



LifeKeeper[®] for Linux v5.0

Command Line Interface Guide

August 2005

SteelEye, SteelEye Technology, and LifeKeeper are registered trademarks of SteelEye Technology, Inc.

Java is a registered trademark of Sun Microsystems, Inc. Linux is a registered trademark of Linus Torvalds. Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other brand and product names used herein are for identification purposes only and may be trademarks of their respective companies.

It is the policy of SteelEye Technology, Inc. to improve products as new technology, components, software, and firmware become available. SteelEye Technology, Inc., 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.

Address correspondence to:
ip@steeleye.com

Copyright © 2005
By SteelEye Technology, Inc.
Palo Alto, CA U.S.A.
All Rights Reserved

Table of Contents

Introduction	3
Document Contents	3
LifeKeeper Documentation	3
Commands.....	4
Groupings and Basic Descriptions of LifeKeeper bin Commands	4
Starting and Stopping LifeKeeper, the GUI, etc.	4
Monitoring LK and Other Misc. LifeKeeper Commands	4
Bringing a Hierarchy into and out of Service.....	5
SYS - LifeKeeper Commands Related to the Systems in the LifeKeeper Cluster	5
NET - Communications Paths Related Commands.....	6
FLAG - Commands Related to Internal LifeKeeper Flags.....	7
TYP - LifeKeeper Commands Related to Resource Hierarchy Types	7
APP - LifeKeeper Commands Related to Resource Applications (Group of Related Types).....	8
DEP - LifeKeeper Commands Related to How Resource Applications Relate to Each Other.....	8
INS - Commands Related to Individual LifeKeeper Hierarchy Instances.....	9
Unextend a Hierarchy	10
Shell Script Examples.....	11
Some Examples Pulled Out of Shell Scripts to Create and Extend Hierarchies.....	11
Creating IP Hierarchy	11
Extend IP Hierarchy	11
Create File System Hierarchy	12
Extend File System Hierarchy.....	12
Create Oracle Hierarchy	12
Extend Oracle Hierarchy	12
List Hierarchy Dependencies	12
Create Dependency Between Hierarchies	12

Command Line Interface Guide

Introduction

LifeKeeper's Command Line Interface can be used as an alternative to the Graphical User Interface. Common tasks may be automated by incorporating calls to the CLI in shell scripting.

Document Contents

This guide contains the following topics:

- [LifeKeeper Documentation](#). Provides a list of LifeKeeper for Linux documentation and where to find each document.
- [Commands](#). Describes CLI commands.
- [Shell Script Examples](#). Provides some examples of CLI use in scripting.

LifeKeeper Documentation

The following LifeKeeper product documentation is available from SteelEye Technology, Inc.:

- *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 SteelEye Technology, Inc. website at:

www.steeeye.com/support/documentation

Commands

Groupings and Basic Descriptions of LifeKeeper bin Commands

The commands will be in: `/opt/LifeKeeper/bin`

To place this in your path, execute: `‘./etc/default/LifeKeeper’`

Starting and Stopping LifeKeeper, the GUI, etc.

lkstart Start LifeKeeper core

Options: **-w** Timeout wait period in seconds if LK does not start properly

lkstop Stop LifeKeeper core

Options: **-f** Do not stop the protected resource

lktest Test to see if LifeKeeper is currently running

Options: none

lkGUIserver Start and stop the LifeKeeper GUI daemon processes

Options: **start**

stop

lkGUIapp Starts the LifeKeeper Java application

Options: none

lktest Checks to see if LifeKeeper is configured and running properly

Monitoring LK and Other Misc. LifeKeeper Commands

lk_log Manage, display, and output LifeKeeper logs to a file

Options: **-f** Keeps log file open to display new log data

-t # where # is to list the last # lines of the file

<log file> ex log, GUI, LCM, LCD

lk_log log > file_name outputs log file into file_name

lcdstatus Display status of LifeKeeper resources, comm paths, etc.

Options: **-d <node to run command on>**

-q short reports

lcdsync Writes LifeKeeper configuration information from memory to disk

Options: **-d <other node to run it on>**

lcdrcp Transfer files from one LifeKeeper node to another via the comm. path

Options: **lcdrcp <file names> {dest:ofile | dest:odir}**

lcdremexec Execute the given command on the given LifeKeeper node

Options: **-d <node to run command on> <command>**

lcdrecover Checks and sets resource hierarchy instance settings

Options: see documentation

Bringing a Hierarchy into and out of Service

perform_action Performs a given action on a given resource.

Can be used to switch a given hierarchy to another node.

Options: **-a <action name>**

-t <tag name>

Examples: **perform_action -a restore -t \$LKTag** bring tier into service

perform_action -a remove -t \$LKTag take tier out of service

SYS - LifeKeeper Commands Related to the Systems in the LifeKeeper Cluster

sys_list Lists out the systems known to a particular LifeKeeper node

Options: **-d <other node to run it on>**

sys_create Creates knowledge of another system on LifeKeeper node

Options: **-s <system name>**

-d <node to run command on>

sys_remove Removes knowledge of another system on a LifeKeeper node

Options: **-d <dest>**

-s <system name to be removed from list>

sys_getstate Lists the state of a given LifeKeeper node on the given LifeKeeper node

Options: **-d <node to run command on>**

-s <system concerning the state being checked>

- sys_setstate** Sets the state of a given LifeKeeper node on a given LifeKeeper node
- Options: **-d** <node to run command on>
-s <system concerning the state being set>
-S <actual state> {DEAD|ALIVE|UNKNOWN}
-R <reason for state setting>
- sys_getdescr** Prints some information of why the system went to its current state
- Options: **-d** <node to run command on>
-s <system to get data on>

NET - Communications Paths Related Commands

- net_create** Creates a communications path between two LifeKeeper nodes
- Options: **-d** <node to run command on>
-s <other system>
-D <device path>
-n <TTY or TCP>
-b <baud rate>
-r <remote IP address>
-l <local IP address>
-p <priority>
- net_remove** Removes a communications path between two LifeKeeper nodes
- Options: **-d** <node to run command on>
-s <server name to be removed from>
-D <device path>
-r <IP address>
- net_list** Lists communications path information on a given LifeKeeper node
- Option: **-d** <node to run command on>
-f: <field separator of ':'>
-s <system name>
- net_change** Modify specific information about a given communications path
- Options: **-d** <node to run command on>
-s <server name for data to be modified>
-D <device>

crelcm Create a communications path

`/opt/LifeKeeper/bin/crelcm <node 1> <node 2> <net type> <baud rate> <IP address 1> <IP address 2> <prio>`

portio Tests the serial connection between two LifeKeeper nodes

FLAG - Commands Related to Internal LifeKeeper Flags

flg_create Set a given LifeKeeper flag on a given LifeKeeper node

Options: **-d <node to run command on>**

-f <flag name>

flg_remove Remove a given LifeKeeper flag on a given node

Options: **-d <node to run command on>**

-f <flag name>

flg_list List all LifeKeeper flags that are set on a given node

Options: **-d <node to run command on>**

TYP - LifeKeeper Commands Related to Resource Hierarchy Types

typ_create Create a given resource type on a given LifeKeeper node

Options: **-d <node to run command on>**

-a <app type> (need an app first)

-r <resource type>

typ_remove Remove a given resource type on a given LifeKeeper node

Options: **-d <node to run command on>**

-a <application type>

-t <resource type>

typ_list Lists all resource types on a given LifeKeeper node

Options: **-d <node to run command on>**

-f: <field separator of ':'>

-a <app type>

APP - LifeKeeper Commands Related to Resource Applications (Group of Related Types)

- app_create** Create a given resource application on a given LifeKeeper node
Options: **-d** <node to run on>
-a <application name>
- app_remove** Removes a given resource application on a given LifeKeeper node
Options: **-d** <dest>
-a <application type>
- app_list** Lists all resource applications on a given LifeKeeper node
Options: **-d** < node to run on>

DEP - LifeKeeper Commands Related to How Resource Applications Relate to Each Other

- dep_create** Creates a dependency between two resource instances
Options: **-p** <parent tag>
-c <child tag>
- dep_remove** Removes a dependency between two resource instances
- dep_list** Lists the dependency relationship between two instances
Options: see on-line documentation
- eqv_create** Creates an equivalency of a given resource between two nodes
Options: **-d** <dest>
-t <first tag name>
-o <second tag name>
-S <other system>
-e SHARED ?
- eqv_remove** Removes an equivalency of a given resource between two nodes
Options: **-d** <dest>
-s <system to get info on>
-t <tag name>
-f: <field separator of ':'>

- eqv_list** Lists equivalency relationships between resource instances
- Options: **-d <dest>**
-s <system to get info on>
-t <tag name>
-f: <field separator of ':'>
- hry_setpri** Sets the priority of a given LifeKeeper node, or hierarchy on the node

INS - Commands Related to Individual LifeKeeper Hierarchy Instances

- ins_create** Define a new resource instance on the given node
- Options: **-d <dest>**
-a <app type> (need an app first)
-r <resource type> (need a resource type first)
-t <tag name>
-i <id>
- ins_remove** Remove a given resource instance on a given node
- Options: **-d <dest>**
-R <root tag>
-r <resource type>
-t <tag name>
-I <id>
- ins_list** Lists the current information of the given resource hierarchy instance
- Options: **-d <dest>**
-f: <field separator of ':'>
-a / -r / -t / -i specify optional app, type, tag, and id info
- ins_setas** Sets the automatic switchback strategy for a given hierarchy
- Options: **-d <dest>**
-t <tag name>
-s <switchback typ> {INTELLIGENT|AUTOMATIC}
- ins_setinit** Define how a given resource should initialize when LifeKeeper starts
- Options: **-d <dest>**
-t <tag name>
-I <init state> {AUTORES_ISP|INIT_ISP|INIT_OSU}

ins_setinfo Define an information string for a given resource hierarchy

Options: **-d** <dest>
-t <tag name>
-v <string of information>

ins_setstate Set the state of a given resource hierarchy on a given node

Options: **-d** <dest>
-t <tag name>
-S <state to set instance> {ISP|ISU|OSU}
-R <reason for state setting>
-A <recursively set all resources that depend on this one>

ins_gettag Lists the tag name of the associated ID

Options: **-i**<id>

Unextend a Hierarchy

/opt/LifeKeeper/lkadm/bin/unextmgr <Node_Name> <Tag_Name>

Shell Script Examples

Some Examples Pulled Out of Shell Scripts to Create and Extend Hierarchies.

Also how to create a dependency between two hierarchies.

Needed System Parameters

```
LKROOT=/opt/LifeKeeper
OBJ_DIR=/opt/LifeKeeper/lkadm
LKBIN=/opt/LifeKeeper/bin
ExtendPath=/opt/LifeKeeper/lkadm/bin
PATH=$PATH:$LKBIN
```

#Generic ARGS

```
LocalServer=unix121.ha.uk.sbphrd.com
TargetServer=unix122.ha.uk.sbphrd.com
Node2Priority=10
```

The above variables are used for the commands below

Creating IP Hierarchy

```
IPCrePath=/opt/LifeKeeper/lkadm/subsys/comm/ip/bin
IPBundle="$IPResourceTag","$ProtectedIP","$Netmask","$NetworkInterface","$BackupNetworkInterface","$IPResourceTag"
```

```
$IPCrePath/creIPhier $LocalServer $ProtectedIP $NetworkInterface $Netmask IP1Tag
$IPPriSwitchBack $IPResourceTag $CreateFlag $BackupNetworkInterface
```

Extend IP Hierarchy

```
$ExtendPath/extmgrDoExtend.pl -p1 -f, "$IPResourceTag" "$TargetServer" "$Node2Priority"
"$IPTargetSwitchBack" \"$IPBundle\"
```

Create File System Hierarchy

```
FSPath=/opt/LifeKeeper/lkadm/subsys/gen/filesys/bin
```

```
FSBundle="$FS1ResourceTag","$MountPoint1","$FS1ResourceTag"
```

```
$FSPath/creFShier $LocalServer $MountPoint1 $FS1ResourceTag $FSPriSwitchBack
```

Extend File System Hierarchy

```
$ExtendPath/extmgrDoExtend.pl -p1 -f, "$FS1ResourceTag" "$TargetServer" "$Node2Priority"  
"$FSTargetSwitchBack" \"$FSBundle\"
```

Create Oracle Hierarchy

```
Oracle1Path=/opt/LifeKeeper/lkadm/subsys/database/oracle/bin
```

```
Oracle1Bundle="$Oracle1Tag","$Oracle1Sid","$Oracle1Home","$Oracle1Tag"
```

```
$Oracle1Path/databasehier $LocalServer $Oracle1Sid $Oracle1Home $Oracle1Tag  
$Oracle1PriSwitchBack
```

Extend Oracle Hierarchy

```
$ExtendPath/extmgrDoExtend.pl -p1 -f, "$Oracle1Tag" "$TargetServer" "$Node2Priority"  
"$Oracle1TargetSwitchBack" \"$Oracle1Bundle\"
```

List Hierarchy Dependencies

```
$LKBIN/dep_list -p $Oracle1Tag | /bin/grep $IPResourceTag
```

Create Dependency Between Hierarchies

```
$LKBIN/dep_create -d $TargetServer -p $IPResourceTag -c $Oracle1Tag
```

Make sure you do an lcdsync after a dep_create

```
$LKBIN/lcdsync -d $LocalServer
```

```
$LKBIN/lcdsync -d $TargetServer
```