to a switched application. Examples might include remote system management and the LifeKeeper communications path. (In this case, for example, 25.0.1.6 and 25.0.1.7 would be used for the LifeKeeper communications path.)

IP Resource Monitoring

LifeKeeper monitors the health of the IP resources under its control on a periodic basis, using the following techniques, in this order.

1. Check the link status for the network interface on which the IP resource is configured to determine whether the interface is properly connected to the physical network.
2. Verify that the IP resource is still configured as an alias on the appropriate network interface.
3. Perform a broadcast ping test or ping a pre-configured list of addresses, using the protected IP address as the source address of the pings, to determine whether the IP resource can successfully send and receive data on the network.

   The broadcast ping test is the default test mechanism. It operates by sending a broadcast ping packet to the broadcast address of the subnet associated with the IP resource, using the protected IP address as the source address. If a response is received from any address other than addresses on the local system, the test is considered successful. To reduce the network traffic associated with these tests, LifeKeeper will save the address of the first response and perform direct pings to that address during future tests, falling back to the broadcast mechanism only if the direct ping fails.

> For environments in which there are no systems on the network that can respond to the broadcast ping test, LifeKeeper also offers the ability to configure a list of addresses to be pinged as an alternative to the broadcast ping test. If such a list has been specified, the broadcast ping test is skipped, and all of the addresses in the list are pinged in parallel. The test is considered successful if a ping response is received from any one of the addresses in the *Ping List*.

If any of these tests fail during the periodic health check of an IP resource, LifeKeeper is notified of the failure. LifeKeeper will first attempt a local recovery operation to try to restore the IP resource to a working state on the local node. See the section *IP Local Recovery and Configuration Considerations* for more information about the local recovery procedure. If local recovery is unsuccessful in restoring the IP resource to a working state, LifeKeeper will then attempt to migrate the application hierarchy containing the IP resource to another LifeKeeper system in the cluster.

LifeKeeper also uses these same health checks to verify the proper operation of an IP resource immediately after it is brought in-service. A failure of any of the checks will cause the in-service operation to fail.

The IP health check mechanisms can be tuned and adjusted in many ways. See the sections *Viewing/Editing IP Configuration Properties* and *Adjusting IP Recovery Kit Tunable Values* for details.

Configuring TCP/IP with LifeKeeper

This section contains information you should consider before you start to configure TCP/IP and examples of typical LifeKeeper IP configurations.

Please refer to your *LifeKeeper Online Product Manual* for instructions on configuring your LifeKeeper Core resource hierarchies.

Specific Configuration Considerations for TCP/IP

In order to properly configure your IP Recovery Kit, you should review the following topics to ensure that you have the information necessary to complete the configuration tasks:

- **Interface Selection**
- **User System Setup**
- **General IP Planning Considerations**

Interface Selection

When creating an IP resource, you select the IP resource address, the netmask to use with the address, and the network interface. Not all combinations are allowed. The address/netmask pair you provide, and all the address/netmask pairs currently *in-service*, determine your choices. You should also see the section on *IP Local Recovery* for additional configuration considerations if you are planning on using this feature of the recovery kit.

The selected address/netmask determines the subnet for the resource. If another address on the same subnet (either a physical or logical interface address) is currently *in-service* on any interface, then the IP resource *must* be configured on that interface. The software performs tests to determine the allowed choices based upon the current network configuration. You can select from any of the choices provided.

Because the IP software does not distinguish between physical media types, you must determine the physical network for the resource and select the address appropriately. For example, assume that you have a server connected to an Ethernet backbone on subnet xx.yy.12 and Ethernet LANs on subnets xx.yy.20 and xx.yy.30. If you want to create a resource on the first Ethernet subnet, select an address on that subnet, such as xx.yy.20.120.

In general, even though the IP recovery software allows you to select almost any value for the netmask, you should avoid selecting multiple netmasks for the same physical interface because multiple masks can cause packet misrouting.

One further consideration is the need to be consistent in your selection of interfaces on all LifeKeeper servers. If you configure several IP resources on a single interface on Server A, they should also be configured on a single interface on Server B.

User System Setup

When the IP software switches an IP resource from one server to another, the MAC address associated with the switched IP address changes because the interface changes. Each router and user system on the LAN must reflect this change in its ARP table before it can contact the IP address at its new location. In certain operating systems, when a new IP address is added to a network interface, an ARP packet is automatically sent out by the operating system to update all clients' ARP tables on the subnet. This feature does not exist in Linux. LifeKeeper therefore must send out an ARP packet after adding a switchable IP address to an interface to force this client ARP cache update.

TCP/IP implementations differ in their ability to implement the required ARP updates in response to this ARP packet. The following list describes some important cases:

- **Full Linux TCP/IP implementation.** Fully functional TCP implementations in Linux and most other operating systems support ARP cache updates when the systems receive an ARP request packet. LifeKeeper uses this feature, as described above, to force ARP cache updates on such systems.
- **ARP cache.** User systems that do not support the ARP refinements but do support an ARP cache usually have a timer associated with the cache to maintain some level of currency. For some implementations, decreasing the timer value can minimize the time required for that particular user system to reflect the changed address mapping. If the number of users on the LAN is small, this option may be acceptable. For other systems, decreasing the timer value may not be necessary. For example, the TCP implementation shipped with Windows NT uses a ten second timer value, so no change in timer value would be needed.
- **Static address mapping.** For systems without a dynamic ARP cache or those where cache timing is not tunable, routers can be used to handle mapping changes. Such user systems would access the IP resource subnet by way of a router (gateway). In this configuration, cache update is needed only for the routers directly connected to the resource subnet and no changes are needed on the user systems themselves.

General IP Planning Considerations

After you have selected the addresses, netmasks, and associated host/domain names you intend to use for IP resource hierarchies, add the appropriate entries to each server's */etc/hosts* file, and to the Domain Name Server (DNS), if used.

> **Note:** Even if you are using a DNS, it is strongly recommended that you place entries for the IP resources in the local */etc/hosts* files on all LifeKeeper servers. This will reduce recovery times. However, if the resource name that you enter when creating the IP instance is the IP address itself, then the host file entry is unnecessary.

Do not configure these IP addresses into your system as you would if you were creating a permanent logical interface to be activated at system boot time.

If any of the resource addresses are on new (logical) subnets, update routers to handle routing to these subnets.

IP Resource Monitoring and Configuration Considerations

By default, the LifeKeeper IP Recovery Kit monitors IP resources by executing a broadcast ping on the IP addresses logical subnet, then listening for replies. For this test to work properly, at least one additional non-LifeKeeper system capable of responding to broadcast pings must exist on the physical network, with an IP address on the same logical subnet as the IP resource. A router on the same logical subnet is usually sufficient to meet this need.

If this requirement cannot be met, you can choose to either disable the broadcast ping test completely, or you can configure a static list of IP addresses that should be pinged as an alternative to the broadcast ping test mechanism. See the *Adjusting IP Recovery Kit Tunable Values* and *Viewing/Editing IP Configuration Properties* sections for more information about how to configure these options.

IP Local Recovery and Configuration Considerations

The standard Linux NIC bonding mechanism is the **recommended** means of providing network interface redundancy in a high availability configuration. The LifeKeeper IP Recovery Kit fully supports the creation of virtual IP addresses on bonded interfaces.

However, if NIC bonding cannot be used in your environment, the IP local recovery feature allows LifeKeeper to move a protected IP address from the interface on which it is currently configured to another interface in the same server when the IP Recovery Kit detects a failure. Local IP recovery provides you an optional backup mechanism, so that when a particular interface fails on a server, the protected IP address can be made to function on the backup interface, therefore avoiding an entire application/resource hierarchy failing over to a backup server.

It should be noted that even when a backup interface is not defined and an IP resource instance fails for some reason, local recovery will attempt to bring the resource instance back in-service on the current primary interface before initiating a resource failover to a backup server.

Local Recovery Scenario

IP local recovery allows you to specify a single backup network interface for each LifeKeeper-protected IP address on a server. Configuring more than one protected virtual IP address on a single backup network interface is not supported. Use the Linux NIC bonding mechanism for these configurations.

When a failure of an IP address is detected by the IP Recovery Kit, the resulting failure triggers the execution of the IP local recovery script. LifeKeeper will first attempt to bring the IP address back in-service on the current network interface. If that fails, LifeKeeper will check the resource instance to determine if there is a backup interface available. If so, it will then attempt to move the IP address to the backup interface. If all local recovery attempts fail, LifeKeeper will perform a failover of the IP address and all dependent resources to a backup server.

When the protected IP address is moved to the backup interface, the original backup interface becomes the primary, and vice versa. The result is that during LifeKeeper startups, switchovers, and failovers, LifeKeeper will always attempt to bring the IP address in-service on the interface on which it was last configured.

Local Recovery Configuration Restrictions

In order for the backup interface to work properly, it must be attached to the same physical network as the primary interface. You need to make sure that a valid interface is being chosen. Note that it is completely reasonable and valid to specify a backup interface on one server but not on another within the cluster (i.e., the chosen backup interface on one server has no impact on the choice of a backup on any other server).

To support both redundant TCP communication paths and a backup interface for IP Local Recovery, a server ideally needs 3 network adapters, 2 on the same physical network and 1on an independent network. (A TTY communication path is an alternative that can reduce the number of required adapters to 2.)

It is critically important that you understand the concept of TCP/IP subnets in selecting and configuring a backup interface, since the movement of an IP address from one interface to another has a dramatic impact on any other addresses on the server(s) which share the same subnet. If the IP address being moved is on the same logical subnet as any other active IP addresses on the server, those addresses will be unconfigured during the move operation, unless they are already active on the backup interface. This impact is necessary to insure proper subnet routing, and applies equally to primary addresses of an interface as well as alias addresses, including other LifeKeeper-protected addresses.

With this impact in mind, you must choose between the following two options when planning and configuring LifeKeeper-protected IP addresses with a backup interface.

1. Configure the LifeKeeper-protected address on its own logical subnet, distinct from the subnets of all non-LifeKeeper addresses on the server.
2. Configure the LifeKeeper-protected address on the same logical subnet as the addresses of both the primary and backup interfaces.

The most important factor in choosing the correct option is the configuration of the server's default route to a router or gateway device. If the server has a default route, and if that route is configured such that it is normally accessed via the interface that is also the primary interface for a LifeKeeper-protected address, LifeKeeper is capable of moving the default route to the backup interface, along with the LifeKeeper-protected address, in the event of a failure of the primary interface. This allows the server to continue to communicate beyond its local subnet (e.g. to the internet) following such a failure.

If you want LifeKeeper to switch the default route to the backup interface, you must choose option 2 above, configuring the LifeKeeper-protected address on the same logical subnet as the addresses of both the primary and backup interfaces. The backup interface should be configured such that it is not automatically activated at boot time, to ensure that only one interface is used as the default route at any given point in time. You must also ensure that the default route configuration is not tied to a specific network device. For Red Hat Enterprise Linux (RHEL) installations, this means that the GATEWAYDEV entry should not be included in *the /etc/sysconfig/network* file. For SUSE SLES installations, the default route entry in */etc/sysconfig/network/routes* should contain a dash (-) in the device field.

If there is no need or desire for LifeKeeper to move the server's default route to a backup interface, it is recommended that you choose option 1 above, configuring the LifeKeeper-protected address on its own logical subnet, distinct from the subnets of all non-LifeKeeper addresses on the server.

Note that all of these route management issues can be avoided by using the Linux NIC bonding mechanism to achieve network interface redundancy, rather than the IP Recovery Kit's backup interface mechanism, because the shifting of network traffic from one network adapter to another is handled below the level of the bonded network interface upon which the IP addresses and routes are configured.

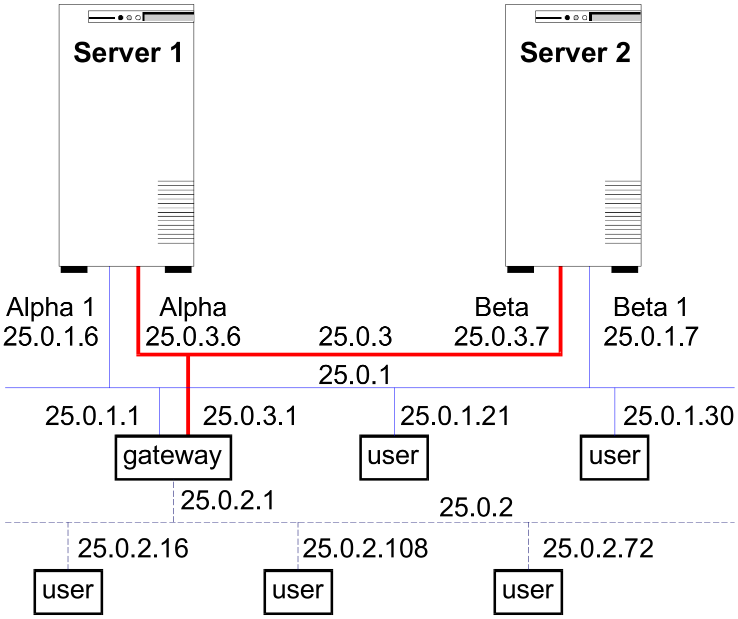
Configuration Examples

This section identifies example network configurations and then describes three sample IP configuration exercises. The first example illustrates a typical case of a database application dependent upon a single IP resource and configured on a pre-existing subnet. The second example illustrates an active/active scenario where multiple IP resources are configured. The third example illustrates a typical LifeKeeper configuration with the IP Local Recovery feature enabled.

Network Configuration

The first two configuration examples assume the network configuration diagrammed in Figure 2.

Figure 2. Network Configuration



The network configuration has these components:

- **Servers.** The configuration has two servers, Server 1 and Server 2, each with the appropriate LifeKeeper and application software installed.
- **Interfaces.** Each server has two Ethernet interfaces, *eth0* and *eth1*, configured as follows:

Interface	Server 1	Server 2
eth0	Server1	Server2
	25.0.3.6	25.0.3.7
eth1	Server11	Server21
	25.0.1.6	25.0.1.7

- **Network.** The network consists of three subnetworks:
  - Low traffic backbone (25.0.3) primarily for servers
  - High traffic backbone (25.0.1) with both servers and clients
  - High traffic client network (25.0.2.)

  A gateway provides interconnection routing between all LANs. A Domain Name Server (not shown) is used for address resolution.
- **Heartbeat.** TCP heartbeat communication paths would be configured using either or both of the server subnetworks.

Typical Configuration Example

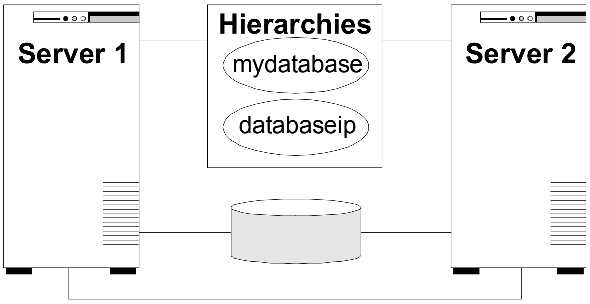
Server 1 and Server 2 have access to an application called *mydatabase* that resides on a shared disk. To ensure that the application *mydatabase* and the IP resources used to access it are switched together, the system administrator creates a *mydatabase* application resource and adds the IP resource to the application hierarchy as a dependency.

These are the configuration issues:

- **Application hierarchy.** The application hierarchy must exist before the administrator names it as a parent of the IP resource. For the purposes of this example, Server 1 is the primary server. The application resource tags are *mydatabase-on-server1* and *mydatabase-on-server2*.
- **IP resource name.** The administrator adds the name and address of the IP resource to the */etc/hosts* file on both Server 1 and Server 2 and to the DNS database. In this example, the IP resource name is *databaseip* and its network address is 25.0.1.2. If no name-to-IP address association is necessary, then this is not required.
- **Routers, gateways, and users.** Because *databaseip* is an address on an existing subnet, no additional configuration is necessary. The IP resource is on the 25.0.1 subnet. All users connect to *databaseip* via the route they currently use to get to the 25.0.1 subnet. For example, users on 25.0.2 go through the gateway and users on 25.0.1 connect directly.
- **IP instance definition.** When the administrator enters *databaseip* as the IP resource on the Resource Hierarchy Create screen, the software performs several tests. It verifies that Server 1 can determine the address that goes with *databaseip* (it is in the hosts file and/or can be retrieved from the DNS). It also verifies that the address retrieved, address 25.0.1.2, is not already in use. Since the IP resource is on the 25.0.1 subnet, the IP Recovery software will ensure that it is configured on the *eth1* interface. If the IP resource is acceptable, the software fills in the remainder of the wizard dialog boxes with default values, as shown in the table below Figure 3. If you selected all the default values, an independent IP resource hierarchy called *ip-databaseip* would be created.

Note: The tables associated with each configuration illustration provide examples of the appropriate information that would be entered in the Create Resource Hierarchy wizard for the primary server (Server 1) and Extend Resource Hierarchy wizard for the backup server (Server 2). For additional details on what information should be entered into the wizards, refer to the *LifeKeeper Configuration Tasks* section later in this guide. These tables can be a helpful reference when configuring your recovery kit.

Figure 3. Typical Configuration Example of IP Resource Creation



Configuration Notes:

1. The application resource is *mydatabase-on-server1*.
2. The IP resource is *databaseip* with a tag name of *ip-databaseip.*
3. If *mydatabase-on-server1* fails, LifeKeeper switches it to Server 2; (*ip-databaseip* is only switched if a dependency exists).
4. If Server 1 fails, both resources are brought in-service on Server 2.
5. During a switchover, *databaseip* users would be disconnected. When they log back in, they can access any applications on Server 2.
6. During a manual switchover, users connected to Server 1 via connections other than *databaseip* remain connected to Server 1.

Creating an IP resource hierarchy on Server 1:

Server:	Server1
IP Resource:	databaseip
Netmask:	255.255.252.0
Network Interface:	eth1
Backup Interface:	none
IP Resource Tag:	ip-databaseip

Note: See the section on *Guidelines for Creating an IP Dependency* later in this guide before extending an IP resource to a backup server.

Extending an IP resource hierarchy to Server 2:

Template Server:	Server1
Tag to Extend:	databaseip
Target Server:	Server2
Target Priority:	10

** IP Resource:	25.0.1.2
Netmask:	255.255.252.0
Network Interface:	eth1
Backup Interface:	none
IP Resource Tag:	ip-databaseip

** Note that the actual IP address associated with the DNS name is displayed in the Extend Wizard as the IP resource.

Test Your IP Resource

To verify the successful creation of the IP resource, the administrator should perform the following tasks:

1. From the LifeKeeper GUI, observe whether *ip-databaseip* is in-service (ISP) on Server 1.
2. From a remote server, connect to address *databaseip* using *ping* or *telnet*.
3. Test manual switchover by selecting the *in_service* option on Server 2 and selecting *ip-databaseip*. Verify that the IP address migrates to Server 2.

Active/Active Configuration Example

The second example, using the same network configuration, describes two IP resources, one active on each server.

Resource Addresses

For this example, the IP resources are *server1ip* (address 25.0.6.20) and *server2ip* (address 25.0.6.21). Entries for these resources must be in the */etc/hosts* files on each server and in the DNS database.

Router Configuration

Because the selected addresses are on a new (logical) subnet, they can be configured for either *eth0* or *eth1*. However, both must go on the same interface.

For this example, choosing *eth0* means that all users would have to go through the gateway. Choosing *eth1* would allow the users on the 25.0.1 subnet to access the resources directly (assuming that the new subnet had been added to their internal routing tables). Users on subnet 25.0.2 would still require the gateway. For the purposes of this example, the selected interface is *eth1*.

Regardless of which physical network is chosen to support the new subnet, the network administrator would have to add routing information to the gateway system before creating the IP resources.

First IP Resource Definition

The administrator creates the first IP resource on Server 1. *eth0* is the first available interface on each server and would appear as the default. To define *eth1* as the interface, the administrator selects it from the list of available interfaces.

Creating an IP resource hierarchy on Server 1:

Server:	Server1

IP Resource:	server1ip
Netmask:	255.255.252.0
Network Interface:	eth1
Backup Interface:	none
IP Resource Tag:	ip-server1ip

Note: See the section on *Guidelines for Creating an IP Dependency* later in this section before extending an IP resource to a backup server.

Extending an IP resource hierarchy to Server 2:

Template Server:	Server1
Tag to Extend:	server1ip
Target Server:	Server2
Target Priority:	10
** IP Resource:	25.0.6.20
Netmask:	255.255.252.0
Network Interface:	eth1
Backup Interface:	none
IP Resource Tag:	ip-server1ip

** Note that the actual IP address associated with the DNS name is displayed in the Extend Wizard as the IP resource.

Second IP resource definition

The administrator creates the second IP resource on Server 2. *eth0* is the first available interface on each server and would appear as the default. To define *eth1* as the interface, the administrator selects it from the list of available interfaces.

Creating an IP resource hierarchy on Server 2:

Server:	Server2
IP Resource:	server2ip
Netmask:	255.255.252.0
Network Interface:	eth1
Backup Interface:	none
IP Resource Tag:	ip-server2ip

Note: See the section on *Guidelines for Creating an IP Dependency* later in this guide before extending an IP resource to a backup server.

Extending an IP resource hierarchy to Server 1:

Template Server:	Server2

Tag to Extend:	server2ip
Target Server:	Server1
Target Priority:	10
** IP Resource:	25.0.6.21
Netmask:	255.255.252.0
Network Interface:	eth1
Backup Interface:	none
IP Resource Tag:	ip-server2ip

** Note that the actual IP address associated with the DNS name is displayed in the Extend Wizard as the IP resource.

Note: Since subnet 25.0.6 is not active on Server 2, both *eth0* and *eth1* are available choices for the Primary network interface. On Server 1 (the backup server), the only choice is *eth1* because the first IP resource, 25.0.6.20, is in service there. When the administrator saves the definition, LifeKeeper brings address 25.0.6.21 in-service on *eth1* on Server 2.

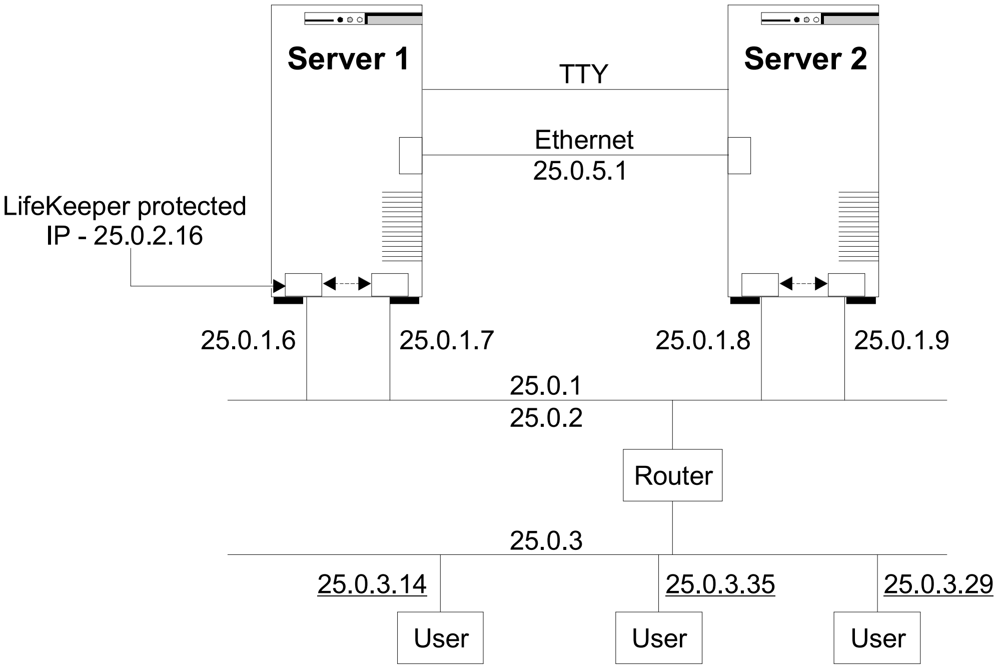
Testing IP resources

The administrator should verify that the new resources are functioning on both servers by performing the following tests:

1. With each resource on its primary server, verify that each is accessible by using either *ping* or *telnet*. The administrator may also want to test connectivity from all user sites.
2. Test switchover by manually bringing *ip-server1ip* into service on Server 2. Verify both resources are functional on Server 2.
3. Bring both resources into service on Server 1. Verify both resources are functional on Server 1.
4. Bring *ip-server2ip* back into service on its primary server, Server 2.

Local IP Recovery Configuration Example

The figure below illustrates a typical LifeKeeper cluster environment with the Local IP Recovery feature. Each server has both a primary and a backup network interface card (NIC), along with another dedicated NIC for the Ethernet/TCP heartbeat between Server 1 and Server 2.



This example follows option 1 in the configuration guidelines described in the *Local Recovery Configuration Restrictions* section earlier in this document. The interface configurations are listed below:

eth0:	25.0.1.6 Physical network #1 (public network) Logical subnet 25.0.1 (Netmask 255.255.255.0) LifeKeeper communication path, priority 2
eth1:	25.0.5.1 Physical network #2 (private network) Logical subnet 25.0.5 (Netmask 255.255.255.0) LifeKeeper communication path, priority 1
eth2:	25.0.1.7 Physical network #1 (public network) Logical subnet 25.0.1 (Netmask 255.255.255.0) Cannot be used for LifeKeeper communication path (same network as eth0)
LK IP1:	25.0.2.16 Physical network #1 (public network) Logical subnet 25.0.2 (Netmask 255.255.255.0) Primary interface eth0, backup interface eth2

In this example, *eth2* is basically unused until it becomes necessary to move a LifeKeeper-protected address to its backup interface. If the configuration had more than one LifeKeeper-protected address sharing the same distinct subnet, they would move together, and no other addresses on the system would be impacted.

Guidelines for Creating an IP Dependency

How and/or when you are going to create a parent/child dependency between a LifeKeeper-protected application and a LifeKeeper-protected IP address is typically dependent on the LifeKeeper-protected application. For example, in a LifeKeeper-protected Apache environment, the parent/child dependency is created during the creation of the Apache resource hierarchy (assuming you have already created a protected IP address). In other applications that do not create this dependency automatically, it is recommended you use the following steps:

1. Create the application/parent resource hierarchy. **Note:** Do not extend the resource hierarchy to a backup server at this time. You will receive a warning message when you elect not to extend your hierarchy, but in this particular situation, it is the proper action to take.
2. Create the IP resource hierarchy. **Note:** Do not extend the IP resource hierarchy to a backup server at this time. You will receive a warning message when you elect not to extend your hierarchy, but in this particular situation, it is the proper action to take.
3. Create the parent/child dependency between the parent application resource hierarchy and the IP resource hierarchy using the Create Resource Dependency configuration task (see the *LifeKeeper Online Product Manual* topic, *Creating a Resource Dependency*, under the GUI Administration Tasks book.
4. Finally, extend the application resource hierarchy to the backup server. Since the dependency has already been created, the dependent IP resource instance will also be extended to the backup server as part of the parent application resource hierarchy.

The steps outlined above save you from performing one extra extension (i.e. the extension of the IP resource to the backup server).

Command Line Operations for Local Recovery

The **lkipbu** command provides a mechanism for manipulating backup interfaces for existing IP resource instances.

The **add** operation (specified via the ***-a*** option) is used to add a backup interface (*interface*) to the IP resource instance (*tag*). This command will fail if a backup interface has already been defined for this instance, or if an invalid interface name is provided. The command and syntax are:

lkipbu [-d *machine*] -a -t *tag* -f *interface*

The **remove** operation (specified via the ***-r*** option) is used to remove a backup interface from the IP resource instance. This command will fail if the specified interface is not the current backup interface for this instance. The command and syntax are:

lkipbu [-d *machine*] -r -t *tag* -f *interface*

The **lkipbu** command can also manually **move** an IP address to its backup interface. This capability is specified via the **-m** option, using the following syntax:

lkipbu [-d *machine*] -m -t *tag*

This operation will fail if there is no backup interface configured for this instance. If the IP address is not currently active, this command simply swaps the primary and backup interface values for the IP resource instance *tag*.

To ensure proper subnet routing after an address is moved, any active IP addresses on the same logical subnet as the moved IP address will be unconfigured by the **lkipbu -m** command, unless they are already active on the backup interface. This action applies to both the primary addresses of an interface, as well as the alias addresses, including other LifeKeeper-protected addresses.

The **lkipbu** utility provides an option for **retrieving** the currently defined primary and backup interfaces for a specified IP resource instance, along with the state of the resource on the primary interface (up or down). This capability is specified via the **-q** option, using the following syntax:

> lkipbu [-d *machine*] -q -t *tag*

For example:

```
# lkipbu -q -t ip-25.0.2.16
IP address: 25.0.2.16
Netmask: 255.255.255.0
Primary interface: eth0 (up)
Backup interface: eth2
```

Configuring more than one protected virtual IP address on a single backup network interface is not supported. Use the Linux NIC bonding mechanism for these configurations.

Refer to the **lkipbu**(8) man page in the *LifeKeeper Online Product Manual* for further detail.

LifeKeeper Configuration Tasks

The following configuration tasks for virtual IP address resources are described in this guide, as they are unique to an IP resource instance, and different for each recovery kit.

- **Creating an IP Resource Hierarchy.** Creates an application resource hierarchy in your LifeKeeper cluster.
- **Deleting a Resource Hierarchy.** Deletes a resource hierarchy from all servers in your LifeKeeper cluster.
- **Extending Your Hierarchy.** Extends a resource hierarchy from the primary server to a backup server.
- **Unextending Your Hierarchy.** Unextends (removes) a resource hierarchy from a single server in the LifeKeeper cluster.
- **Testing Your Resource Hierarchy.** Tests a virtual IP resource hierarchy for proper configuration and operation.
- **Viewing/Editing IP Configuration Properties.** Displays configuration details for an IP resource and allows some of them to be modified.
- **Adjusting IP Recovery Kit Tunable Values.** Tunes characteristics of the overall behavior of the IP Recovery Kit.

The following tasks are described in the GUI Administration section within the *LifeKeeper Online Product Manual*, because they are common tasks with steps that are identical across all recovery kits.

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

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

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

Creating an IP Resource Hierarchy

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

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

2. A dialog box will appear with a drop down list box menu listing all recognized recovery kits installed within the cluster. Select **IP** from the drop down list and click **Next**.

3. You will be prompted to enter the following information. When the **Back** button is active in any of the dialog boxes, you can go back to the previous dialog box. This is especially helpful should you encounter an error that might require you to correct previously entered information.

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

Field	**Tips**
Switchback Type	This dictates how the IP instance will be switched back to this server when the server comes back up after a failover. You can choose either *intelligent* or *automatic.* Intelligent switchback requires administrative intervention to switch the instance back to the primary/original server. Automatic switchback means the switchback will occur as soon as the primary server comes back on line and reestablishes LifeKeeper communication paths. The switchback type can be changed later from the General tab of the Resource Properties dialog box.
Server	Select the **Server** where you want to place the IP Address (typically this is referred to as the primary or template server). All the servers in your cluster are included in the drop down list.
IP Resource	Select or enter the actual **IP Resource.** This is the IP address or symbolic name that LifeKeeper will use for this resource. This is used by client applications to login to the parent application over a specific network interface. If you use a symbolic name, it must exist in the local */etc/hosts* file or be accessible via a Domain Name Service (DNS). Alias names and domain names are acceptable as long as they meet the criteria listed above. No defaults are provided for this information field. **Note:** If you choose to use a symbolic name, be advised that when you extend this resource, the actual IP address will appear in one of the dialog boxes as the IP resource designation.

Field	Tips
Netmask	Select or enter the network mask, **Netmask**, which your IP resource will use on the target server. Any standard netmask for the class of the specific IP resource address is valid. **Note:** The netmask you choose, combined with the IP address, determines the subnet that will be used by the IP resource and should be consistent with the network configuration.
Network Interface	Select or enter the **Network Interface** where your IP resource will be placed under LifeKeeper protection. This is the physical Ethernet card that the IP address is interfacing with. Valid choices will depend on the existing network configuration and values chosen for the IP resource address and netmask. The default value is the interface within the set of valid choices which most closely matches the address and netmask values you have selected.
Backup Interface	Select a **Backup Interface** if you want to engage the IP Local Recovery feature on this server. The default value is *none*; however, if you have another network interface card configured on this server, it should be listed in the drop down list box.
IP Resource Tag	Select or enter a unique **IP Resource Tag** name for the IP resource instance you are creating. This field is populated automatically with a default tag name, ip-<resource>, where <resource> is the resource name or IP address. You can change this tag if you want to.

4. Click **Create**. The Create Resource Wizard will then create your IP resource.

5. At this point, an information box appears and LifeKeeper will validate that you have provided valid data to create your IP resource hierarchy. If LifeKeeper detects a problem, an ERROR will appear in the information box. If the validation is successful, your resource will be created. Click **Next.**

6. Another information box will appear explaining that you have successfully created an IP resource hierarchy, and you must **Extend** that hierarchy to another server in your cluster in order to place it under LifeKeeper protection.

   When you click **Continue**, LifeKeeper will launch the *Pre-Extend* configuration task. Refer to the *Extending Your Hierarchy* section for details on how to extend your resource hierarchy to another server.

   If you click **Cancel** now, another dialog box will appear alerting you that you'll need to come back and extend your IP resource hierarchy to another server at some other time to put it under LifeKeeper protection.

Deleting a Resource Hierarchy

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

1. From the LifeKeeper GUI menu, select **Edit**, then **Resource**. From the drop down menu, select **Delete Resource Hierarchy**.
2. Select the name of the **Target Server** where you will be deleting your IP resource hierarchy and then click **Next**. (This dialog will not appear if you selected the Delete Resource task by right clicking on a resource instance in either pane.)
3. Select the **Hierarchy to Delete**. Identify the resource hierarchy you wish to delete, and highlight it and then click **Next**. (This dialog will not appear if you selected the Delete Resource task by right clicking on a resource instance in the left or right pane.)
4. An information box appears confirming your selection of the target server and the hierarchy you have selected to delete. Click **Delete** to proceed with resource deletion.
5. Another information box appears confirming that the IP resource was deleted successfully.
6. Click **Done** to exit out of the Delete Resource Hierarchy menu selection.

Extending Your Hierarchy

Note: See the section on *Guidelines for Creating an IP Dependency* earlier in this guide before extending an IP resource to a backup server.

After you have created a hierarchy, you will want to extend that hierarchy to another server in the cluster. There are three possible scenarios to extend your resource instance from the template server to a target server. The first scenario is when you “Continue” from creating the resource into extending that resource to another server. The second scenario is when you enter the Extend Resource Hierarchy task from the edit menu as shown below. The third scenario is when you right click on an unextended hierarchy in either the left or right hand pane. Each scenario takes you through the same dialog boxes (with a few exceptions, which are clearly detailed below).

1. If you are entering the Extend wizard from the LifeKeeper GUI menu, select **Edit**, then **Resource**. From the drop down menu, select **Extend Resource Hierarchy**. This will launch the Extend Resource Hierarchy wizard. If you are unfamiliar with the Extend operation, click **Next**. If you are familiar with the LifeKeeper **Extend Resource Hierarchy** defaults and want to bypass the prompts for input/confirmation, click **Accept Defaults**.
2. The *Pre-Extend Wizard* will prompt you to enter the following information. **Note:** The first two fields appear only if you initiated the Extend from the **Edit** menu. It should be noted that if you click **Cancel** at any time during the sequence of extending your hierarchy, LifeKeeper will cancel the extension process to that particular server. However, if you have already extended the resource to another server, that instance will continue to be in effect until you specifically unextend it.

Field	**Tips**
Template Server	Enter the server where your IP resource is currently in service.
Tag to Extend	Select the IP resource you wish to extend. This is the name of the IP instance you wish to extend from the template server to the target server. The wizard will list in the drop down list box all the resources that you have created on the template server that you selected in the previous dialog box.
Target Server	Select the **Target Server** where you are extending your IP resource hierarchy. The drop down box provides the names of the servers in your cluster that are not already in the selected hierarchy.
Switchback Type	Select the **Switchback Type.** This dictates how the IP instance will be switched back to this server when it comes back into service after a failover to the backup server. You can choose either *intelligent* or *automatic*. Intelligent switchback requires administrative intervention to switchback the instance to the primary/original server. Automatic switchback means the switchback will occur as soon as the primary server comes back on line and reestablishes LifeKeeper communication paths. The switchback type can be changed later, if desired, from the General tab of the Resource Properties dialog box.
Template Priority	Select or enter a **Template Priority**. This is the priority for the IP hierarchy on the server where it is currently in service. Any unused priority value from 1 to 999 is valid, where a lower number means a higher priority (1=highest). The extend process will reject any priority for this hierarchy that is already in use by another system. The default value is recommended. **Note:** This selection will appear only for the initial extend of the hierarchy.
Target Priority	Select or enter the **Target Priority**. This is the priority for the new extended IP hierarchy relative to equivalent hierarchies on other servers. Any unused priority value from 1 to 999 is valid, indicating a server’s priority in the cascading failover sequence for the resource. A lower number means a higher priority (1=highest). Note that LifeKeeper assigns the number “1” to the server on which the hierarchy is created by default. The priorities need not be consecutive, but no two servers can have the same priority for a given resource.

3. An information box will appear explaining that LifeKeeper has successfully checked your environment and that all the requirements for extending this IP resource have been met. If there were some requirements that had not been met, LifeKeeper would not allow you to select the **Next** button, and the **Back** button would be enabled. If you click **Back**, you can make changes to your resource extension according to any error messages that may appear in the information box. If you click **Cancel** now, you will need to come back and extend your IP resource hierarchy to another server at some other time to put it under LifeKeeper protection. When you click **Next**, LifeKeeper will launch you into the Extend Resource Hierarchy configuration task.

4. The Extend Resource Hierarchy configuration task will prompt you to enter the following information.

Field	**Tips**
IP Resource	This is the same **IP Resource** or address used in the Create Resource Wizard. This dialog box is for information purposes only. You cannot change the **IP Resource** that appears in the box.
Netmask	This is the same **Netmask** that was selected when the IP resource was created for the template server and will now be used by the IP resource for the target server. This dialog box is for information purposes only. You cannot change the **Netmask** that appears in the box.
Network Interface	Select or enter the **Network Interface.** This is the name of the network interface (i.e. Ethernet card) the IP resource will use on the target server.
Backup Interface	Select a **Backup Interface** if you want to engage the IP Local Recovery feature on the server that you are extending the IP resource. The default value is *none*; however, if you have another network interface card configured on this server, it should be listed in the drop down list.
IP Resource Tag	Select or enter the **IP Resource Tag.** This is the resource tag name to be used by the IP resource being extended to the target server.

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

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

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

6. Click **Done** to exit from the Extend Resources Hierarchy menu selection.

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

Unextending Your Hierarchy

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

2. Select the **Target Server** where you want to unextend the IP resource. It cannot be the server where the IP address is currently in service.

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

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

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

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

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

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

   Click **Unextend**.

5. Another information box appears confirming that the IP resource was unextended successfully.
6. Click **Done** to exit out of the Unextend Resource Hierarchy menu.

Testing Your Resource Hierarchy

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

Performing a Manual Switchover from the GUI

You can initiate a manual switchover from the LifeKeeper GUI by selecting **Edit**, then **Resource**, then finally **In Service** from the drop down menu. For example, an *in service* request executed on a backup server causes the application hierarchy to be placed in service on the backup server and taken out of service on the primary server. At this point, the original backup server is now the primary server and original primary server has now become the backup server.

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

In a manual switchover, the IP Recovery Kit removes the address from service on the active server before adding it to the backup server.

After switchover, the IP resource has a different hardware (MAC) address because it is associated with a different LAN interface. Before user systems can reconnect, the user systems' TCP/IP software must determine this new address mapping. The IP Recovery Kit automatically informs all connected servers that they must update their ARP (Address Resolution Protocol) tables to reflect the new mapping.

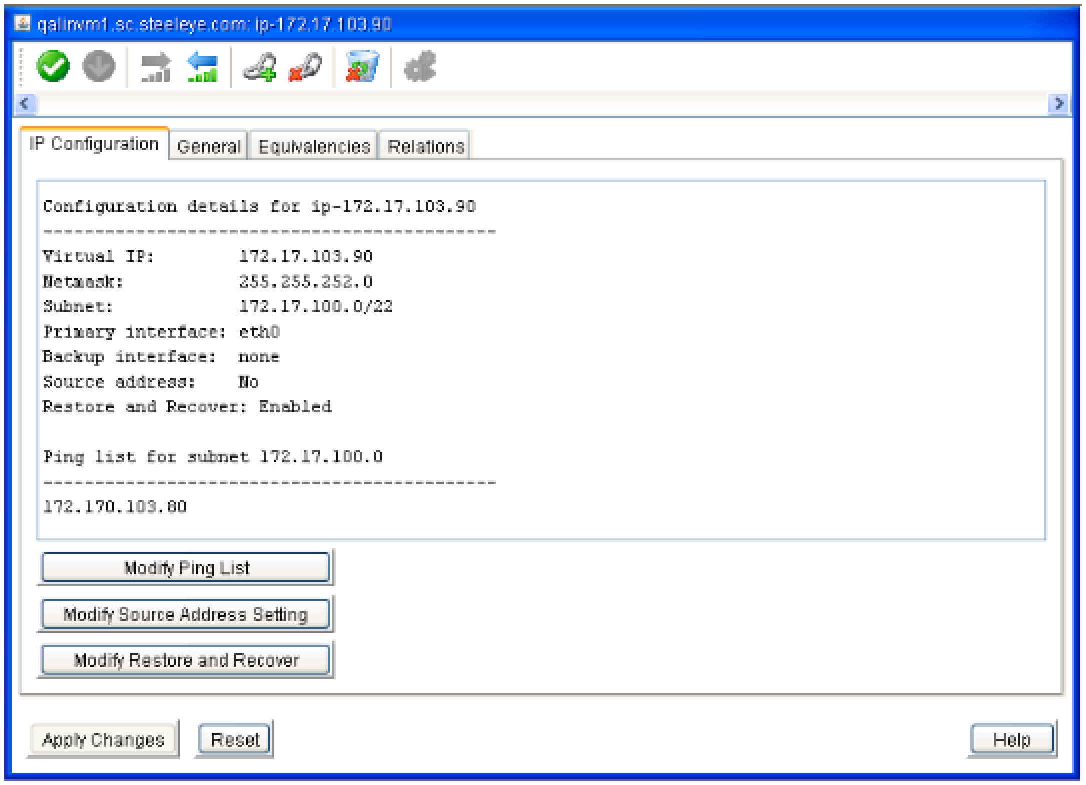
User systems running full TCP/IP implementations are updated immediately. User systems with less sophisticated implementations may have delayed update or may require routers as addressing intermediaries.

Viewing/Editing IP Configuration Properties

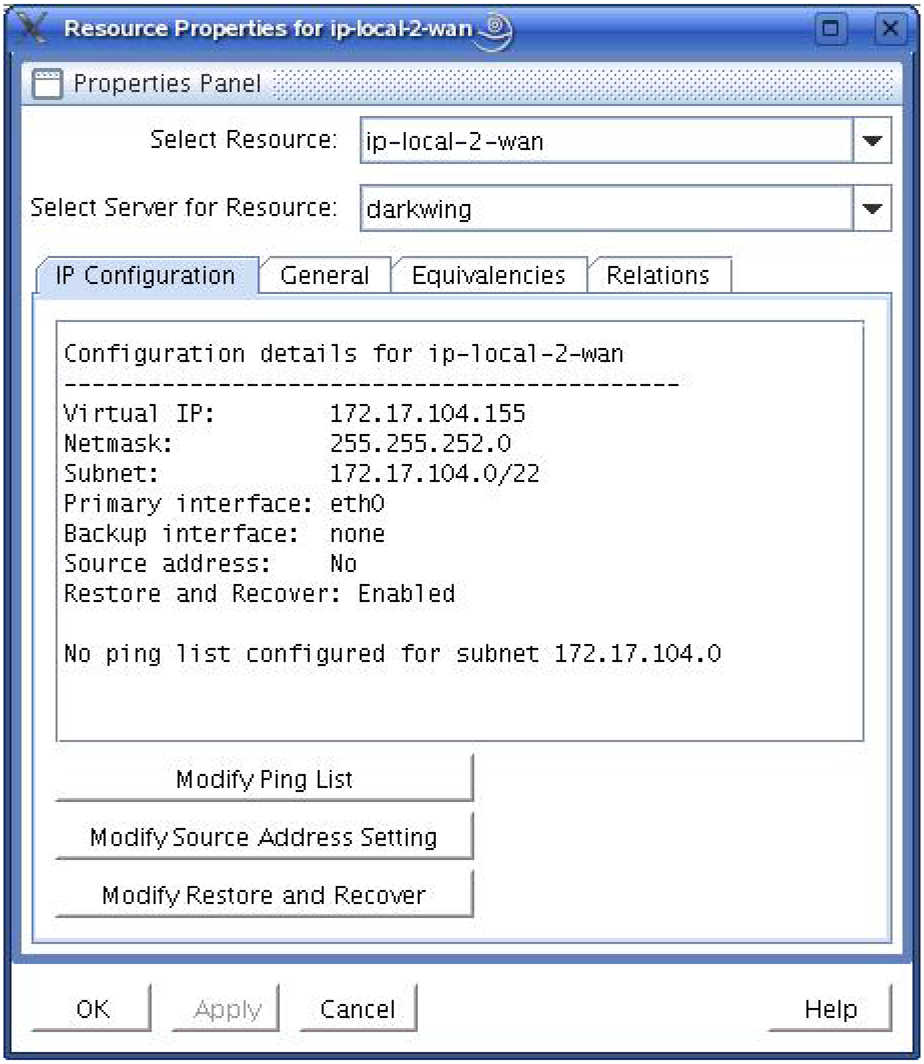
The IP Configuration Properties page allows you to view the configuration details for a specific IP resource, as well as to modify a number of selected configuration items.

To access the IP Configuration Properties page, from the LifeKeeper GUI menu select **Edit**, then **Resource**. From the drop down menu, select **Properties**. Then select the resource for which you want to view properties from the **Resource** list and the server for which you want to view that resource from the **Server** list. You can also access the properties page using the context-sensitive menu that appears when you right-click on a specific IP resource instance.

Below is an example of the properties page that will appear for an IP resource.


qalinvm1.sc.steeleye.com: ip-172.17.103.90
IP Configuration
General
Equivalencies
Relations
Modify Ping List
Modify Source Address Setting
Modify Restore and Recover
Apply Changes
Reset
Help

```
Configuration details for ip-172.17.103.90
--------------------------------------------
Virtual IP:        172.17.103.90
Netmask:           255.255.252.0
Subnet:            172.17.100.0/22
Primary interface: eth0
Backup interface:  none
Source address:    No
Restore and Recover: Enabled

Ping list for subnet 172.17.100.0
--------------------------------------------
172.170.103.80
```


Resource Properties for ip-local-2-wan
Properties Panel
Select Resource: ip-local-2-wan
Select Server for Resource: darkwing
IP Configuration
General
Equivalencies
Relations
Modify Ping List
Modify Source Address Setting
Modify Restore and Recover
OK
Apply
Cancel
Help

```
Configuration details for ip-local-2-wan
--------------------------------------------
Virtual IP:        172.17.104.155
Netmask:           255.255.252.0
Subnet:            172.17.104.0/22
Primary interface: eth0
Backup interface:  none
Source address:    No
Restore and Recover: Enabled

No ping list configured for subnet 172.17.104.0
```

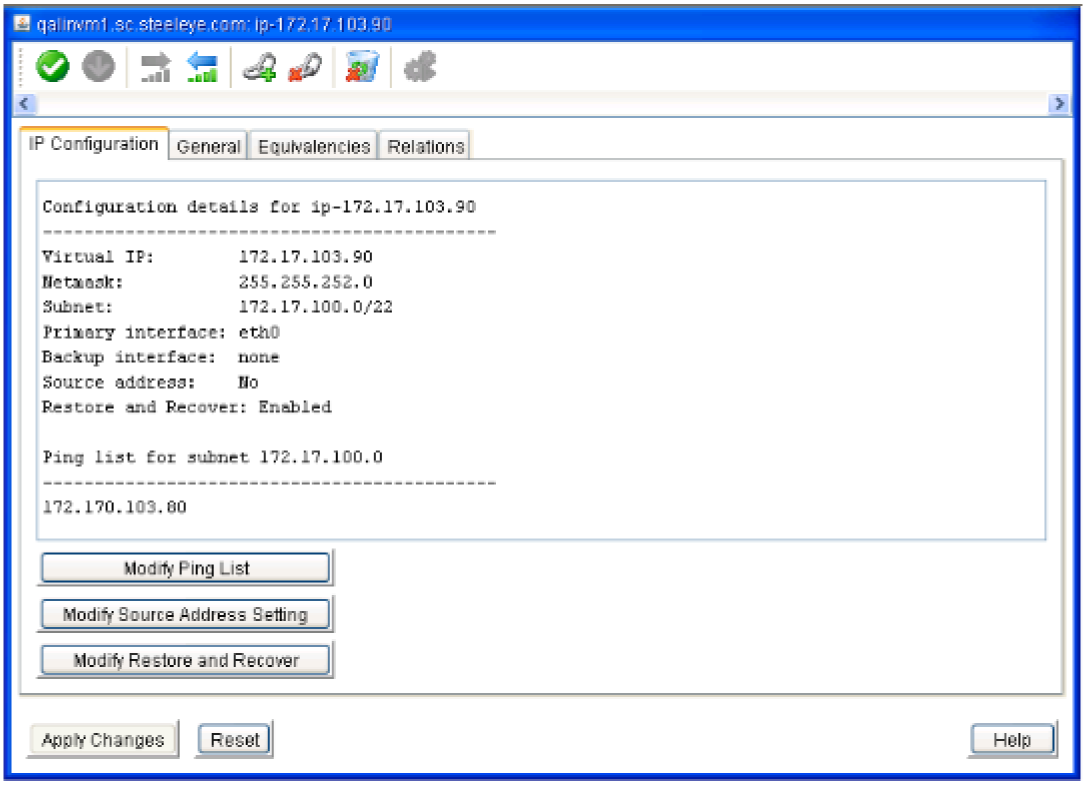
The resulting properties page contains four tabs. The first of those tabs, labeled **IP Configuration**, contains configuration information that is specific to IP resources. The remaining three tabs are available for all LifeKeeper resource types, and their content is described in the *LifeKeeper Online Product Manual*.

The **IP Configuration** tab displays the following information elements about the selected IP resource.

Virtual IP	The virtual IP address associated with this IP resource.
Netmask	The netmask for the virtual IP address. This value determines how much of the address makes up the subnet portion.
Subnet	The logical subnet address for the virtual IP address, including the number of bits included in the subnet portion of the address.
Primary interface	The network interface on which the virtual IP address should be configured when it is active.
Backup interface	The backup network interface for the IP resource, if any, for the purposes of IP local recovery.

Source address setting	Specifies whether the virtual IP address should be configured as the source address for outbound IP traffic onto its associated subnet.
Ping List	The optional list of IP addresses to be pinged during IP health checks for this IP resource (and others on the same subnet), as an alternative to the normal broadcast ping mechanism.

In the example above, there is no *Ping List* configured for this IP resource. When a *Ping List* is configured, the resulting properties page looks like the following example.


qalinwm1.sc.steeleye.com: ip-172.17.103.90
IP Configuration
General
Equivalencies
Relations
Modify Ping List
Modify Source Address Setting
Modify Restore and Recover
Apply Changes
Reset
Help

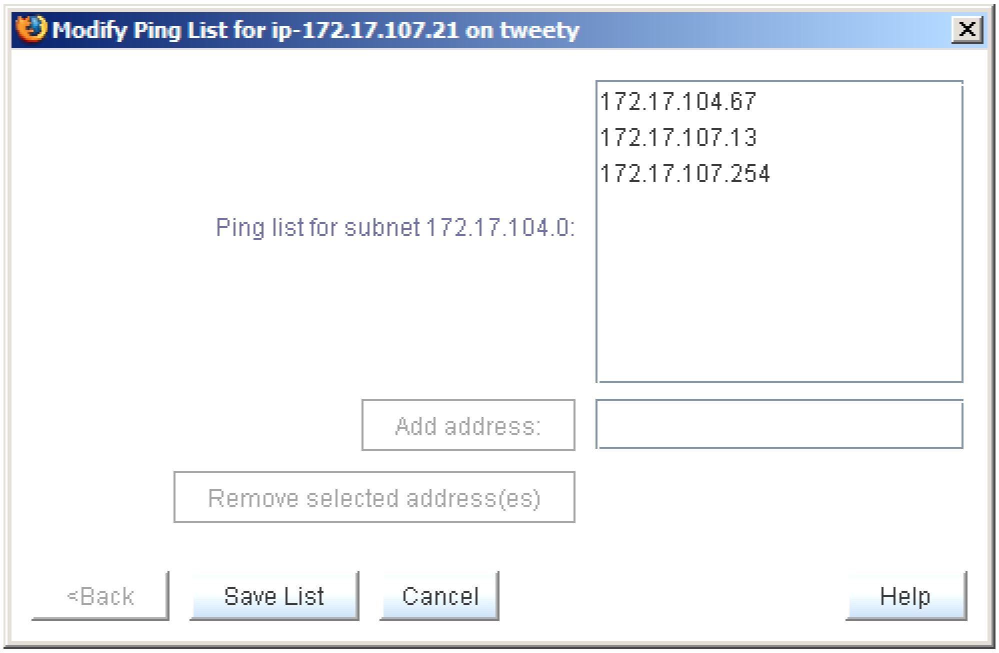
```
Configuration details for ip-172.17.103.90
--------------------------------------------
Virtual IP:        172.17.103.90
Netmask:           255.255.252.0
Subnet:            172.17.100.0/22
Primary interface: eth0
Backup interface:  none
Source address:    No
Restore and Recover: Enabled

Ping list for subnet 172.17.100.0
--------------------------------------------
172.170.103.80
```

The **Modify Ping List** and **Modify Source Address Setting** buttons can be used to perform modifications to those configuration items, as described in the sections below.

Modifying the Ping List

For a description of the use and function of the *Ping List* for an IP resource, see the section *IP Resource Monitoring* earlier in this document.

To create a *Ping List* for an IP resource, or to modify an existing list, click the **Modify Ping List** button on the **IP Configuration** properties page. This brings up a dialog window similar to the following example.

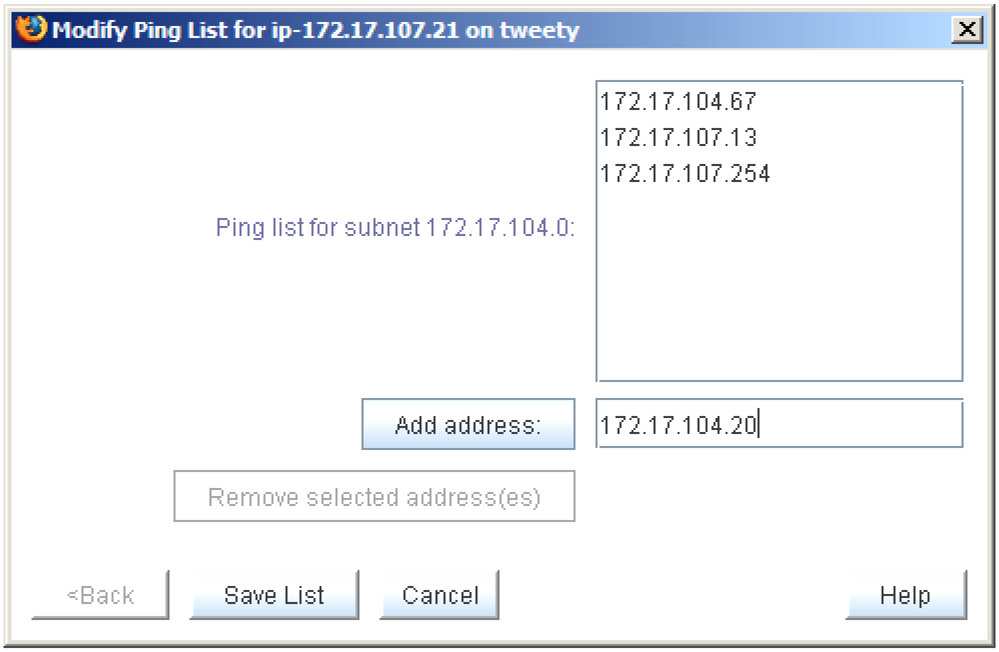


Modify Ping List for ip-172.17.107.21 on tweety

Ping list for subnet 172.17.104.0:

```
172.17.104.67
172.17.107.13
172.17.107.254
```

To add an address to the *Ping List*, type the address in the field next to the **Add address:** button, and push the button, as shown in the following two images. Note that the **Add address:** button is grayed out until you begin typing an address in the field.

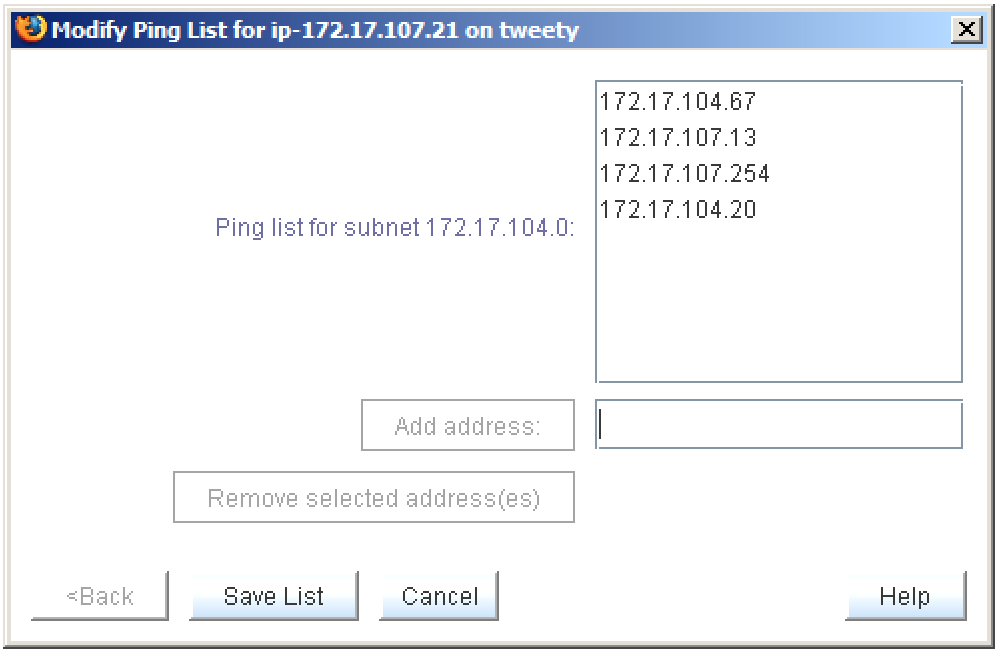


Modify Ping List for ip-172.17.107.21 on tweety

Ping list for subnet 172.17.104.0:

```
172.17.104.67
172.17.107.13
172.17.107.254
```

Add address: 172.17.104.20

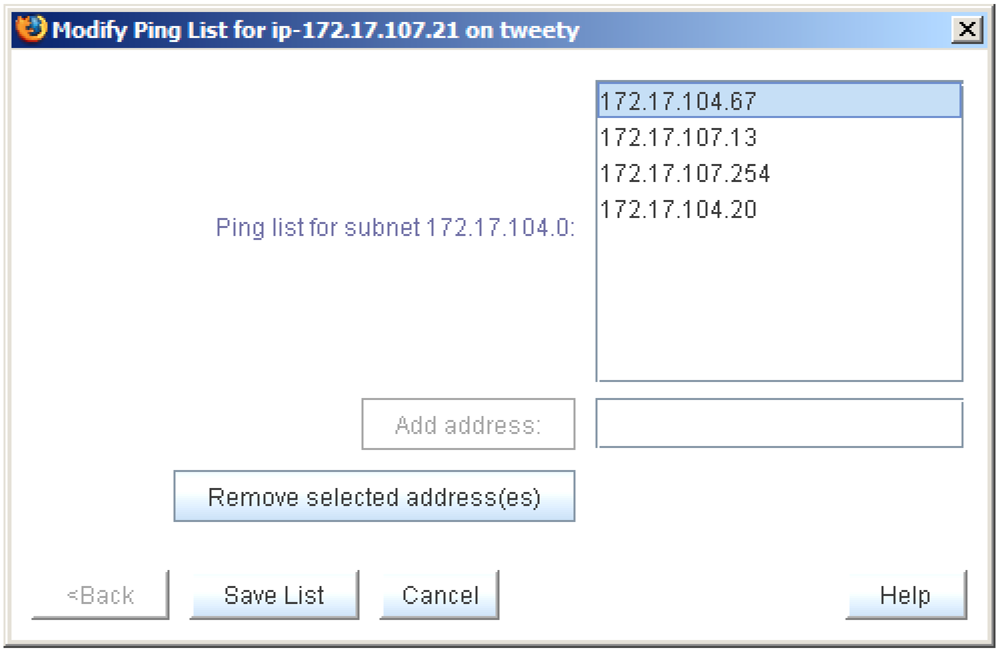


Modify Ping List for ip-172.17.107.21 on tweety

Ping list for subnet 172.17.104.0:

172.17.104.67
172.17.107.13
172.17.107.254
172.17.104.20

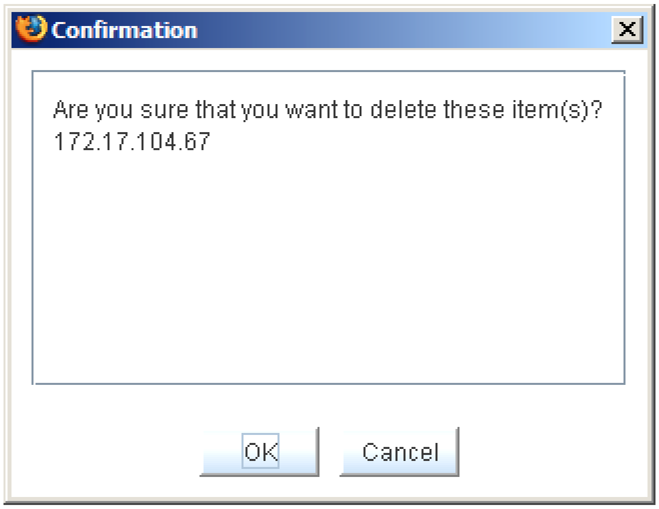
To remove one or more addresses from the *Ping List*, click to select the addresses to be removed and click the **Remove selected address(es)** button. The **Remove selected address(es)** button is also grayed out until at least one address in the list has been selected.



Modify Ping List for ip-172.17.107.21 on tweety

Ping list for subnet 172.17.104.0:

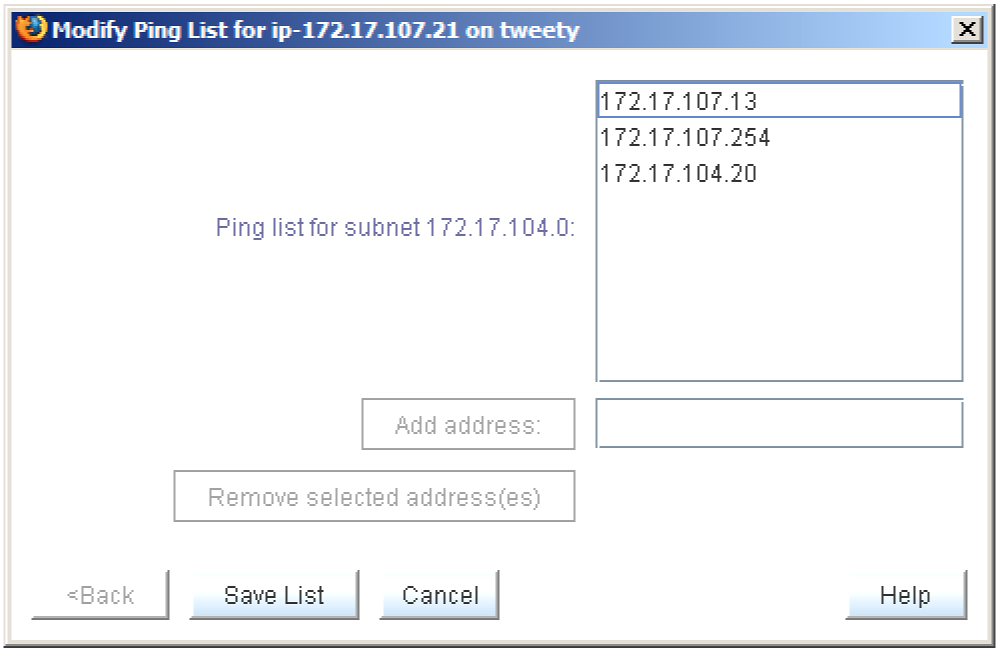
172.17.104.67
172.17.107.13
172.17.107.254
172.17.104.20



Confirmation

Are you sure that you want to delete these item(s)?
172.17.104.67

Click **OK** to confirm that you want to remove the indicated addresses.

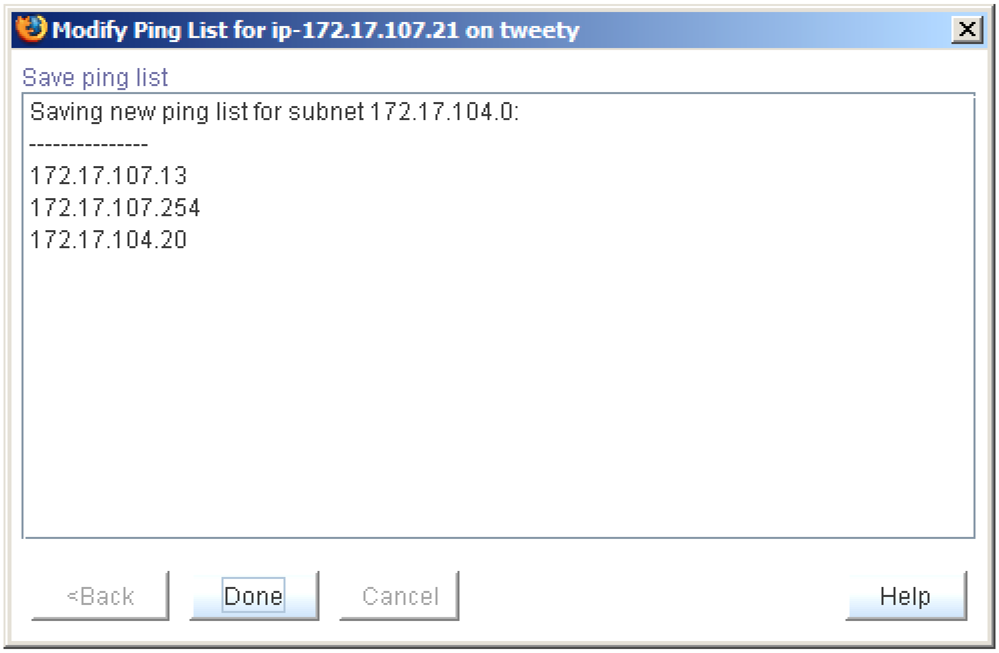


Modify Ping List for ip-172.17.107.21 on tweety

Ping list for subnet 172.17.104.0:

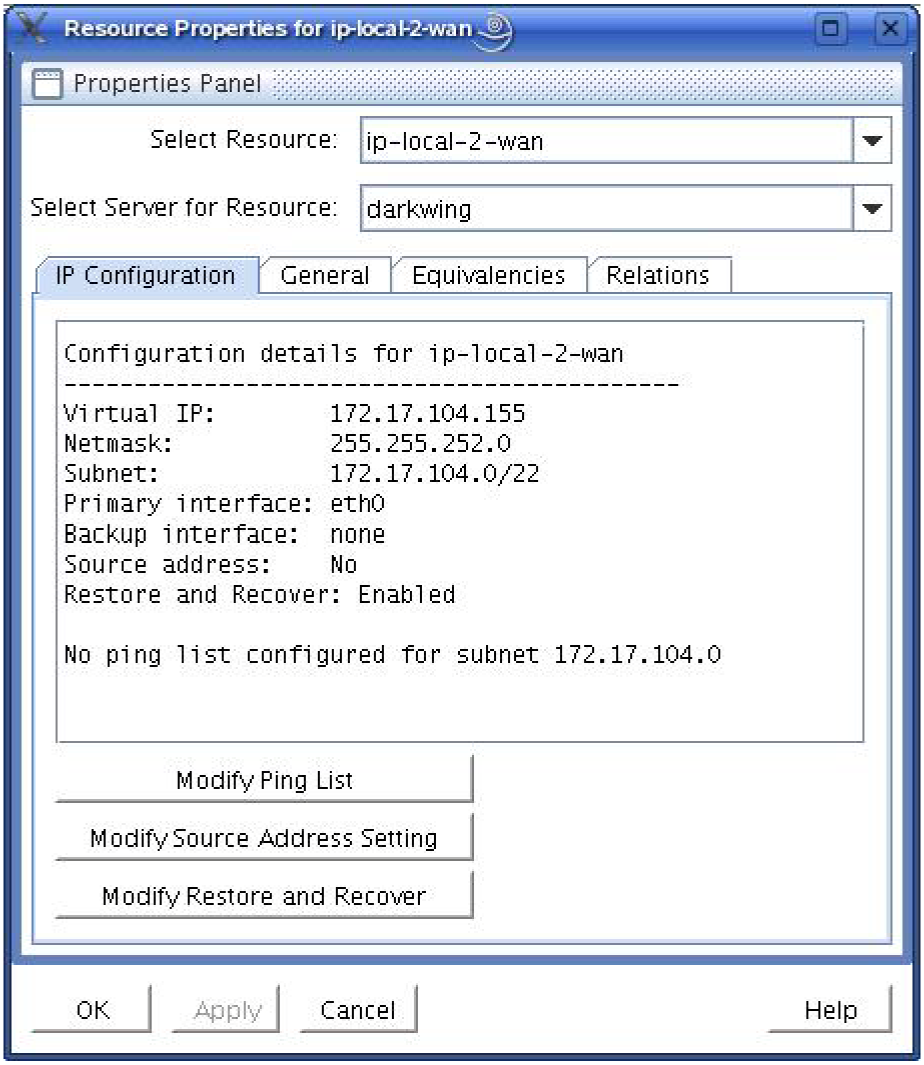
172.17.107.13
172.17.107.254
172.17.104.20

To save the modified list, click **Save List**. This produces the following confirmation window.



```
Saving new ping list for subnet 172.17.104.0:
---------------
172.17.107.13
172.17.107.254
172.17.104.20
```

Click **Done** to close the window, bringing you back to the **IP Configuration** properties page, where you can see the modified *Ping List*.



Select Resource: ip-local-2-wan

Select Server for Resource: darkwing

```
Configuration details for ip-local-2-wan
--------------------------------------------
Virtual IP:        172.17.104.155
Netmask:           255.255.252.0
Subnet:            172.17.104.0/22
Primary interface: eth0
Backup interface:  none
Source address:    No
Restore and Recover: Enabled

No ping list configured for subnet 172.17.104.0
```

Important Notes About Using a Ping List

A *Ping List* for an IP resource is unique to the LifeKeeper system on which it is configured. If the IP resource is extended to another LifeKeeper system after the *Ping List* has been created, the *Ping List* will be copied to the other system as a part of the extension. However, if the IP resource has already been extended, the *Ping List* must be configured individually for each system on which the IP resource is defined. *Ping List* modifications can be made to an IP resource regardless of its state, so there is no need to perform switchovers of the IP resource in order to modify the *Ping List* on each system.

If there are multiple IP resources defined on the same logical subnet, all of those IP resources share a common *Ping List*. This is reflected in the **IP Configuration** properties page and the dialogs associated with modifying the *Ping List*, where the list is identified as being for the subnet associated with the IP resource.

Once a *Ping List* has been defined for an IP resource, all health checks for that resource will use the *Ping List* mechanism rather than the default broadcast ping mechanism. To revert back to the broadcast ping mechanism, you must delete the *Ping List* by removing all of the address entries in the list.

LifeKeeper performs no validation of the IP addresses entered into a *Ping List*, other than ensuring that they are validly formatted addresses. It is important that you ensure that the addresses you are entering actually exist on your network, can be pinged from the LifeKeeper systems, and are expected to be active at all times. You should not choose addresses that exist on the LifeKeeper systems themselves, because local pings to such addresses may be successful regardless of the actual status of the network interface on which the monitored IP resource is defined.

As mentioned above, the definition of a *Ping List* for an IP resource on a given system causes LifeKeeper to automatically use the *Ping List* mechanism rather than a broadcast ping for that resource and all other IP resources on the same subnet. It is not necessary to disable the broadcast ping mechanism using the **`NOBCASTPING`** setting described in the *Adjusting IP Recovery Kit Tunable Values* section below. However, if you have a configuration in which there are no systems available on the network to respond to a broadcast ping, you may have to use the **`NOBCASTPING=1`** setting initially in order get the IP resource created, before you can then define a *Ping List* using the procedure described above. Once the *Ping List* has been created, you can revert back to the default **`NOBCASTPING=0`** setting.

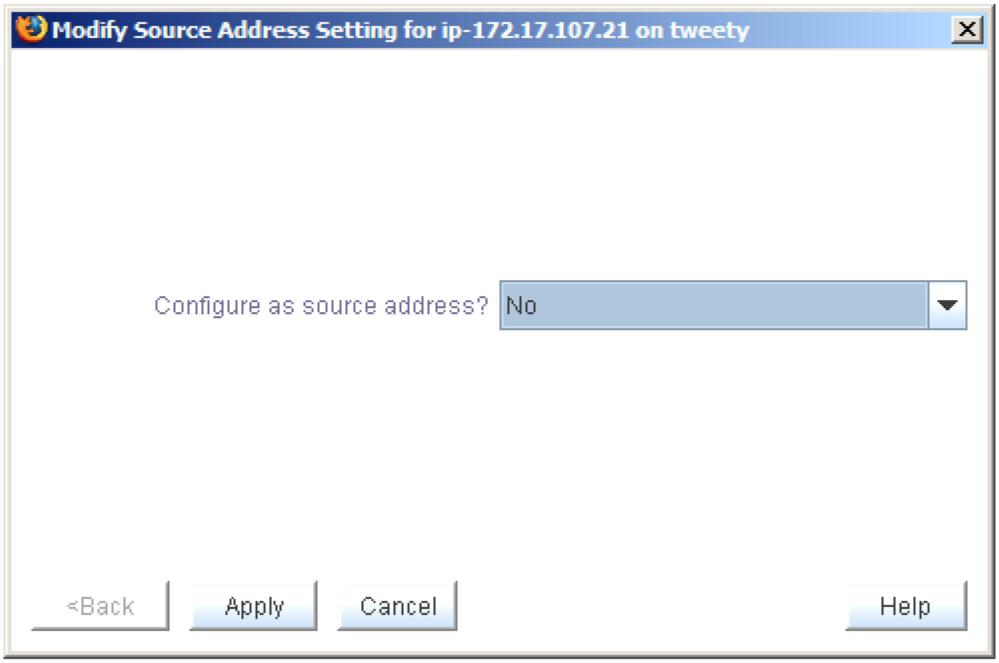
Modifying the Source Address Setting

The *Source Address Setting* for an IP resource determines whether the virtual IP address should be used as the source address for outgoing traffic onto the subnet associated with that IP resource, when the IP resource is in-service. This value defaults to **No**, which means that if the virtual IP address is on the same subnet as the primary IP address for the network interface, outgoing traffic onto the subnet will normally appear to be coming from that primary IP address. This is usually appropriate for most configurations, because the virtual IP address is generally used as an incoming connection point for clients, meaning that all connections in which the virtual IP address is used are initiated as incoming traffic.

However, there may be situations or configurations in which it is important for connections initiated from the LifeKeeper system to appear to be coming from the virtual IP address. By changing the *Source Address Setting* for the IP resource to **Yes**, when the IP resource is brought in-service, the TCP/IP routes on the system are modified such that this will be the case.

Note that if the virtual IP address is on its own distinct logical subnet from the permanent IP addresses on the system, all outgoing traffic onto that subnet will always come from the virtual IP address without any modifications to the *Source Address Setting*.

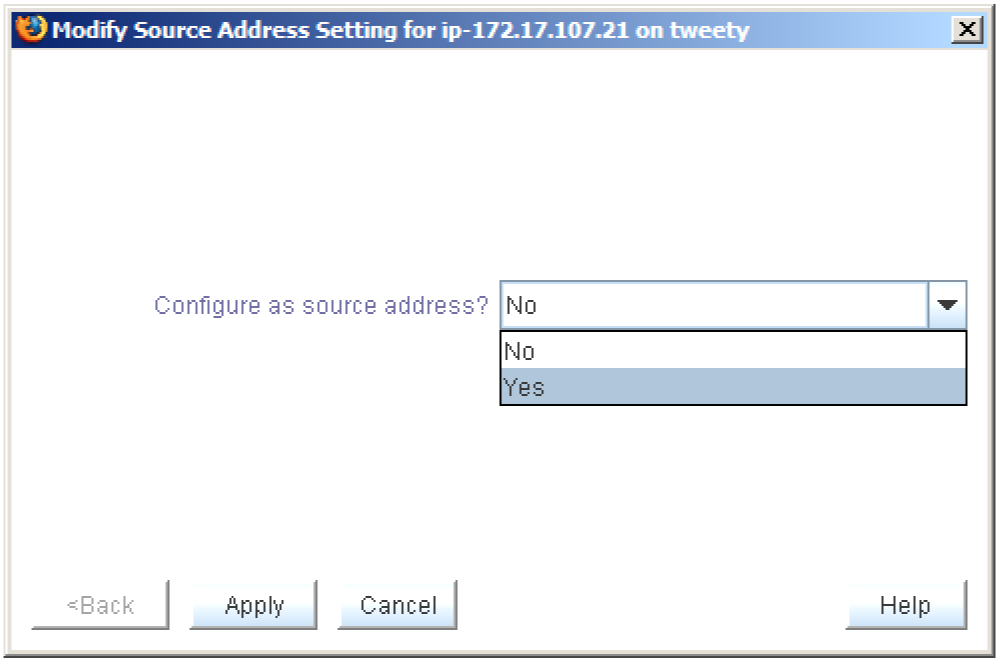
To modify the *Source Address Setting* for an IP resource, click the **Modify Source Address Setting** button on the **IP Configuration** properties page. This brings up a dialog window similar to the following example.



Modify Source Address Setting for ip-172.17.107.21 on tweety

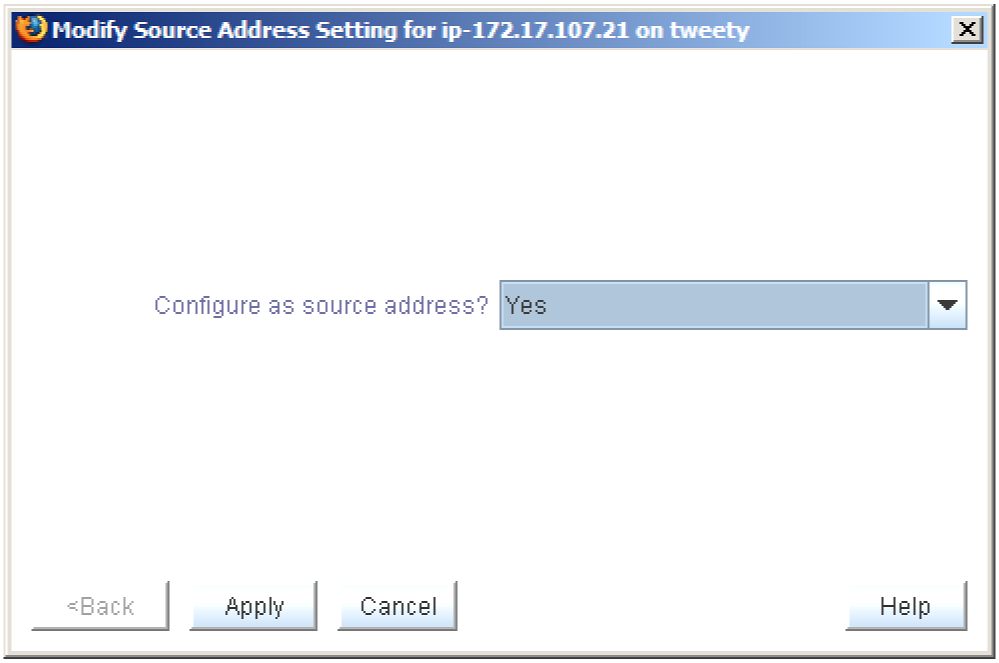
Configure as source address? No

To change the setting, use the drop down list to select the new value, either **Yes** or **No**.



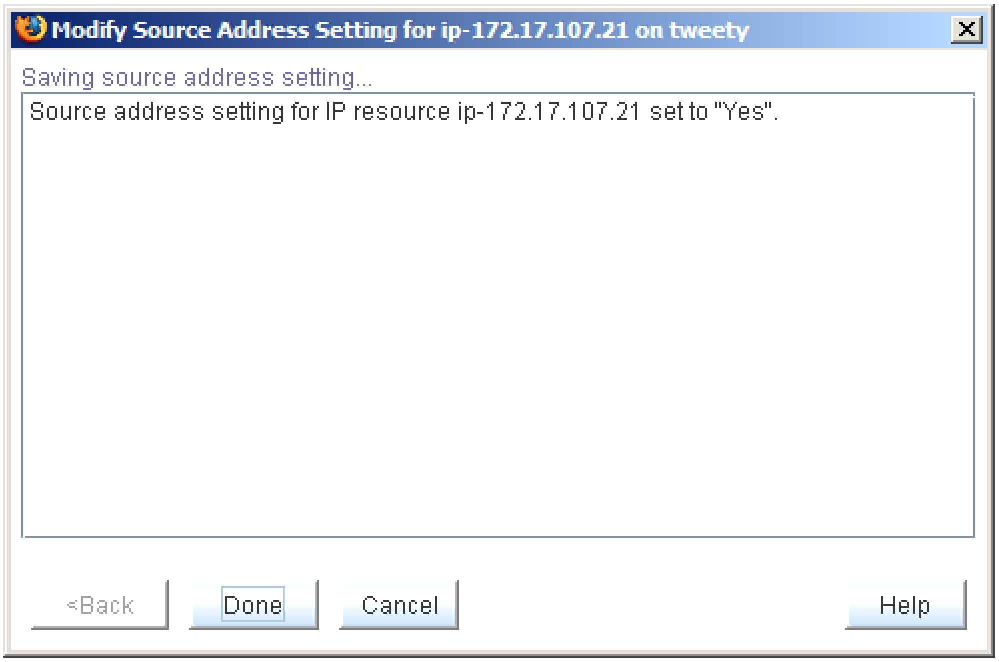
Modify Source Address Setting for ip-172.17.107.21 on tweety

Configure as source address? No


Modify Source Address Setting for ip-172.17.107.21 on tweety
<Back
Apply
Cancel
Help

Configure as source address? Yes

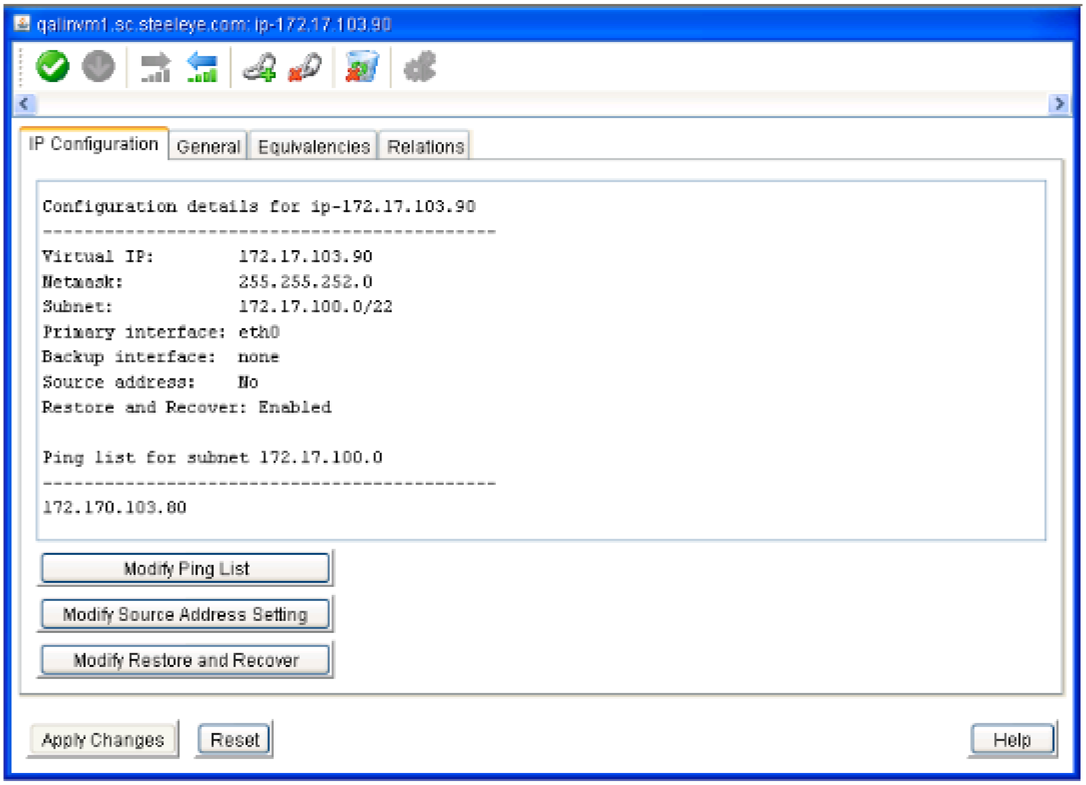
Click **Apply** to save the new setting. This produces the following confirmation window.


Modify Source Address Setting for ip-172.17.107.21 on tweety
<Back
Done
Cancel
Help

Saving source address setting...

```
Source address setting for IP resource ip-172.17.107.21 set to "Yes".
```

Clicking **Done** will close the window and take you back to the **IP Configuration** properties page, where you can see the modified setting.


qalinwm1.sc.steeleye.com: ip-172.17.103.90
IP Configuration
General
Equivalencies
Relations

```
Configuration details for ip-172.17.103.90
--------------------------------------------
Virtual IP:         172.17.103.90
Netmask:            255.255.252.0
Subnet:             172.17.100.0/22
Primary interface:  eth0
Backup interface:   none
Source address:     No
Restore and Recover: Enabled

Ping list for subnet 172.17.100.0
--------------------------------------------
172.170.103.80
```

Modify Ping List
Modify Source Address Setting
Modify Restore and Recover
Apply Changes
Reset
Help

Important Notes About the Source Address Setting

The *Source Address Setting* for an IP resource is unique to the LifeKeeper system on which it is configured. If the IP resource is extended to another LifeKeeper system after the *Source Address Setting* has been modified, the setting will be copied to the other system as a part of the extension. However, if the IP resource has already been extended, the *Source Address Setting* modification must be made individually for each system on which the IP resource is defined.

It only makes sense for at most one IP resource on a given subnet to have its *Source Address Setting* set to **Yes**, because only a single IP address can actually be the source address for outgoing traffic onto the subnet. If there are multiple IP resources on the same subnet with a setting of **Yes**, the most recent resource to be brought in-service will override any others and become the source address for outgoing traffic onto the subnet.

The *Source Address Setting* only affects the local TCP/IP configuration when the IP resource is brought into service. So if the resource is already active when the setting is changed, the resource must be taken out-of-service and then back in-service before the change is reflected in the TCP/IP configuration.

Modifying Restore and Recover

The *Restore and Recover* setting for an IP resource should be set to Disable in a Multi-Site Cluster environment where the disaster recovery system is on a different subnet.

This feature allows a user to choose to enable or disable the default restore and recovery behavior for an existing IP address resource. If configured with the Enable option, the IP address will be brought in-service as normal and the regular monitoring and recovery process will occur. The Enable option is the current default behavior for an IP address restore.

If the Restore and Recover option is set to **Disable**, the resource will come in-service, but the IP address will not be brought active on the network or network adapter. This setting allows hierarchies in a Multi-Site Cluster (or WAN) environment that depend on an IP to be brought in-service on the Disaster Recovery (DR) system where it may be difficult to configure the IP on the DR system due to the WAN configuration.

This setting can be selected after the resource is created and extended.

Important consideration for Active IP addresses (ISP): Setting the action to Disable on an ISP and active IP address, does not take the active IP out of service.

Adjusting IP Recovery Kit Tunable Values

The table below lists and explains the tunable values that are available for modifying the behavior of the IP Recovery Kit. These values are tuned by editing the */etc/default/LifeKeeper* configuration file. Because none of the components of the IP Recovery Kit are memory resident, changes to these particular values become effective immediately after they are changed in */etc/default/LifeKeeper*, without requiring a LifeKeeper restart.

Tunable Value	**Explanation**
`NOBCASTPING`	Can be used to disable the broadcast ping mechanism for checking the health of IP resources. 0 = Keep the broadcast ping test enabled (Default) 1 = Disable the broadcast ping test
`NOIPUNIQUE`	Can be used to disable the check that an IP address is not already active somewhere on the network before it is brought in-service. 0 = Keep the IP uniqueness check enabled (Default) 1 = Disable the IP uniqueness check
`IP_PINGTIME`	Time in seconds that LifeKeeper will wait for a ping reply during IP health checks. Default = 1 (Note: When using a manually configured *Ping List* rather than the broadcast ping mechanism, any value greater than 3 for this tunable is ineffective, because the Linux TCP/IP implementation always returns a “Destination Host Unreachable” error after 3 seconds with no reply, regardless of the timeout value specified in the ping command.)
`IP_PINGTRIES`	The number of ping retries that will be performed during an IP health check. Default = 3
`IP_PINGPRELOAD`	The number of ping packets that will be preloaded onto the network during an IP health check. Default = 1
`IP_NOSAVEREPLY`	Can be used to disable the saving of the address that first responds to a broadcast ping for use in subsequent IP health checks. 0 = Keep address saving and use enabled (Default) 1 = Disable saving and use of responding address
`IP_NOLINKCHECK`	Can be used to disable the link status check portion of the IP health check. 0 = Keep the link status check enabled (Default) 1 = Disable the link status check

Troubleshooting

This section provides a list of messages that you may encounter during the process of creating and extending a LifeKeeper IP resource hierarchy, removing and restoring a resource, and, where appropriate, provides additional explanation of the cause of the errors and necessary action to resolve the error condition. Other messages from different LifeKeeper scripts and utilities are also possible. In these cases, please refer to the documentation for the specific script or utility. Messages in this section fall under these topics:

- Hierarchy Creation
- Extend Hierarchy
- Hierarchy Remove, Restore and Recovery
- Local Recovery

Hierarchy Creation Error Messages

You may encounter the following errors while you are creating your IP resource:

***ERROR* must specify a create flag**

***ERROR* must specify machine name containing primary hierarchy**

***ERROR* must specify machine name**

***ERROR* must specify application**

***ERROR* must specify resource id**

***ERROR* must specify tag**

***ERROR* must specify primary IP Resource Tag**

***ERROR* must specify IP resource name**

***ERROR* must specify Primary Parent Application Resource Tag**

***ERROR* must specify switchback type selection**

***ERROR* must specify Primary Switchback Type**

***ERROR* must specify primary network interface**

***ERROR* must specify primary netmask**

***ERROR* invalid choice of netmask**

The value entered for the netmask is out of range. The value must be in the form of 255.xxx.yyy.zzz, where xxx, yyy, and zzz are between 0 and 255.

***ERROR* must specify init state**

***ERROR* must specify state**

***ERROR* no primary tag found**

ERROR: Couldn't determine IP address from xxx (where xxx is the hostname)

The entered hostname did not translate to an IP address using the /etc/hosts file or DNS.

ERROR: xxx is already an active IP resource on <machine_name> (where xxx is an IP address)

Unknown error in script ipins, err=#

Unknown error in script iphier, err=#

xxx: not found in /etc/hosts or DNS (where xxx is the hostname)

The entered hostname did not translate to an IP address using the /etc/hosts file or DNS.

Interface "ethx" not found on machine yyy (where "ethx" is the interface and yyy is the name of the template or target server)

Interface ethx was not configured on server *yyy* or there is a problem with the networking hardware or software such that it no longer appears to be configured (does not appear in *netstat -i* list).

Invalid interface "ethx" for alias yyy (where "ethx" is the interface and yyy is the IP address)

The combination of address yyy, the specified netmask, and *xxx* would result in a routing conflict (same subnet, different interfaces). Click on the drop down box to specify a different network interface.

Network is down.

TCP is not operational. *netstat -i* shows no interfaces configured.

An error occurred during creation of LifeKeeper application=comm on xxx (where xxx is the system name).

An error occurred during creation of LifeKeeper resource type=ip on xxx (where xxx is the system name).

ins_create failed on machine xxx (where xxx is the name of the server).

ins_setinfo failed on machine xxx (where xxx is the name of the server)

dep_create failed on machine xxx (where xxx is the name of the server)

Netmask of xxx causes entered IP address of yyy to be the subnet (where xxx is the Mask, and yyy is the Tag)

You are not allowed to protect the subnet address for the network.

Netmask of xxx causes entered IP address of yyy to be the broadcast address for the subnet (where xxx is the Mask, and yyy is the Tag)

You are not allowed to protect the broadcast address for the network.

Extend Hierarchy Error Messages

You may see the following errors while extending an existing IP resource from a template server to a target server. Any errors detected during an extension operation will cause the Hierarchy Extend Manager to stop immediately and to perform a roll back procedure that removes the extension of the current resource thus far.

Most Extend errors are preceded by an error header that includes the following basic information to help identify the problem:

ERROR - (TemplateServerName, TemplateTagName, ApplicationName\ResourceType, TargetServerName) - Message

The following error messages may appear:

ERROR - canextend (Error Header) - Missing required input parameters

ERROR - canextend (Error Header) - Required template machine name is null

ERROR - canextend (Error Header) - Required template resource tag name is null

ERROR - canextend (Error Header) - Script was terminated by a signal

ERROR - canextend (Error Header) - Unable to access template resource xxx (where xxx is the template resource tag name)

The IP resource on the template no longer exists, or a communication failure has occurred between the target server and the template server.

ERROR - extIPres (Error Header) - Missing required input parameters

ERROR - extIPres (Error Header) - Input Template System Name is null

ERROR - extIPres (Error Header) - Input Template Tag Name is null

ERROR - extIPres (Error Header) - Input Switchback Type is null

ERROR - extIPres (Error Header) - Input IP Resource is null

ERROR - extIPres (Error Header) - Input Subnet Mask is null

ERROR - extIPres (Error Header) - Input Network IF is null (where “IF” is “Interface”)

ERROR - extIPres (Error Header) - Input Target System Name is null

ERROR - extIPres (Error Header) - Input Target Tag Name is null

ERROR - extIPres (Error Header) - Input Target LifeKeeper ID is null

ERROR - extIPres (Error Header) - The subnet mask validation failed

An invalid subnet mask was used when extending the IP resource to the target server.

ERROR - extIPres (Error Header) - xxx was not found in /etc/hosts of DNS (where xxx is the IP Resource)

ERROR - extIPres (Error Header) - The app_create command failed to create an application for xxx (where xxx is the Target Server Tag Name)

The *extIPres* module is reporting that the LifeKeeper API *app_create* failed when attempting to create a new application type to be used by the extended IP resource on the target server. Either a required parameter for *app_create* was missing, or the LifeKeeper installation on the target server has been corrupted.

ERROR - extIPres (Error Header) - The typ_create command failed to create a resource type for xxx (where xxx is the Target Server Tag Name)

The *extIPres* module is reporting that the LifeKeeper API *typ_create* failed when attempting to create a new resource type to be used by the extended IP resource on the target server. Either a required parameter for *typ_create* was missing or the LifeKeeper installation on the target server has been corrupted.

ERROR - extIPres (Error Header) - The ins_create command failed to create a resource for xxx (where xxx is the Target Server Tag Name)

The *extIPres* module is reporting that the LifeKeeper API *ins_create* failed when attempting to create a new resource object to be used by the new extended IP resource on the target server. Either a required parameter for *ins_create* was missing, or the LifeKeeper installation on the target server has been corrupted.

ERROR - extend (Error Header) - Missing required input parameters

ERROR - extend (Error Header) - Required template machine name is null

ERROR - extend (Error Header) - Required template resource tag name is null

ERROR - extend (Error Header) - Required switchback type selection is null

ERROR - extend (Error Header) - Required IP Resource is null

ERROR - extend (Error Header) - Required subnetwork mask is null

ERROR - extend (Error Header) - Required network interface is null

ERROR - extend (Error Header) - Required target resource tag name is null

ERROR - extend (Error Header) - Script was terminated by a signal

ERROR - extend (Error Header) - Unable to access template resource xxx (where xxx is the Target Server Tag Name)

ERROR - extend (Error Header) - LifeKeeper internal ID xxx was already being used by another resource type on yyy (where xxx is the LifeKeeper ID on the Template Server and yyy is the Target Server Tag Name)

A previously existing LifeKeeper resource (not an IP resource) on the target server had already been assigned the same LifeKeeper ID that was to be used for the extended IP resource on the target server.

Unknown error in script extIPres, err=#

Unknown error in script extend, err=#

Hierarchy Restore, Remove and Recover Error Messages

The following errors may appear during a restore or remove process (switchover or failover). The messages indicate that the switchover process has failed and appropriate action must be taken. If the invalid interface message appears under these circumstances, it is most likely due to the manual configuration of a conflicting address since the resource was last brought into service.

Interface xxx not found on server yyy.

Interface *xxx* was not configured on server *yyy* or there is a problem with the networking hardware or software such that it no longer appears to be configured (does not appear in *netstat -i* list).

Invalid interface xxx for alias yyy (where xxx is a network interface and yyy is an IP address)

The combination of address yyy, and netmask, and *xxx* would result in a routing conflict (same subnet, different interfaces). Most likely, this is due to a network reconfiguration outside of LifeKeeper.

Network is down.

netstat -i shows no interfaces configured.

xxx already in alias list of yyy (where xxx is an IP address and yyy is either the template or target server)

This is an error only if it is an attempt to restore *a.b.c.d* into service on an interface different than the one it was assigned to when created. Trying to restore an address already in-service on its proper network interface is not an error (although you might see this message when restarting LifeKeeper after executing *lkstop -f*).

xxx is currently in-use somewhere (where xxx is an IP address)

Sometime since this resource was last put in-service, a duplicate address has appeared on the net. Since access to the resource (and any dependent applications) is unreliable and unpredictable under such conditions, the hierarchy is not brought into service. Find and eliminate the source of the duplicate and manually bring the hierarchy into service.

xxx is the only IP associated with yyy (where xxx is an IP address and yyy is either the template or target server)

This is an attempt to remove the last address configured on yyy. This will most likely come about when someone inadvertently unconfigures the *physical* interface. Stopping and restarting the network should fix this.

xxx is not an alias on yyy (where xxx is an IP address and yyy is either the template or target server)

An attempt was made to remove a resource from service on server *yyy* that was not really in-service (did not appear in *ifconfig -a* or *netstat -i* list). The remove operation and any associated switchover will not fail because of this, but it is abnormal and may indicate network configuration problems. Stopping and restarting the network will usually clear these errors.

Loss of xxx on server yyy detected; remove proceeding.
xxx on server yyy is down; remove proceeding. (where xxx is a network interface and yyy is either the template or target server)

These messages indicate that LifeKeeper detected a problem with the physical interface from which the logical interface is being removed. The *remove* operation and any associated *restore* will proceed, but you should verify the status of your network on server *yyy*.

IP Health Check and Local Recovery Error Messages

The following are the error messages that may be produced by the IP resource monitoring mechanisms. Monitoring is performed periodically by the IP quickCheck mechanism, but also during the IP restore and local recovery operations, so these messages may appear in the logs under many different scenarios.

IP health check: ERROR: Link check failed for virtual IP xxx on interface yyy.

The IP link status check failed, indicating that the network interface may not be properly connected to the physical network.

IP health check: ERROR: Virtual IP xxx is not active on interface yyy.

The IP health check mechanism found that the virtual IP address is no longer configured and active on the specified network interface.

IP health check: ERROR: Broadcast ping test failed for virtual IP xxx on interface yyy.

The broadcast ping test mechanism has failed for the indicated virtual IP address, indicating that no responses were received to a broadcast ping sent out from the virtual IP address within the currently configured retry and timeout parameters.

IP health check: ERROR: List ping test failed for virtual IP xxx on interface yyy.

No ping responses were received from any of the addresses in the manually configured *Ping List* for the virtual IP address.

The following are error messages which may be produced by the **lkipbu** command when it is being run by the user. Some of these errors may also appear in the LifeKeeper logs as a result of the **lkipbu** command being used internally by LifeKeeper during local recovery operations.

lkipbu: Error: Another lkipbu command is currently running.

Indicates that another instance of the **lkipbu** command is currently running on the system. Only one instance of the **lkipbu** command can be active at a time.

lkipbu: Error: No matching resource instance with tag xxx

The tag provided to the **lkipbu** command with the **-t** argument is invalid.

lkipbu: Error: Resource xxx is not an IP resource instance

The specified resource tag name is a valid resource instance, but identifies a resource of some type other than an IP resource (for example, a file system, database, etc.)

lkipbu: Error: xxx already has a backup interface defined

You have attempted to add a backup interface to an IP resource instance which already has a backup interface. You must first remove the existing backup interface (via **lkipbu -r**) before adding a new one.

lkipbu: Error: xxx is not a valid backup interface

The backup interface provided to the **lkipbu -a** command is invalid, probably because the interface does not exist or is not active on the local system.

lkipbu: Error: Interface xxx is not the current backup interface for yyy

The backup interface provided to the **lkipbu -r** command is not the current backup interface for the IP resource instance yyy. Use the **lkipbu -q** command to determine the current backup interface value for this resource.

lkipbu: Error: xxx has no backup interface

You have attempted to move an IP address to its backup interface via the **lkipbu -m** command, but the resource instance xxx has no backup interface defined.

lkipbu: Error: could not bring xxx in-service on backup interface yyy

The **lkipbu -m** command was unable to activate the IP address xxx on the backup interface yyy, probably due to network configuration problems. Look for additional error messages prior to this one for further explanation of the exact cause of the problem.

lkipbu: Error: broadcast ping failed after address move to xxx, removing address

The **lkipbu -m** command was able to move the IP address to the backup interface, but did not get any response to the broadcast ping test after the address was activated. The backup interface may not be properly connected to the same physical network as the primary interface, or there may be no other active systems on the subnet to respond to the broadcast ping test.

The following errors may appear in the LifeKeeper log file as a result of errors in the IP local recovery procedure:

LifeKeeper: pingfail: Local recovery failed for IP instance xxx

All attempts at local recovery of IP instance xxx have failed. Look for additional error messages prior to this one for further explanation of the exact cause of the problem. This error should result in the IP instance and its associated hierarchy being failed-over to a backup server.

/opt/LifeKeeper/subsys/comm/resources/ip/actions/quickCheck: Local recovery still in progress: xxx

lkcheck has attempted to run the IP quickCheck script for the IP resource instance xxx before the local recovery attempt from a previous quickCheck failure for that instance was completed. This and further quickCheck attempts will be aborted until the local recovery attempt has been completed.

Troubleshooting Hints and Tips

Changing Interfaces

When you need to replace a LAN card with another of the same type, you probably do not need to perform any steps related to your IP hierarchies. Complete the following steps:

1. **Shut down the server.** If the server is going to be down for an extended period, you may wish to first manually switch over all resources to the backup server(s).
2. **Replace the defective card.** Refer to the documentation accompanying the card for instructions.
3. **Reboot the server.**
4. **Restore resources.** Execute the **lkstart** from the command line to start LifeKeeper, if necessary. (If you switched resources to the backup server, you can bring them back *in-service* at this time.)

If you are changing interface or adapter types (as in an upgrade from Ethernet to FDDI, or a switch to an adapter from a different vendor), you will most likely need to reconfigure TCP/IP and remove and recreate the resource. This is necessary to account for the fact that the interface name itself will probably be changed, as well as the network address.

Clients' ARP Cache not getting updated

If your clients are unable to reconnect to the server using the protected TCP/IP address after a manual switchover or a failover, then the clients' ARP cache is probably not getting updated correctly. This is most likely because your clients do not support a full TCP/IP implementation and are not responding to the ARP table update mechanism used by LifeKeeper. If this is the case, the Client ARP cache entries will not get updated until the old values time out. You should adjust the ARP cache timeout values on the clients to a lower value so that the old ARP table entries are removed sooner, allowing the clients to reconnect.

False Failovers

LifeKeeper monitors IP resources by executing a broadcast ping on the IP addresses logical subnet and then listening for replies. There are several cases when this check will fail, resulting in a failover of an IP resource to the backup server. The cases are:

- There is a legitimate long-term network, address or interface failure.
- There is a short-term network failure that occurred during the interval when the check was executed. If you have had a failure, but the network seems functional, look for signs of a possible temporary outage at the time of the failover. The time of the failure can be determined by looking at the LifeKeeper log.
- There were no other servers on the network to respond to the broadcast ping. This is most likely to occur when you have configured an IP resource with an address on a different

logical subnet from the other addresses on the interface where it is configured, and no other addresses on that logical subnet exist on the network.

For example, consider the network shown in Figure 1. If interface *eth0* is configured for logical subnet 25.0.3, and you create an IP resource for the IP address 25.0.10.1 with a netmask of 255.255.255.0 (logical subnet 25.0.10), there must be at least one other server connected to the physical network with an IP address configured on the 25.0.10 logical subnet.

If your network configuration is such that there are no other systems on the network available to respond to the broadcast ping, you should consider defining a *Ping List* for the IP resource, as an alternative to the broadcast ping health check mechanism.