

# LifeKeeper Web Management Console

1 — Last update: 27 October 2023

SIOS TECHNOLOGY CORP.

# Table of Contents

<b>1. LifeKeeper Web Management Console (LKWMC)</b> .....	<b>4</b>
1.1. Architecture .....	5
1.2. System Requirements .....	7
1.3. Getting Started .....	8
1.3.1. Prerequisites .....	9
1.3.2. Port Settings .....	10
1.3.3. Firewall Rules .....	11
1.3.4. Connecting to the LKWMC .....	12
1.3.4.1. Connecting to the LKWMC Through a Bastion Host in a Cloud Environment.....	13
1.3.5. Configuration Changes .....	14
1.3.5.1. Modifying the GUI Server Port .....	15
1.3.5.2. Modifying the REST API Server Port.....	16
1.3.6. Using Custom SSL Certificates .....	17
1.4. LKWMC GUI Operations and Layout .....	18
1.4.1. Signing In.....	20
1.4.2. Header .....	21
1.4.2.1. Viewing the Online Manual .....	22
1.4.2.2. Viewing the GUI Version.....	23
1.4.2.3. Changing the Display Theme .....	24
1.4.2.4. Signing Out.....	25
1.4.3. Server .....	26
1.4.3.1. Communication Path.....	27
1.4.3.1.1. Creating a Communication Path .....	28
1.4.3.1.2. Deleting a Communication Path.....	30
1.4.3.1.3. Viewing the Status of a Communication Path .....	31
1.4.3.2. Properties .....	32
1.4.3.3. Sign-in status.....	34
1.4.4. Resource .....	35
1.4.4.1. Resource Tree .....	36
1.4.4.1.1. Creating a Resource.....	37
1.4.4.1.2. Extending a Resource Hierarchy .....	38
1.4.4.1.3. Unextending a Resource Hierarchy .....	39
1.4.4.1.4. Deleting a Resource Hierarchy .....	40
1.4.4.1.5. Bringing a Resource In Service .....	41
1.4.4.1.6. Taking a Resource Out of Service .....	42
1.4.4.1.7. Viewing Resource and Server Status.....	43
1.4.4.1.8. Creating or Deleting a Resource Dependency .....	48
1.4.4.1.9. Viewing and Editing Resource Properties .....	49
1.4.4.2. Properties .....	50
1.4.4.2.1. General .....	51

1.4.4.2.2. Equivalencies .....	52
1.4.4.2.3. Relations .....	53
1.4.4.2.4. Recovery Kit-Specific Properties .....	54
1.4.5. Logs.....	55
1.4.6. Licensing .....	56
1.5. Known Issues and Restrictions.....	57

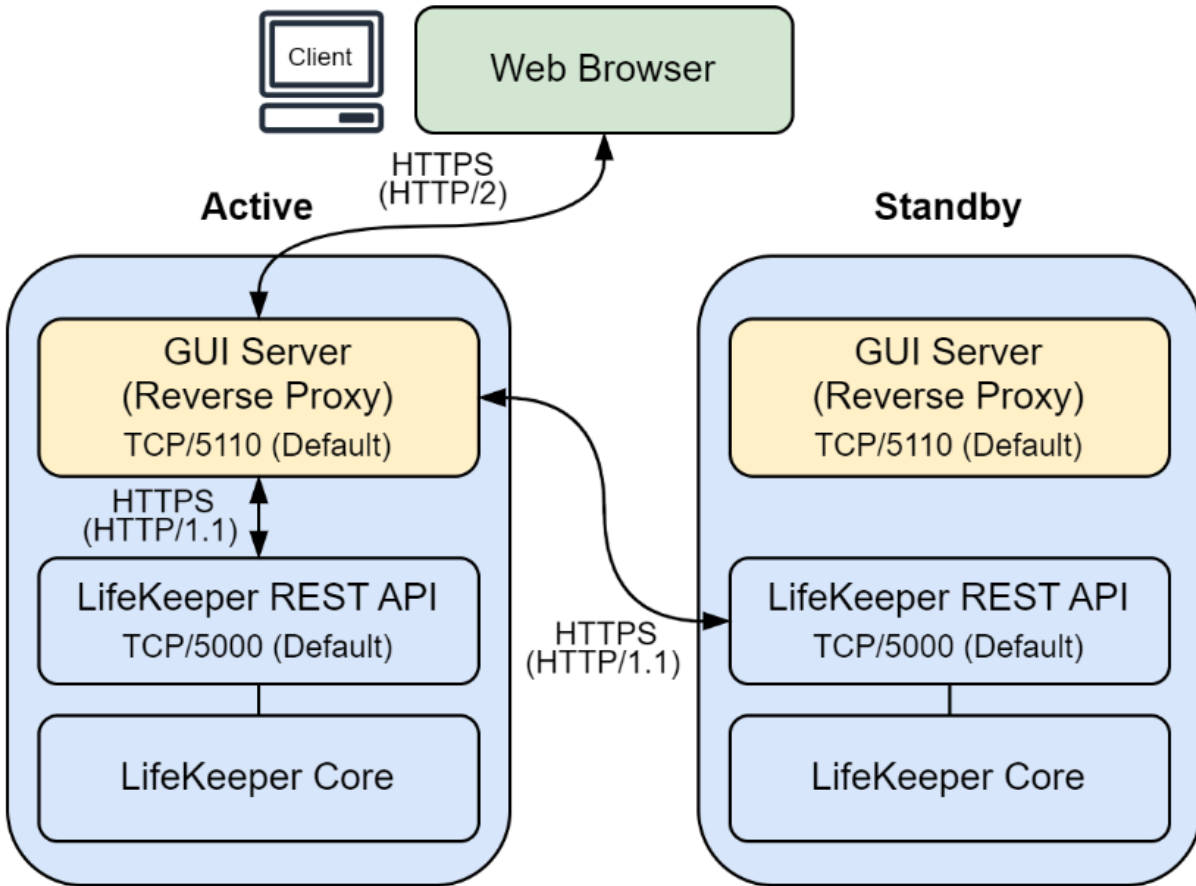
# 1. LifeKeeper Web Management Console (LKWMC)

---

## Overview

The **LifeKeeper Web Management Console** (LKWMC) is a web application that allows users to manage servers and resources in a LifeKeeper cluster from a web browser on a client PC.

## 1.1. Architecture



In the LKWMC, the [GUI Server](#) and [REST API Server](#) components are installed on each node in the LifeKeeper cluster. A user may access the GUI by using a web browser to connect to the appropriate port on any node in the cluster where the GUI server is running. See [Connecting to the LKWMC](#) for more details.

When the user performs an operation or requests information via the UI in the web browser, the GUI server communicates with the REST API server on the appropriate node(s) in the cluster to carry out the operation or obtain the requested information.

This architecture eliminates the need to connect to a dedicated GUI client application (e.g., via SSH X11 forwarding or a VNC session) running directly on a LifeKeeper cluster server, as was required when using the original Java-based LifeKeeper GUI client.

### GUI Server

The GUI server provides the GUI (Graphical User Interface) portion of the LKWMC web application. It is responsible for performing operations by gathering relevant data from the user and making appropriate API requests to the REST API server processes running on the nodes in the LifeKeeper cluster.

## REST API Server

Various RPC and REST API endpoints are exposed which allow for management of the servers and protected resources in a LifeKeeper cluster. When using the LKWMC, the same version of the REST API server must be installed on all of the nodes within the LifeKeeper cluster.

## 1.2. System Requirements

---

The LKWMC is supported for use on the following operating systems and browsers.

### Server Environment

LifeKeeper v9.8.0

### Client Environment

Operating System

Windows 10 / 11 (64-bit)

Web Browser

- Google Chrome (latest version)
- Microsoft Edge (latest version)
- Mozilla Firefox (latest version)

Currently the use of touch-capable displays (such as the touchscreen of a tablet PC or cell phone) is not supported when using the LKWMC. Interactions must be performed using a mouse and keyboard.

## 1.3. Getting Started

---

This section describes the general procedure to connect to the LKWMC.

[Prerequisites](#)

[Port Settings](#)

[Firewall Rules](#)

[Connecting to the LKWMC](#)

[Connecting to the LKWMC Through a Bastion Host in a Cloud Environment](#)

[Configuration Changes](#)

[Modifying the GUI Server Port](#)

[Modifying the REST API Server Port](#)

[Using Custom SSL Certificates](#)



## 1.3.1. Prerequisites

---

Before connecting to the LKWMC, the REST API server must be running on every node in the LifeKeeper cluster, and the GUI server must be running on the node that the client will connect to.

The GUI server may be running on multiple nodes in the cluster at the same time, but the use of concurrent GUI sessions to perform multiple simultaneous administrative operations is not supported.

The REST API server and GUI server processes are managed by the `lifekeeper-api` and `lifekeeper-wmc` `systemd` services, respectively. The status of each service may be obtained using the `systemctl status` command:

```
[root@node1 ~]# systemctl status lifekeeper-api
```

```
● lifekeeper-api.service - SIOS LifeKeeper REST API Server Loaded: loaded (/usr/lib/systemd/system/lifekeeper-api.service; static; vendor preset: disabled)
Active: active (running)
```

```
[root@node1 ~]# systemctl status lifekeeper-wmc
```

```
● lifekeeper-wmc.service - LifeKeeper Web Management Console Service Loaded: loaded (/usr/lib/systemd/system/lifekeeper-wmc.service; enabled; vendor preset: disabled)
Active: active (running)
```

## 1.3.2. Port Settings

---

By default, the following TCP ports are used to connect to the GUI server and REST API server in the LKWMC.

Service	Default TCP Port
GUI Server	5110
REST API Server	5000

If the GUI server or REST API server needs to listen on a different port (for example if there is a conflict with a different process listening on the same port), please refer to the instructions in [Modifying the GUI Server Port](#) or [Modifying the REST API Server Port](#).

## 1.3.3. Firewall Rules

---

When using a firewall (or security group, network ACL, etc.) to restrict traffic within the network containing the LifeKeeper cluster nodes, the following types of traffic must be allowed in order for the LKWMC to function properly:

- For all LifeKeeper cluster nodes, allow traffic on TCP port 5000 (or the custom REST API server port, RESTAPI\_PORT, as described in [Modifying the REST API Server Port](#)) from each node where the GUI server will run. This allows the GUI server to communicate with the REST API server on all cluster nodes.
- If clients will connect directly to the node where the GUI server is running (e.g., if the node is located in a public subnet in a cloud environment), then on each node where the GUI server will run, allow traffic on TCP port 5110 (or the custom GUI server port, WEBUI\_PORT, described in [Modifying the GUI Server Port](#)) from each client system that will connect to the LKWMC.
- If the node where the GUI server is running is located in a private subnet or is otherwise not directly accessible by clients needing to connect to the LKWMC, it is possible to use port forwarding to connect to the LKWMC through a bastion host located in a public subnet. See [Connecting to the LKWMC Through a Bastion Host in a Cloud Environment](#) for more details.

## 1.3.4. Connecting to the LKWMC

---

The LKWMC may be accessed from a client PC by navigating to the following address in a web browser:

`https://:5110/lkgui/#/`



The host system running the GUI server must be reachable from the client PC via the network.

When using the default self-signed SSL certificates included with LifeKeeper, the browser may display a security warning stating that the identity of the remote server could not be verified. This warning can be avoided by generating and installing custom SSL certificates signed by a trusted certificate authority, as described in [Using Custom SSL Certificates](#). If the server's identity is trusted, then the security warning in the browser may be bypassed in order to access the LKWMC sign-in page.

If the client PC is not able to connect directly to the GUI server host (e.g., if the host system running the GUI server is located in a private subnet in a cloud environment), then a different method, such as a VPN or SSH port forwarding, must be used to connect to the GUI server.

## 1.3.4.1. Connecting to the LKWMC Through a Bastion Host in a Cloud Environment

The steps given below describe how to use SSH port forwarding through a bastion host located in a public subnet to allow client access to the GUI server running on a host located in a private subnet in a cloud environment.

### Prerequisites

1. LifeKeeper has been installed on a set of cluster nodes located in a private network in a cloud environment (i.e., the nodes do not have public IP addresses and are not accessible directly from the internet).
2. A bastion host server exists in a public subnet and can access the private subnet(s) containing the LifeKeeper cluster nodes.
3. The client PC can establish an SSH connection to the bastion host. Optionally, this connection may be established by using a private SSH key which is stored on the client PC.
4. The bastion host can establish SSH connections to all LifeKeeper cluster nodes in the private subnet(s).

### Steps

1. To enable SSH port forwarding, run the following command from the client PC terminal:

```
ssh [-i ] -L 8085::5110 -o ServerAliveInterval=60 @
```

This command forwards traffic on TCP port 8085 of the client PC to TCP port 5110 on the LifeKeeper node running the GUI server.



If the GUI server port number was changed from its default value of 5110 as described in [Modifying the GUI Server Port](#), use the updated port number in the SSH command above.



The -i option is only required when using an SSH key to establish the connection to the bastion host.

2. Connect to the GUI by entering the following address in the address bar of a web browser on the client PC:

`https://localhost:8085/lkgui/#`

## 1.3.5. Configuration Changes

---

This section describes how to change LKWMC configuration settings.

[Modifying the GUI Server Port](#)

[Modifying the REST API Server Port](#)

## 1.3.5.1. Modifying the GUI Server Port

By default, the GUI server listens for client connections on TCP port 5110. If necessary (e.g., if there is a conflict caused by a different process listening on the same port), this port number may be changed.

To change the port number, follow these steps:

1. On each LifeKeeper cluster node, open the `/etc/default/LifeKeeper` file using a text editor:

```
# vi /etc/default/LifeKeeper
```

and add an entry of the following form (or modify the entry if it already exists):

```
WEBUI_PORT=<GUI Server TCP Port>
```

For example, to set the GUI server port to TCP port 8000, the following entry would be added:

```
WEBUI_PORT=8000
```



It is important to make this modification to `/etc/default/LifeKeeper` on all nodes in the LifeKeeper cluster.

2. In order to apply the changes, the `systemd` service that manages the GUI server (`lifekeeper-wmc.service`) must be restarted on all LifeKeeper nodes where it is running:

```
# systemctl restart lifekeeper-wmc
```

## 1.3.5.2. Modifying the REST API Server Port

By default, the REST API server listens for client connections on TCP port 5000. If necessary (e.g., if there is a conflict caused by a different process listening on the same port), this port number may be changed.

To change the port number, follow these steps:

1. On each LifeKeeper cluster node, open the `/etc/default/LifeKeeper` file using a text editor:


```
# vi /etc/default/LifeKeeper
```

and add an entry of the following form (or modify the entry if it already exists):

```
RESTAPI_PORT=<REST API Server TCP Port>
```


where `<REST API Server TCP Port>` is a valid TCP port in the range 1-65535 which is not already in use. For example, to set the REST API server port to TCP port 5010, the following entry would be added:

```
RESTAPI_PORT=5010
```

 It is important to make this modification to `/etc/default/LifeKeeper` on all nodes in the LifeKeeper cluster.

2. In order to apply the changes, the systemd services that manage the REST API server (`lifekeeper-api.service`) and GUI server (`lifekeeper-wmc.service`) must both be restarted on all LifeKeeper nodes where they are running:

```
# systemctl restart lifekeeper-api lifekeeper-wmc
```

 It is important to restart the GUI server service so that its configuration is updated appropriately to send API requests to the correct port.



## 1.3.6. Using Custom SSL Certificates

---

The SSL certificates used to establish a secure HTTPS connection with the LKWMC can be replaced with custom user-provided certificates. Please refer to [Using Certificates with the LifeKeeper API](#) for more details about how to replace the existing certificates with custom certificates.

## 1.4. LKWMC GUI Operations and Layout

---

This section describes the LKWMC GUI Operations and Layout.

### [Signing In](#)

### [Header](#)

[Viewing the Online Manual](#)

[Viewing the GUI Version](#)

[Changing the Display Theme](#)

[Signing Out](#)

### [Server](#)

#### [Communication Path](#)

[Creating a Communication Path](#)

[Deleting a Communication Path](#)

[Viewing the Status of a Communication Path](#)

#### [Properties](#)

[Sign-in status](#)

### [Resource](#)

#### [Resource Tree](#)

[Creating a Resource](#)

[Extending a Resource Hierarchy](#)

[Unextending a Resource Hierarchy](#)

[Deleting a Resource Hierarchy](#)

[Bringing a Resource In Service](#)

[Taking a Resource Out of Service](#)

[Viewing Resource and Server Status](#)

[Creating or Deleting a Resource Dependency](#)

[Viewing and Editing Resource Properties](#)

## [Properties](#)

[General](#)

[Equivalencies](#)

[Relations](#)

[Recovery Kit-Specific Properties](#)

## 1.4.1. Signing In

---

In the LKWMC, PAM (Pluggable Authentication Module) authentication is used to authenticate users based on their membership in groups which define LifeKeeper user roles of varying privilege levels. Please refer to [Configuring GUI Users](#) for more details about configuring users with different LifeKeeper roles.

In order to sign in, enter the username and password of the desired operating system user and click the **Sign In** button.

## 1.4.2. Header

---

The LKWMC header, located at the top of the window, provides access to the following actions.

[Viewing the Online Manual](#)

[Viewing the GUI Version](#)

[Changing the Display Theme](#)

[Signing Out](#)

## 1.4.2.1. Viewing the Online Manual

---

Clicking the **Help** (?) icon in the header followed by clicking the **SIOS Technical Documentation link** will open the SIOS technical documentation website in a new tab.

## 1.4.2.2. Viewing the GUI Version

---

Clicking the **Help** (?) icon in the header followed by clicking the **About** link will display the version of the LKWMC.



The **About** window displays the version of the LKWMC on the system that the client is currently connected to. In order to view the version of the LifeKeeper Core software currently installed on a particular server in the LifeKeeper cluster, select **Properties** from the **Server** section in the left menu and set the desired server as the **Local Server**.

## 1.4.2.3. Changing the Display Theme

---

### Changing the Display Theme

The display theme used by the LKWMC can be modified by clicking the **Settings** (gear) icon in the header and selecting a new theme. The following color themes are available.

**Light** – Light color theme. This is the default display theme for the LKWMC.

**Dark** – Dark color theme.

**Tritanopia** – Theme for users with tritanopia (a type of blue-yellow color vision deficiency).

**Protanopia & Deuteranopia** – Theme for users with protanopia or deuteranopia (types of red-green color vision deficiency).



## 1.4.2.4. Signing Out

---

Clicking the **Account** (user) icon in the header followed by clicking **Sign out** will sign the current user out of the LKWMC.

## 1.4.3. Server

---

The **Server** section of the left menu allows users to view server-specific property and status information for each server in the LifeKeeper cluster, create or delete communication paths between servers, and view the authentication status for the current user on each server.

## 1.4.3.1. Communication Path

---


The **Communication Path** window can be used to create or delete a communication path, or to view the status of existing communication paths.

## 1.4.3.1.1. Creating a Communication Path

To create a new communication path, first click the **Communication Path** menu item in the Server section of the left menu, then click the **Operations** button and select **Create Communication Path**.

In the resulting **Create Communication Path** dialog, perform the following steps.

1. Enter the required communication path settings and click the **Confirm** button.
  - **Local Server** – Select the server where creation of the communication path should be performed. Servers with active LKWMC connections are listed as choices.
  - **Local IP Address** – Select the static IP address on the chosen **Local Server** which should be used for the communication path.
  - **Remote Server** – Select a remote server to act as the endpoint for the new communication path from the chosen **Local Server**. Servers with active LKWMC connections are listed as the initial choices.

 To add a communication path to a server that does not exist in the list, click **Add a New Server** to add the server.

1. Clicking **Add a New Server** will display a new text input field under the **Remote Server** choice list. Enter an IP address or hostname for the new server into the new text field and click **Confirm**.
2. When connecting to a server which does not have an active LKWMC connection, the user is prompted to sign into the LKWMC on the new server. Enter the username and password for the new server and click **Sign In** to authenticate. Once authentication is complete, the new server can be selected as a remote server for the communication path.
  - **Remote IP Address** – Select the static IP address on the chosen **Remote Server** which should be used for the communication path.
  - **Priority** – Enter the priority of the new communication path between **Local Server** and **Remote Server**. When multiple communication paths exist between a pair of servers, this value is used to determine the order in which the communication paths should be used for communication between the servers. Valid communication path priority values range from 1 to 99, with priority value 1 indicating highest priority and priority value 99 indicating lowest priority.
2. A confirmation window appears showing the selected communication path settings. Verify the chosen settings and click **Create** to create the communication path.
3. A window appears showing the results of the communication path creation. If creation of the

communication path fails, the user may click **Return to Creating a Communication Path** to retry the operation with new settings, or may click **Close** to abort the operation. If the communication path is created successfully, the user may click **Create Another Communication Path** to immediately create another communication path or may click Close to exit the dialog.

## 1.4.3.1.2. Deleting a Communication Path

---

To delete an existing communication path, first click the **Communication Path** menu item in the **Server** section of the left menu, then click the **Operations** button and select **Delete Communication Path**.

In the resulting Delete Communication Path dialog, perform the following steps.

1. Click the checkboxes corresponding to the communication paths that will be deleted, then click the **Confirm** button.
2. A confirmation window appears showing the list of communication paths to be deleted. Verify that the list contains the expected communication paths and click **Delete** to delete the communication paths.
3. A window appears showing the results of the operation and a list of communication paths which were successfully deleted. Click **Close** to exit the dialog.

## 1.4.3.1.3. Viewing the Status of a Communication Path

---

Clicking the **Communication Path** menu item in the **Server** section of the left menu will display a list of all communication paths along with status information.

Each line of the communication path table displays the following information.

- **Priority** – Communication path priority. The priority value determines the order in which the communication paths are used for communication between the servers. Valid communication path priority values range from 1 to 99, with priority value 1 indicating highest priority and priority value 99 indicating lowest priority.
- **Type** – Communication path type. The LKWMC only supports creation of TCP (TCP/IP) communication paths.
- **Status** – Current state of the communication path. The possible values are:
  - **ALIVE** – The communication path is functioning normally.
  - **DEAD** – The communication path is in a disconnected state. LifeKeeper heartbeats are not being sent or received over this communication path within the specified interval.
  - **UNKNOWN** – The state of the communication path could not be determined.

## 1.4.3.2. Properties

---

Clicking the **Properties** menu item in the **Server** section of the left menu will display information and property settings for the selected server.

For each server in the LifeKeeper cluster, the **Properties** page displays the following information.

- **General** – General status information and settings for the selected server.
- **Status** – Current state of the server. The possible values are:
  - **ALIVE** – The server is available.
  - **DEAD** – The server is unavailable.
  - **UNKNOWN** – The server state cannot be determined. The LifeKeeper REST API may not be available.
- **Shutdown Strategy** – The Shutdown Strategy is a LifeKeeper configuration option that governs whether or not resources are automatically brought in-service on a backup server when a server shuts down gracefully. The possible values are:
  - **Do not switchover resources** (default) – LifeKeeper will not bring resources in service on a backup server during an orderly shutdown.
  - **Switchover resources** – LifeKeeper will bring resources in service on a backup server during an orderly shutdown.
- **OS** – The type and version of the operating system on the server.
- **LifeKeeper version** – The version of the LifeKeeper Core software installed on the server.
- **Failover Strategy** – Settings which allow for additional customization of failover behavior for the selected server when LifeKeeper detects a resource or node failure.
  - **Confirm Failover** – When set to **ON** for a remote server, a confirmation process for executing a machine failover occurs on the remote server when a failover occurs due to a failure of the selected local server. This process involves checking the default action to take based on the CONFIRMSODEF variable setting, and checking how long to wait before taking that action based on the CONFIRMSOTO variable setting. When set to **OFF** for a remote server, failover confirmation does not occur on the remote server when the selected local server fails. See [Confirm Failover and Block Resource Failover Settings](#) for more information.
  - **Block Resource Failover** – When set to **ON** for a server, all resource transfers are blocked



when a resource failure is detected on that server. See [Confirm Failover and Block Resource Failover Settings](#) for more information.

## 1.4.3.3. Sign-in status

---

Clicking the **Sign-in status** menu item in the **Server** section of the left menu will display the authentication status for the current user on each server in the LifeKeeper cluster.

The **Sign-in status** page displays the following information for each server in the LifeKeeper cluster.

- **Server** – Server name.
- **User** – Name of the user being used for authentication on the server.
- **Permissions** – Permission level of the authenticated user on the server (**Guest**, **Operator**, or **Administrator**).
- **Expiry** – The date and time when the user's authentication token for the server will expire. The user will be required to sign into the server again when the token expires.
- **Authentication** – Current authentication status of the user. The possible values are:
  - **Authenticated** – The user has been successfully authenticated to make API requests to the server.
  - **Unauthenticated** – The user has not been successfully authenticated on the server. Successful authentication is required before API requests can be made to the server.
  - **Expired** – The user's authentication token has expired. Re-authentication is required before API requests can be made to the server.
  - **Signing in...** – The authentication process is in progress.

## 1.4.4. Resource

---

The **Resource** section of the left menu allows users to view the status of LifeKeeper-protected resources and perform operations related to resource administration.

## 1.4.4.1. Resource Tree

---

The Resource Tree provides a view of LifeKeeper-protected resources and resource hierarchies, along with the current status of each resource. This view also allows users to perform administrative operations, such as creating or deleting resources or changing the state of a resource.

Some important components of the Resource Tree layout are:

- The overall status of the LifeKeeper cluster is shown in the top left corner under the **Resource Hierarchies** heading. This status takes into account the states of all servers in the cluster as well as the states of all protected resources.
- The current status of each server is shown in the top row.
- The **Resource Hierarchies** column on the left shows all protected resources and parent-child dependency relationships.
- The cells in the table show the current states of protected resources on each server in the cluster.

The status of a protected resource on a particular server is reflected in the text, color, and icon shown in the corresponding cell. If a resource has not been extended to a particular server, the corresponding cell will appear blank.

The status of the LifeKeeper cluster can be refreshed in two ways:

- By clicking the **Refresh** icon in the top right section of the Resource Tree header, or
- By setting the status to automatically update every 15 seconds by enabling the **Auto-Update Toggle Button** in the top right section of the Resource Tree header.

The Resource Tree view may be customized by clicking the **Options** icon in the top right section of the Resource Tree header. The following options are available:

- **Display Name** – Select whether the resource ID or tag name is shown in the Resource Hierarchies column.
- **Sort** – Change the sort order of the Resource Tree.
- **Tree** – Fully expand or collapse all resource hierarchies in the tree.

Right-clicking on any cell in the table will display a context menu showing the actions available for the resource on the specified server.

## 1.4.4.1.1. Creating a Resource

---

To create a resource, click the **Operations** button in the **Resource Tree** and select **Create Resource Hierarchy** from the drop-down menu.

Follow the steps below to create a resource.

1. **Select Server and Recovery Kit** – Select the **Server** on which the resource should be created and the **Application Recovery Kit** that will be used to protect and administer the resource. The list contains all recognized recovery kits installed on the selected server.
2. **Required Settings** – Enter values for the requested settings that will be used when creating the resource.



The fields for some settings may be automatically populated with default values, and some default values are determined based on values of other fields. For such fields, the value may be edited once the default has been computed.

3. **Additional Settings** – Enter values for additional settings that will be used when creating the resource. Default values are automatically populated for all of these additional settings. The default values can be accepted by clicking Next.
4. **Confirm** – Review the specified resource creation settings, then click Execute to create the resource.
5. **Result** – The results of the resource creation process are shown. Once the resource has been successfully created, the user may click Create Another Resource to create another resource, Extend Resource Hierarchy to extend the newly-created resource hierarchy to a standby server, or Close to exit the dialog.

## 1.4.4.1.2. Extending a Resource Hierarchy

There are three ways to extend an existing resource hierarchy to a standby server.

- Click the **Operations** button in the **Resource Tree** and select **Extend Resource Hierarchy** from the drop-down menu.
- In the **Resource Tree**, right-click the resource to extend and select **Extend Resource Hierarchy** from the context menu. This menu option is only available on servers where the resource hierarchy is in-service.
- Click the **Extend Resource Hierarchy** button at the end of the Create Resource Hierarchy workflow.

Follow the steps below to extend a resource hierarchy.

1. **Select a server and a resource to extend** – Select the **Template Server** where the template resource hierarchy exists, the **Resource** on the template server whose hierarchy will be extended, and the **Target Server** that the resource hierarchy will be extended to. If desired, the priorities of the template and target servers for the selected resource hierarchy may be modified. Once all settings have been specified, click **Next**. After clicking **Next**, pre-extend validation is performed and the dialog will proceed to the next step if no issues are detected.
2. **Additional Settings** – Enter values for the requested settings that will be used when extending the resource hierarchy. Settings for dependent resources are displayed in a collapsed accordion view. If necessary, expand the settings for each resource in the hierarchy and modify the values as needed.

✿ The fields for some settings may be automatically populated with default values, and some default values are determined based on values of other fields. For such fields, the value may be edited once the default has been computed.

3. **Confirm** – Review the specified resource hierarchy extension settings, then click **Execute** to extend the resource hierarchy.
4. **Result** – The results of the resource hierarchy extension process are shown. Once the hierarchy has been successfully extended to the target server, the user may click **Close** to exit the dialog. In a cluster environment with more than two nodes, the user may instead click **Extend This Resource to Another Server** to extend the same resource hierarchy to an additional standby server.

## 1.4.4.1.3. Unextending a Resource Hierarchy

---

There are two ways to unextend a resource hierarchy from a server.

- Click the **Operations** button in the **Resource Tree** and select **Unextend Resource Hierarchy** from the drop-down menu.
- In the **Resource Tree** , right-click the resource to unextend and select **Unextend Resource Hierarchy** from the context menu. This menu option is only available on servers where the resource hierarchy is out of service.

Both of these options will bring up a confirmation dialog to verify that the expected resource hierarchy will be unextended. Clicking **Unextend** in the confirmation dialog will unextend the chosen resource hierarchy from the selected server.

## 1.4.4.1.4. Deleting a Resource Hierarchy

---

There are two ways to delete a resource hierarchy.

- Click the **Operations** button in the **Resource Tree** and select **Delete Resource Hierarchy** from the drop-down menu.
- In the **Resource Tree** , right-click the resource to delete and select **Delete Resource Hierarchy** from the context menu.



Deleting a resource hierarchy while it is still in-service may cause services to remain running unexpectedly and may block access to some system resources. To prevent issues like this, take the entire resource hierarchy out of service on all nodes prior to deleting it.

Both of these options will bring up a confirmation dialog to verify that the expected resource hierarchy will be deleted. Clicking **Delete** in the confirmation dialog will delete the resource hierarchy.



## 1.4.4.1.5. Bringing a Resource In Service

---

There are two ways to bring a resource hierarchy in service.

- Click the **Operations** button in the **Resource Tree** and select **In Service** from the drop-down menu.
- In the **Resource Tree** , right-click the resource to bring in-service and select **In Service** from the context menu.

Both of these options will bring up a confirmation dialog to verify that the expected resource will be brought in-service. Clicking **In Service** in the confirmation dialog will bring the resource and all child resources in-service.

## 1.4.4.1.6. Taking a Resource Out of Service

---

There are two ways to take a resource hierarchy out of service.

- Click the **Operations** button in the **Resource Tree** and select **Out of Service** from the drop-down menu.
- In the **Resource Tree**, right-click the resource to bring in-service and select **Out of Service** from the context menu.

Both of these options will bring up a confirmation dialog to verify that the expected resource will be taken out of service. Clicking **Out of Service** in the confirmation dialog will take the resource and all parent resources out of service.

## 1.4.4.1.7. Viewing Resource and Server Status

The status of each server and resource may be viewed from the **Resource Tree**.

The status of a protected resource on a particular server is reflected in the text, color, and icon shown in the corresponding cell. If a resource has not been extended to a particular server, the corresponding cell will appear blank.

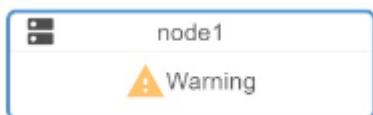
### Server States

The possible server states are shown below.

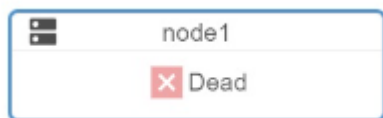
- **Alive** – LifeKeeper is functioning normally. All communication paths from this server to remote servers are ALIVE.



- **Warning** – Client has a valid connection to the server. One or more communication paths from this server to a given remote server are marked as DEAD, or no redundant communication paths exist from this server to a given remote server.



- **Dead** – Reported as DEAD by other servers in the cluster.



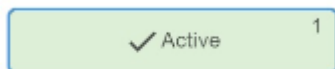
- **Unknown** – Network connection was lost. Last known LifeKeeper server state is ALIVE or the REST API server is not running on the server.



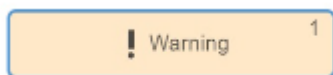
## Resource States

The possible resource states are shown below. In addition to these generic resource states, some application recovery kits also provide resource-specific custom states (such as the “In-Sync” state for a DataKeeper resource indicating that replication is in-sync to a particular mirror target).

- **Active** – The resource is operational. LifeKeeper local recovery is operating normally. LifeKeeper inter-server recovery and failure recovery is operational.



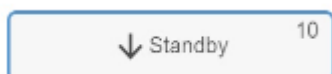
- **Warning** – The resource is operational. However, no local recovery or failure recovery will occur because the LifeKeeper health check is not currently being performed for the resource.



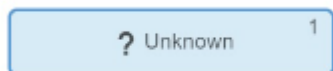
- **Failed** – The resource has failed and has been taken out of service. Recovery has not been completed or has failed. LifeKeeper alarming is not operational for this resource.



- **Standby** – The resource is out of service but is available to take over an equivalent resource from another server.



- **Unknown** – The state of the resource could not be determined.



- **Illegal** – This state appears in situations where no state has been set for a resource instance. Under normal circumstances, this is a transient state which may appear while the resource is transitioning to another state. This state will be shown if switchover occurs before all LifeKeeper information tables have been updated (for example, when LifeKeeper is first started up).



## Recovery Kit-Specific Resource States

### - DataKeeper Resource States

- **Source** – DataKeeper resource primary server, running successfully.
- **Target** – DataKeeper resource replication target. Available to take over the resource from the primary server.
- **Out Of Sync** – Mirror is stopped and data is not synchronized.
- **Wait for Previous Source** – Mirror is stopped and data is not synchronized. DataKeeper resource is waiting to be synchronized with the previous source.
- **Wait to Resync** – Data is not synchronized and the mirror is pending.  
To avoid replicating corrupt or inconsistent data to targets, LifeKeeper can wait for the parent resource to be in-service before replicating data.  
Refer to [Verify Data Before Resync](#) for more information.
- **Resyncing** – Mirror is running but DataKeeper resource is resyncing.
- **Paused** – Mirror is paused.
- **Kernel Module Failure** – Failed to load a required kernel module. The resource is not functioning properly.

### - SAP Recovery Kit Resource States

- **Attention needed** – The SAP resource has failed or is in a warning state. If it has failed and automatic recovery is enabled (Protection Level Full or Standard), then LifeKeeper will try to automatically recover the resource. Right-click on the SAP resource and choose Properties. This will show which SAP process is currently in a warning state. An SAP process state of Yellow may be normal (for example if the process is still initializing), but may signify that it is running slowly or encountering performance bottlenecks.

### - SAP HANA Recovery Kit Resource States

- **Active** – Sapstartsrv and the HDB instance are running properly.
- **Active – sapstartsrv Failure** – Sapstartsrv is not running. The HDB instance status is unknown since sapstartsrv is used to check the status of HDB.
- **Active – HSR Disabled** – Sapstartsrv and HDB instance are running properly. The replication mode is 'none', indicating that SAP HANA Replication is disabled.

- **Active – HDB Stopped** – Sapstartsrv is running. The HDB instance is not running properly.
- **Active – Secondary** – Sapstartsrv and the HDB instance are running properly but the active node is registered as a secondary replication site. In normal operation, the active node is expected to be the primary replication site.
- **Active – Unknown Repl Mode** – Sapstartsrv and HDB instance are running properly. The replication mode cannot be determined using 'hdbnsutil -sr\_state' or it returns an unsupported mode.
- **Active – Suspended** – The database is suspended on the server where the SAP HANA resource is currently in-service. This is an unexpected state, and typically occurs when a user has performed a “takeover with handshake” outside of LifeKeeper.
- **Standby** – This is the normal status for a resource that is out-of-service when an equivalent resource is not in-service on another node.
- **Standby – HDB Running** – Sapstartsrv and the HDB instance are running on the standby (OSU) node. There is either no active (ISP) node, multiple active (ISP) nodes, or quickCheck has not determined the HSR status on the active node.
- **Standby – In Sync** – Sapstartsrv and the HDB instance are running with HSR configured and reporting in-sync.
- **Standby – sapstartsrv Failure** – Sapstartsrv is not running. There is an active (ISP) node. The HDB instance status is unknown since sapstartsrv is used to check the status of HDB.
- **Standby – Unknown Repl Mode** – Sapstartsrv and the HDB instance are running. There is an active (ISP) node. The replication mode cannot be determined using 'hdbnsutil -sr\_state'.
- **Standby – HSR Disabled** – Sapstartsrv and the HDB instance are running. There is an active (ISP) node. Either the replication mode is 'none' according to 'hdbnsutil -sr\_state', or the HANA utility systemReplicationStatus.py returned '10'.
- **Standby – HSR Error** – Sapstartsrv and the HDB instance are running. There is an active (ISP) node. The HANA utility systemReplicationStatus.py returned '11'.
- **Standby – Initializing** – Sapstartsrv and the HDB instance are running. There is an active (ISP) node. HANA utility systemReplicationStatus.py returned '13'.
- **Standby – Syncing** – Sapstartsrv and the HDB instance are running. There is an active (ISP) node. The HANA utility systemReplicationStatus.py returned '14'.
- **Standby – HDB Stopped** – Sapstartsrv is running. The HDB instance is not running. There is

an active (ISP) node.

- **Standby – Primary** – Sapstartsrv and the HDB instance are running properly, but the standby node is registered as a primary replication site. In normal operation, the standby node is expected to be a secondary replication site.

- **Standby – Unknown HSR Status** – Sapstartsrv and the HDB instance are running. There is an active (ISP) node. The HANA utility systemReplicationStatus.py returned '12'.

- **Standby – Unexpected HSR State** – Sapstartsrv and the HDB instance are running. There is an active (ISP) node. The HANA utility systemReplicationStatus.py returned a value that was not in the expected range of 10 to 15.

- **Standby – Suspended** – The database is suspended on a server where the SAP HANA resource is currently out of service.

## 1.4.4.1.8. Creating or Deleting a Resource Dependency

---

There are two ways to create or delete a dependency between two resources.

- Click the **Operations** button in the **Resource Tree** and select **Create Dependency** or **Delete Dependency** from the drop-down menu.
- In the **Resource Tree** , right-click the parent resource and select **Create Dependency** or **Delete Dependency** from the context menu.

Click **Confirm** to execute the action.



## 1.4.4.1.9. Viewing and Editing Resource Properties

---

There are two ways to view or edit the properties for a resource.

- Click the **Operations** button in the **Resource Tree** and select **Properties** or **Edit Properties** from the drop-down menu.
- In the **Resource Tree** , right-click the resource and select **Properties** or **Edit Properties** from the context menu.

Clicking **Properties** will display the **Properties** page for a specified resource and server. Clicking **Edit Properties** will display the Edit Properties dialog on the same screen.

To edit the properties for a resource on a server, perform the following steps.

1. **Select the Server and Resource** – If Edit Properties is selected from the Operations menu in the Resource Tree, then a resource and server must be chosen. Select the resource to edit the properties of and the server where the resource exists. This dialog is not shown if Edit Properties is selected from the resource context menu (by right-clicking on the resource in the Resource Tree).
2. **Edit Properties** – Modify the property value(s) as desired. Available properties vary depending on resource type.
3. **Result** – Click **Execute** to apply the changes to the resource property values.

## 1.4.4.2. Properties

---

The current property settings for the resources on each server in the cluster may be viewed in three ways.

- Click **Properties** in the **Resource** section of the left menu. Specify the desired server and resource by selecting values in the **Local Server** and **Resource** drop-down boxes.
- Click the **Operations** button in the **Resource Tree** and select **Properties** from the drop-down menu. Specify the desired server and resource by selecting values in the **Local Server** and **Resource** drop-down boxes.
- In the **Resource Tree** , right-click the resource and select **Properties** from the context menu.

The following sections and properties are shown on the Properties page for a resource.

## 1.4.4.2.1. General

---

**ID** – A character string associated with a resource instance, unique among all instances of the resource type, that identifies some internal characteristics of the resource instance to the application software associated with it.

- **Status** – Current state of the resource instance. Refer to Viewing Resource and Server Status for descriptions of each possible resource state.

- **Reason** – If present, describes the reason that the resource is in its current state. In other words, this is the reason for its most recent state change. For example, a resource on node2 may be in the OSU state because the corresponding equivalent shared resource on machine node1 is in ISP or ISU state. Shared resources may only be active on one of the clustered systems at a time.

- **Application Type** – Internal LifeKeeper resource application type identifier.

- **Resource Type** – Internal LifeKeeper resource type identifier.

- **Switchback Type** – The setting that governs the recovery behavior of the server where the resource was in service when it failed. If the setting is intelligent, the server acts as a possible backup for the given resource. If the setting is automatic, the server actively attempts to re-acquire the resource, providing the following conditions are met:

- The resource hierarchy must have been in service on the server when it left the cluster.

- If it is in service at all, then the resource must currently be in service on a server with a lower priority.



Checks for automatic switchback are made only when LifeKeeper starts or when a new server is added to the cluster. They are not performed during normal cluster operation.

## 1.4.4.2.2. Equivalencies

---

- **Server** – The name of the server on which the resource has a defined equivalency.
- **Priority** – The failover priority value of the standby server hosting an equivalent copy of the resource. This value is editable if the user has Administrator permission. The priority values may be edited by clicking the gear icon in the top-right corner of the Equivalencies table. After modifying the server priority values, click **Execute** to apply the changes.
- **Tag** – The tag name of this resource on the equivalent server.

## 1.4.4.2.3. Relations

---

- **Parent Resource(s)** – Identifies the tag names of the resources that are directly dependent on this resource.
- **Child Resource(s)** – Identifies the tag names of all resources that this resource depends on.

## 1.4.4.2.4. Recovery Kit-Specific Properties

---

This section displays recovery kit-specific settings which vary by resource type. Hovering over the information icon next to each property name displays a short description of the property.

## 1.4.5. Logs

---

Clicking **Logs** in the left menu will display the LifeKeeper logs for each server in the cluster. The most recent log messages for the specified **Local Server** are shown in chronological order, up to the specified maximum number of lines.

- **Local Server** – Select the server from which to obtain LifeKeeper log messages.
- Entering text in the search box filters the results to only display log messages that contain the search term.
- Changing the value of **Lines** in the log table will limit the maximum number of log messages that appear in the results.
- Clicking the **Download** icon will download the currently displayed log messages to the system running the LKWMC.

## 1.4.6. Licensing

---

Clicking **Licenses** in the left menu will display a list of licenses that are installed on each server in the cluster.

- **Local Server** – Select the server from which to obtain installed license information.
- The following fields are shown for each license.

**Product** – The product that the license applies to.

- **Type** – The type of the license. Valid types include:
  - **UNLICENSED** – The Core Application license is used.
  - **EVAL** – An evaluation license is used.
  - **PERMANENT** – An optional Recovery Kit license is used.
  - **NEEDED** – An optional Recovery Kit license is not used.
- **Expiry** – Shows the expiration date of the license.
- The license table sorts licenses based on expiration date. The license displayed first in the list will be the first to expire.
- Licenses which have expired or are nearing expiration are highlighted in the license information table.



## 1.5. Known Issues and Restrictions

---

### SIOS DataKeeper for Linux

DataKeeper asynchronous mode is not supported in the current version of the LKWMC. The asynchronous mod will be supported in a future release.

### Logical Volume Manager (LVM) Recovery Kit

LVM volume group reconfiguration is not supported in the current version of the LKWMC. This feature will be supported in a future release.

### Quick Service Protection (QSP) Recovery Kit

Editing the properties for a Quick Service Protection (QSP) Recovery Kit resource is not supported in the current version of the LKWMC. This feature will be supported in a future release.

### Multipath Recovery Kit (HDLM)

Operations involving the Hitachi Dynamic Link Manager (HDLM) Recovery Kit are not supported in the current version of the LKWMC. This recovery kit will be supported in a future release.

### SAP ASE (Sybase) Recovery Kit

Operations involving the SAP ASE (Sybase) Recovery Kit are not supported in the current version of the LKWMC. This recovery kit will be supported in a future release.

### Single Server Protection for Linux

Operations using LifeKeeper Single Server Protection for Linux are not supported in the current version of the LKWMC. These will be supported in a future release.