LifeKeeper® for Linux
Module 10: Command Line Interfaces

Overview

This module reviews the following topic:
- LifeKeeper Command Line Utilities
- GUI Property Files
- LifeKeeper Scripts

LCD Interfaces (LCDI)

The following programs interface with the LCD. These are located in the $LKROOT/bin (/opt/LifeKeeper/bin) directory.

<table>
<thead>
<tr>
<th>Program</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lc</td>
<td>Lcd monitoring process</td>
</tr>
<tr>
<td>lcddemexec</td>
<td>Sends a remote request over the comm path</td>
</tr>
<tr>
<td>ldcsync</td>
<td>Copies or syncs the version of the LCD in shared memory to disk.</td>
</tr>
<tr>
<td>lcdnrecover</td>
<td>Coordinates the transfer of equivalent resources during a system failover.</td>
</tr>
<tr>
<td>ldcrop</td>
<td>Transfers files to another LK system using LCM</td>
</tr>
<tr>
<td>lkstart</td>
<td>LifeKeeper startup script</td>
</tr>
<tr>
<td>lkstop</td>
<td>LifeKeeper shutdown script</td>
</tr>
</tbody>
</table>

Copyright © 2000-2010 by SteelEye Technology, Inc.
All Rights Reserved World Wide.
LifeKeeper Flag Management CLI

- Create a flag
  `flg_create [-d destsys] -f flg

- Remove a flag
  `flg_remove [-d destsys] -f flg

- Test for a flag
  `flg_test [-d destsys] -f flg

- List flags that have been set
  `flg_list [-d destsys]

System Management CLI

- Create a system reference
  `sys_create [-d destsys] -s sys

- Remove a system reference
  `sys_remove [-d destsys] -s sys

- List known systems
  `sys_list [-d destsys]

- Get system state
  `sys_getst [-d destsys] -s sys

- Get description of why system is in the current state
  `sys_getds [-d destsys] -s sys

Commands that add, delete or change system references must be run on all cluster members. Run lcdsync after each command to sync in-memory LCD with backing store.

Network Management CLI

- Create a communications path
  `net_create [-d destsys] -s sys [-n TCP|TTY] [-D devicepath [-b baudrate] [--remote_IP_address] [--local_IP_address] [--priority]]

- Remove a communications path
  `net_remove [-d destsys] -s sys [-D devicepath] [-b baudrate] [--remote_IP_address] [--local_IP_address]

- List communications paths
  `net_list [-d destsys] [-C] [-n sysname]

- Change communications path properties
  `net_change [-d destsys] -s sys [-D devicepath pair] [-n name]

Commands that add, delete or change communications paths must be run on all cluster members. Run lcdsync after each command to sync in-memory LCD with backing store.
LCDI Instances

Resource Instance Management CLI

- Create a resource
  `ins_create [-d destsys] -a appname -r restyp -t tag -i id [-I AUTORES_ISP | INIT_ISP | INIT_OSU] [-v info] [-a AUTOMATIC | INTELLIGENT]`
- Remove a resource
- List resources
  `ins_list [-d destsys] [-v info] [-s AUTOMATIC | INTELLIGENT]

“Extending” a resource requires creating a resource with the same properties on all cluster members. Then, create a Shared Equivalency between resources on all cluster members. Remember to run the same commands on all nodes where the equivalency is to be defined. Run `lcdsync` after each command to sync in-memory LCD with backing store.

LCDI Relationships

Dependency Relationship Management CLI

- Create a dependency relationship between resources
  `dep_create [-d destsys] [-p parent] [-c child]`
- Remove a dependency relationship between resources
  `dep_remove [-d destsys] [-p parent] [-c child]`
- Lists the dependency relationship between resources

Must be run on each server where resources have a shared equivalency. Run `lcdsync` after each command to sync in-memory LCD with backing store.
**LCDI Relationships**

**Shared Equivalency Relationship Management CLI**
- Relationship between resources on different systems that represent the same physical entity. When systems have a shared equivalency relationship, LifeKeeper ensures that only one system has access to the resource at a time (ISP status).
- When a resource is extended to another server, a shared equivalency relationship is created as the last step.

**Create an equivalency between specified resources**
```
eqv_create [-d destsys] -t tag [-p sysPriority] [-e othersys [-o othertag [-p othersysPriority] =e (COMMON | SHARED | COMPOSITE]
```

**Remove the specified equivalency**
```
eqv_remove [-d destsys] -t tag [-e othersys] =e (COMMON | SHARED | COMPOSITE)
```

**List the equivalency relations for the specified resource**
```
eqv_list [-d destsys] [-t tag] =e (COMMON | SHARED | COMPOSITE)
```

Commands that create or remove an equivalency must be run on each server involved in the equivalency relationship. Runlcdsync after each command to sync in-memory LCD with backing store.

**Other LCDI Functions**

**LCDI Resource Type**
- Create a resource type
- Remove a resource type
- List resource types

**LCDI Applications**
- Create an application
- Remove an application
- List applications
LifeKeeper Alarm Interface

- Based on the UNIX SVR4 alarming mechanism, sendevent
- LifeKeeper utilizes sendevent when a failure or recovery event has been detected in the monitored resource.
- sendevent passes all of its options to an event response script defined by the application
- Alarm classes are under the /opt/LifeKeeper/events directory
- sendevent has the following required options:
  -C event class (e.g. LifeKeeper)
  -E event type (e.g. lcm_avail)
  -m program invoking sendevent
  -n resource instance

Example

- The following sendevent command will notify LifeKeeper that the lcm process is available:
  ```bash
  $LKROOT/bin/sendevent -C lifekeeper -E lcm_avail -s1 -mLK:eventslcm
  ```
- Scripts in the following directory would be invoked:
  ```bash
  $LKROOT/events/lifekeeper/lcm_avail/lifekeeper
  ```
- Based on the information to sendevent, the event response script can be located:
  ```bash
  $LKROOT/events/lifekeeper/<class>/<event type> / <class>
  ```

LRACI

- The LifeKeeper Recovery Action & Control Interface (LRACI) determines the appropriate action script to run.
- The arguments passed to the LRACI determine which action to perform.
- The LRACI software can be invoked through a command line interface.
  ```bash
  $LKROOT/bin/perform_action
  ```
- The restore and remove actions should only be invoked through the LRACI and never directly.
- perform_action does the following:
  - finds the resource specified by the tag name
  - finds the action script specified by the action name
  - executes the action script
**LRACI**

- perform_action first looks for action scripts for a particular \(\text{application} / \text{resource type}\) pair:
  \[
  \$\text{LKROOT}/\text{subsys}/\text{application}/\text{resource type}/\text{actions}/\text{action name}
  \]

- The remove script for the IP resource is:
  \[
  \$\text{LKROOT}/\text{subsys}/\text{comm}/\text{resources}/\text{ip}/\text{actions}/\text{remove}
  \]
  where:
  - \text{application} = \text{comm}
  - \text{type} = \text{ip}
  - \text{action name} = \text{remove}

---

**LifeKeeper GUI Architecture**

**Overview**

- The LifeKeeper GUI
  - Client / server application developed using Java technology
  - Provides a graphical administration interface to LifeKeeper
  - LifeKeeper GUI client is implemented as:
    - Stand-alone Java application
    - Java applet invoked from a web browser
  - LifeKeeper GUI server is implemented as a set of JavaBeans and is mirrored on the client-side by a parallel implementation.
  - Network communication between the client/server beans - RMI.
  - The LifeKeeper GUI server communicates with the LifeKeeper core software via the Java Native Interface (JNI).

---

**GUI Directory Structure**
GUI Wizard Framework

- GUI administration tasks such as Recovery Kit resource hierarchy creation and extension are performed via dialog wizards implemented with Java classes and properties files.
- When a user selects a menu in the GUI corresponding to a Recovery Kit, the GUI client requests the server to load the properties file corresponding to that selection. A dialog wizard reads the properties file. The properties file directs the wizard on its user display and presents a list of client beans for execution.

GUI Wizard Framework

Recovery Kit Interface:
- The dialog wizard provides a customizable interface for Recovery Kit configuration. By plugging into the wizard framework, the Recovery Kit developer has access to numerous classes and dialogs already defined in the GUI.
- To use the framework, Recovery Kit developers provide two or more Java properties files:
  - `<application>_<resource type>.properties`
  - `<application>_<resource type>_extend.properties`
- See the Extender for more information on developing a customized recovery kit.

Leveraging Existing Capabilities

- When attempting to do anything that involves LCD configuration, it is typically easier to leverage GUI property files and the scripts that are called from them rather than utilize the LCDI command lines directly.
- GUI property files are located under:
  - `$LKROOT/htdoc/com/steeleye/LifeKeeper/locale`
- Process:
  - Examine property file(s) to determine required arguments and syntax for invoking the script that performs the function.
  - Create alternative to GUI for gathering appropriate arguments.
  - Invoke script with properly formed arguments.