



LifeKeeper for Windows
Microsoft SQL Server Recovery Kit
Administration Guide

January 2012

This document and the information herein is the property of SIOS Technology Corp. (previously known as SteelEye® Technology, Inc.) and all unauthorized use and reproduction is prohibited. SIOS Technology Corp. makes no warranties with respect to the contents of this document and reserves the right to revise this publication and make changes to the products described herein without prior notification. It is the policy of SIOS Technology Corp. to improve products as new technology, components and software become available. SIOS Technology Corp., therefore, reserves the right to change specifications without prior notice.

LifeKeeper, SteelEye and SteelEye DataKeeper are registered trademarks of SIOS Technology Corp.

Other brand and product names used herein are for identification purposes only and may be trademarks of their respective companies.

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

Address correspondence to:
ip@us.sios.com

Copyright © 2012
By SIOS Technology Corp.
San Mateo, CA U.S.A.
All rights reserved

Table of Contents

Chapter 1: Introduction	1
LifeKeeper Microsoft SQL Server	1
SQL Server Services	1
SQL 2000	1
SQL 2005/2008/2008 R2	1
Recovery Kit Requirements	2
Chapter 2: SQL Server Installation	3
Recovery Kit Installation	3
Kit Removal	3
Installing and Configuring SQL Server with LifeKeeper	3
Before Installing SQL Server	3
Installation - Shared Storage Systems	4
On the Primary Server	4
On the Backup Server	4
On the Primary Server	4
Install the SQL Server Software	4
SQL 2000:	4
SQL 2005 or SQL 2008:	5
Installation - Replicated Storage Systems	5
On the Primary Server	5
On the Backup Server	6
SQL 2000:	6
SQL 2005 or SQL 2008:	6
On the Primary Server	7
Additional Setup Tasks for Extended Configurations	7
Creating the SQL Hierarchy	7

Extending a SQL Hierarchy	9
Unextending a SQL Hierarchy	9
Deleting a SQL Hierarchy	10
Chapter 3: SQL Server Configuration Considerations	11
SQL Configuration	11
Failover	12
Multiple SQL Instances	12
Managing a SQL Server Configuration	13
Process Config Menu	14
Database Config Menu	15
Admin Actions Menu	15
Manage User Menu	15
Testing Your SQL Resource Hierarchy	16
Chapter 4: SQL Server Hierarchy Administration	19
Access Via Protected Communication Paths	19
User Resource Name For Remote Access	19
Reserve Volumes For Exclusive SQL Use	19
Understand Manual Switchover Limitations in SQL Server Environment	19
Running Microsoft SQL Management Tools	19
Start and Stop SQL Server Only Through LifeKeeper	20
Adding Microsoft SQL Server Volumes	20
Recovering From Databases in Suspect State After a Switchover	20
Pausing Microsoft SQL Server (MSSQLServer)	21
Configuring SQL Server to Connect Using the Switchable IP Address	21
Maintaining SQL Server Login and Passwords	21
Monitoring Your SQL Hierarchy	22
Chapter 5: Troubleshooting	23
Create and Extend Fail if Master or User Databases on System Drive	23
Symptom:	23
Suggested Actions:	23

Extend of a SQL Resource Fails (SQL 2005 only)	23
Symptom:	23
Suggested Actions:	24
SQLAgent Service Fails to Start Sometimes for Named Instances	24
Symptom:	24
Suggested Actions:	24
Connecting ODBC Clients to Named Instances of SQL Server	25
Symptom:	25
Suggested Action:	25

Chapter 1: Introduction

LifeKeeper Microsoft SQL Server

The LifeKeeper Microsoft SQL Server Recovery Kit software lets you tie the data integrity of Microsoft SQL-based databases to the increased availability provided by LifeKeeper for Windows.

The LifeKeeper GUI allows you to easily create a SQL resource hierarchy. LifeKeeper can then protect all of the disk resources used by the SQL Server instance, as well as the IP socket resources used to access the database.

Important Note: This kit is incompatible with the following SQL features: SQL Replication (Snapshot, Merge and Transactional), SQL Log Shipping and SQL 2005 Database Mirroring.

SQL Server Services

The LifeKeeper Microsoft SQL Server Recovery Kit will monitor and protect the following services:

SQL 2000

Core Services	Optional Services
SQL Server (MSSQLSERVER)	SQL Server Agent
	Distributed Transaction Coordinator
	Microsoft Search

SQL 2005/2008/2008 R2

Core Services	Optional Services
SQL Server (MSSQLSERVER)	SQL Server Agent
	SQL Server Reporting Services
	Distributed Transaction Coordinator
	SQL Server Browser
	SQL Server VSS Writer

All data files are stored on shared or replicated volumes. Thus, upon detecting a failure, LifeKeeper switches the database along with its associated volumes and IP socket resources to a backup

server. Once LifeKeeper switches all dependent resources to the backup server, it starts the Microsoft SQL service and any protected optional services.

Recovery Kit Requirements

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

Operating System software. LifeKeeper supports the following versions of Windows operating systems:

- Windows Server 2008 Standard, Enterprise, Data Center Editions (32- and 64-bit versions)
- Windows Server 2003 Standard, Enterprise, Data Center, Web Editions
- Windows Server 2003 R2 Editions
- **LifeKeeper software.** You must install the same version of LifeKeeper for Windows on all servers in the cluster.
- **SteelEye DataKeeper for Windows (optional).** If you plan to use Microsoft SQL Server with replicated volumes rather than shared storage, you should install the SteelEye DataKeeper software on each server.
- **Microsoft SQL Server RDBMS software.** The kit is compatible with Microsoft SQL 2000 (8.0), Microsoft SQL 2005 (9.0) and Microsoft SQL 2008, R1 and R2, all versions. However, the same version of Microsoft SQL Server must be installed on all systems in the cluster.
- **Communication protocol.** TCP/IP is strongly recommended by Microsoft for use in a clustered environment. Although LifeKeeper supports LAN Manager, this document will assume you are using TCP/IP, and will refer to switchable IP resources (rather than LAN Manager resources) in its configuration instructions.

Consult your LifeKeeper sales representative for release and ordering information.

Chapter 2: SQL Server Installation

Proper operation of the LifeKeeper Microsoft SQL Server Recovery Kit depends upon correct setup of the hardware and software.

Before continuing, please preview the Hierarchy Administration section of this guide. This section provides general guidelines, configuration details, and troubleshooting hints to help you administer Microsoft SQL Server in a LifeKeeper environment.

Recovery Kit Installation

The LifeKeeper Microsoft SQL Server Recovery Kit is distributed via ftp download. Installation is simple and quick using InstallShield to provide a standard installation interface.

Before installing the LifeKeeper Microsoft SQL Server Recovery Kit software, be sure you are familiar with the [product prerequisites](#). A LifeKeeper Microsoft SQL Server Recovery Kit license key must be installed in order to protect a SQL resource using LifeKeeper.

Kit Removal

To remove the LifeKeeper Microsoft SQL Server Recovery Kit software, choose **LifeKeeper Microsoft SQL Server Recovery Kit for LifeKeeper v6 Update 2** in the **Add/Remove Programs** or **Programs and Features** applet in the control panel.

CAUTION: Be sure there are no SQL instances or resources in service when the kit is removed. Once the kit is removed, these resources will be unusable. All SQL hierarchies should be deleted before the kit is removed.

Installing and Configuring SQL Server with LifeKeeper

Proper operation of the LifeKeeper Microsoft SQL Server Recovery Kit depends upon correct setup of the hardware and software.

This section provides general guidelines, configuration details and troubleshooting hints to help you administer Microsoft SQL Server in a LifeKeeper environment. Please remember to review the Hierarchy Administration tasks.

Before Installing SQL Server

Before you install the SQL Server software, the servers and storage must be configured and LifeKeeper must be installed on each server in the cluster.

Installation - Shared Storage Systems

On the Primary Server

1. Power down the backup server so that there is no chance of simultaneous access to your shared storage.
2. Use the **Windows Disk Management** tool to configure your disk resources and define the shared volumes that you want to use. (Be sure the volume size is adequate.)
3. It is recommended that you use **Windows Explorer** to unshare from the network all volumes to be used by the SQL Server Instance.
4. Configure your networking to support the LifeKeeper TCP/IP comm path(s) and, if applicable, the switchable IP address.
5. Install the LifeKeeper Core software on a local disk, followed by the LifeKeeper SQL Server Recovery Kit.

On the Backup Server

1. Bring up the backup server and use the **Disk Management** utility to assign the same drive letter to the shared volume as assigned on the primary server.
2. Install the LifeKeeper Core software on a local disk, followed by the LifeKeeper SQL Server Recovery Kit.

On the Primary Server

Now that you have LifeKeeper installed on both servers, go back to the primary server and do the following:

1. In LifeKeeper, create comm paths between the primary and backup servers.
2. In LifeKeeper, create your volume resource and IP communication resource and extend them to the backup server. Later when you create your SQL Server resource hierarchy, LifeKeeper will automatically bring these resources into the hierarchy as dependencies.

Install the SQL Server Software

1. If using shared volumes, bring the volume resource hierarchy In Service on the backup server using the LifeKeeper GUI.
2. On the backup server, install Microsoft SQL Server using the following guidelines:

SQL 2000:

- In the **Computer Name** dialog, choose **Local Computer**.
- In the **Setup Type** dialog under **Destination Folder**, specify a folder on the local disk for **Program Files** and a shared volume (protected by LifeKeeper) for

Data Files.

- In the **Authentication Mode** dialog, select “**Mixed Mode**” and enter a non-blank password for the SA account. The passwords MUST be the same on all servers in the cluster.

SQL 2005 or SQL 2008:

- In the **Components to Install** dialog, select the components for your installation. Click on the **Advanced** button to go to the **Feature Selection** dialog. On the **Feature Selection** page, change the installation path for the **Data Files** under **Database Services** to the shared volume (protected by LifeKeeper).
- In the **Authentication Mode** dialog, select “**Mixed Mode**”, and enter a non-blank password for the SA account. The passwords MUST be the same on all servers in the cluster.

When the installation is complete, use **Microsoft SQL Enterprise Manager** (SQL 2000) or **Microsoft SQL Server Configuration Manager** (SQL 2005 or SQL 2008) to verify that SQL Server can start properly on the backup server. Stop all Microsoft SQL Services on the backup server.

For shared volumes, do the following steps.

1. Bring the volume resource hierarchy In Service on the primary server.
2. On the primary server, open **Explorer** and access the drive associated with the shared volume.
3. Delete the directory where you previously installed the SQL data files. (You will re-install them in the next step).
4. Install Microsoft SQL Server on the primary server EXACTLY as you did on the backup server (program files on the local disk and data files on the shared volume).

When the installation is complete, use **Microsoft SQL Enterprise Manager** (SQL 2000) or **Microsoft SQL Server Configuration Manager** (SQL 2005 or SQL 2008) to verify that SQL Server can start properly on the primary server.

Installation - Replicated Storage Systems

On the Primary Server

1. Use the **Windows Disk Management** tool to configure your disk resources and define the replicated volumes that you want to use. (Be sure the volume size is adequate.)
2. It is recommended that you use **Windows Explorer** to unshare from the network all volumes to be used by SQL Server.
3. Configure your networking to support the LifeKeeper TCP/IP comm path(s) and, if applicable, the switchable IP address.

On the Backup Server

4. Install the LifeKeeper Core software on a local disk followed by the LifeKeeper SQL Server Recovery Kit.
5. Install the SteelEye DataKeeper software to the local disk now. Refer to [Installation](#) for details.
6. Using the LifeKeeper GUI, create comm paths between the primary and backup servers.
7. In LifeKeeper, create your IP communication resource and extend them to the backup server. Later when you create your SQL Server resource hierarchy, LifeKeeper will automatically bring the resource into the hierarchy as a dependency.

Note: When the SQL Server Hierarchy is created, the SteelEye DataKeeper resource will automatically be created and brought into the SQL Server resource hierarchy as a dependency.

On the Backup Server

1. Bring the volume resource hierarchy In Service using the **LifeKeeper GUI**.
2. Install Microsoft SQL Server using the following guidelines:

SQL 2000:

- In the **Computer Name** dialog, choose **Local Computer**.
- In the **Setup Type** dialog under **Destination Folder**, specify a folder on the local disk for **Program Files** and a replicated volume (protected by LifeKeeper) for **Data Files**.
- In the **Authentication Mode** dialog, select "**Mixed Mode**", and enter a non-blank password for the SA account. The passwords **MUST** be the same on all servers in the cluster.

SQL 2005 or SQL 2008:

- In the **Components to Install** dialog, select the components for your installation. Click on the **Advanced** button to go to the **Feature Selection** dialog. On the **Feature Selection** page, change the installation path for the **Data Files** under **Database Services** to the replicated volume (protected by LifeKeeper).
 - In the **Authentication Mode** dialog, select "**Mixed Mode**", and enter a non-blank password for the SA account. The passwords **MUST** be the same on all servers in the cluster.
3. When installation is complete, use **Microsoft SQL Enterprise Manager** (SQL 2000) or **Microsoft SQL Server Configuration Manager** (SQL 2005 or SQL 2008) to verify that SQL Server can start properly on the backup server. Stop all Microsoft SQL Services on the backup server. **Note:** For replicated volumes, you may wish to move the *SQL tempdb* database to a volume that is not protected by LifeKeeper to improve performance.
 4. Install Microsoft SQL Server on the primary server **EXACTLY** as you did on the backup server

(program files on the local disk and data files on the replicated volume).

- When installation is complete, use **Microsoft SQL Enterprise Manager** (SQL 2000) or **Microsoft SQL Server Configuration Manager** (SQL 2005 or SQL 2008) to verify that SQL Server can start properly on the primary server.

Note: If the data files are installed to a replicated volume, you may wish to move the *SQL tempdb* database to a volume that is not protected by LifeKeeper to improve performance.

On the Primary Server

- Bring the **Communication resource** into service on the primary server.
- Start the **SQL Server Services** on the primary server.
- Create the **SQL Server hierarchy** on the primary server and extend it to the backup server. During the **Extend Volume Resource** process, choose **Created Replicated Mirror** then select **Next** to complete the wizard and finish the configuration. See [Creating the SQL Hierarchy](#) for details.

Test the new **SQL Server hierarchy** by performing a manual failover.

Additional Setup Tasks for Extended Configurations

If your configuration uses a shared storage device or you are using SteelEye DataKeeper, you may choose a configuration that will be extended to a third (or more) server(s).

- Configure two systems following the steps given in [Installing and Configuring SQL Server with LifeKeeper](#).
- Switch your protected volumes to the third server.
- Install the Microsoft SQL Server software on the local drive and the master database on the same shared/replicated volume as used by the other servers. This will permit you to extend the hierarchy and utilize LifeKeeper's cascading feature.

Creating the SQL Hierarchy

After you have completed the necessary setup tasks outlined in [Installing LifeKeeper](#), use the following steps to define the SQL Server hierarchy to protect your database(s).

Important	If you have an existing SQL database installed, you must close any client applications (local or remote) that are accessing the SQL database prior to completing this procedure. During the automated move operation, the database will need to be started and restarted numerous times and running applications may interfere with the commands following each service action. Restart the applications once this procedure has been completed.
-----------	--

- From the LifeKeeper GUI menu, select **Edit** then **Server**. From the menu, select **Create Resource Hierarchy**.
- The **Create Protected Application** dialog box will display. Select the **Primary** and **Backup**

servers from the pull-down list. Select **Next** to continue.

3. The dialog box will appear with a drop down list box displaying all recognized recovery kits installed within the cluster. Select **MS SQL Server** and click **Next**.
4. 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 helpful should you encounter an error requiring you to correct previously entered information. You may click **Cancel** at any time to cancel the entire creation process.

Field	Tips
Select Microsoft SQL Server Instance	Select the instance of Microsoft SQL Server you wish to place under LifeKeeper protection. LifeKeeper will read the configuration data for this instance and pull the associated volumes into the hierarchy.
Enter Microsoft SQL Administrative User Name	Enter the administrative user name that is used for Microsoft SQL on this server. This user account must include SA permissions to the master database.
Enter Password	Enter the administrative password for the user account just entered.
Verify Current Locations	Displays the current location of database files. If any of the detected files for the specified instance to be protected by LifeKeeper are located on the System Drive (c:), they will be highlighted in the table displayed on the screen.
The following fields only appear if an existing database needs to be relocated.	
Select Destination for Database Relocation	When database migration is required, specify the volume destination to relocate affected databases.
Verify the Move Operation	Verify the location and confirm the intent to move the specified databases.
Relocating the Databases	An action window displays showing the progress of the databases being relocated during the move operation.
Select Optional Services for Protection	Select optional SQL services to be protected in this hierarchy. The list includes only those services eligible for LifeKeeper protection.
Protected IP Address	Select an IP address to protect with this instance. IP Address is not required if only named pipes are used (though this is NOT recommended).
Named Pipe Alias	Named Pipe Alias

Field	Tips
Microsoft SQL Server Resource Name	Enter a unique tag name, or you can accept the default tag name offered by LifeKeeper.

- After you click **Create**, the **Wizard** will create your SQL resource. LifeKeeper will validate the data entered. If LifeKeeper detects a problem, an error message will appear in the information box.
- Another information box will appear indicating that you have successfully created a SQL resource hierarchy, and you must **Extend** that hierarchy to another server in your cluster in order to achieve failover protection. Click **Next**.
- After you click **Continue**, LifeKeeper will launch the **Pre-Extend Wizard**.

Extending a SQL Hierarchy

This operation can be started from the **Edit** menu or initiated automatically upon completing the **Create Resource Hierarchy** option, in which case you should refer to Step 2 below.

- From the **Edit** menu, select **Resource** then **Extend Resource Hierarchy**. The **Pre-Extend Wizard** appears. If you are unfamiliar with the **Extend** operation, click **Next**.
- 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.
- After receiving the message that the pre-extend checks were successful, click **Next**.

Field	Tips
Backup Priority	Enter a number between 1 and 999 to specify the template server's priority in the cascading failover sequence for this resource. A lower number means a higher priority. LifeKeeper assigns the number "1" to the server on which the hierarchy was created. No two servers can have the same priority for a given resource.

- Click **Extend**.

Important	After migrating the database using the automated tool, you should verify that you can access your SQL application and database files. All files are relocated using the system copy utility. After you have validated the success of this procedure, you can remove these data and log files.
------------------	---

Unextending a SQL Hierarchy

To remove a resource hierarchy from a single server in the LifeKeeper cluster, do the following:

Deleting a SQL Hierarchy

1. On the **Edit** menu, select **Resource**, then **Unextend Resource Hierarchy**.
2. Select the **Target Server** where you want to unextend the SQL resource. It cannot be the server where the SQL resource is currently in service. (This dialog box will not appear if you selected the **Unextend** task by right-clicking on a resource instance in the right pane.) Click **Next**.
3. Select the SQL hierarchy to unextend and click **Next**. (This dialog will not appear if you selected the **Unextend** task by right-clicking on a resource instance in either pane).
4. An information box appears confirming the target server and the SQL resource hierarchy you have chose to unextend. Click **Unextend**.
5. Another information box appears confirming that the SQL resource was unextended successfully. Click **Done** to exit the **Unextend Resource Hierarchy** menu selection.

Deleting a SQL Hierarchy

Before deleting a SQL hierarchy or instance, make sure that the hierarchy is active (green) on its primary server. You may also wish to remove the dependencies before deleting the hierarchy; otherwise, the dependencies will be deleted also.

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

1. On the **Edit** menu, select **Resource**, then **Delete Resource Hierarchy**.
2. Select the **Target Server** where you will be deleting your SQL resource hierarchy and 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**. (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.) Click **Next**.
4. An information box appears confirming your selection of the target server and the hierarchy you have selected to delete. Click **Next**.
5. Another information box appears confirming that the SQL resource was deleted successfully.
6. Click **Done** to exit.

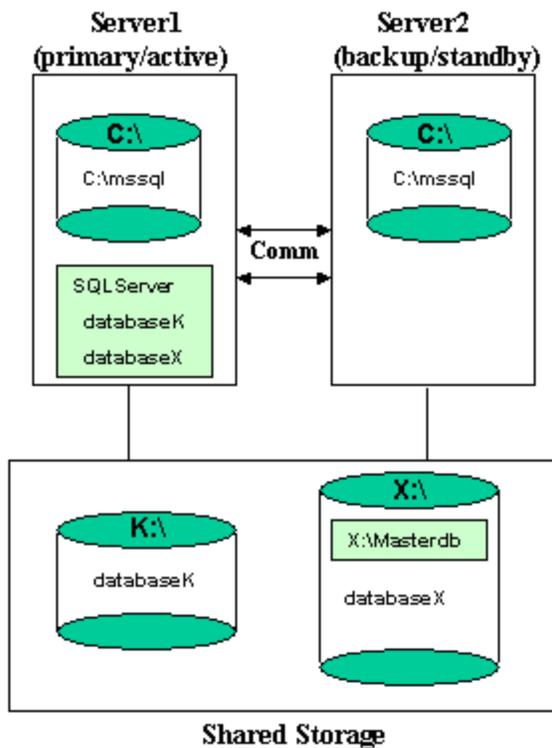
Chapter 3: SQL Server Configuration Considerations

Before you install and configure your clusters, it is important to understand the concepts of Active/Standby configuration, and how multiple instances can be set up in a SQL configuration.

SQL Configuration

A configuration is Active/Standby when there is only one master database for each SQL Server, and it is located on a shared or replicated volume. The services run on only one system at a time. The servers are assigned priorities within LifeKeeper which determine the order of failover for a particular hierarchy.

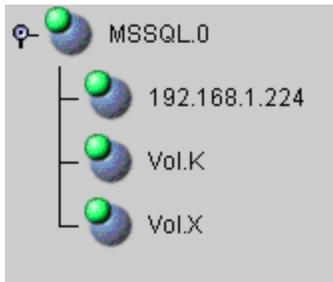
The figure below depicts a single SQL instance installed on a pair of servers. The instance contains two databases, databaseK and databaseX residing on separate volumes. Note that there is a single master database which resides on shared volume X.



Failover

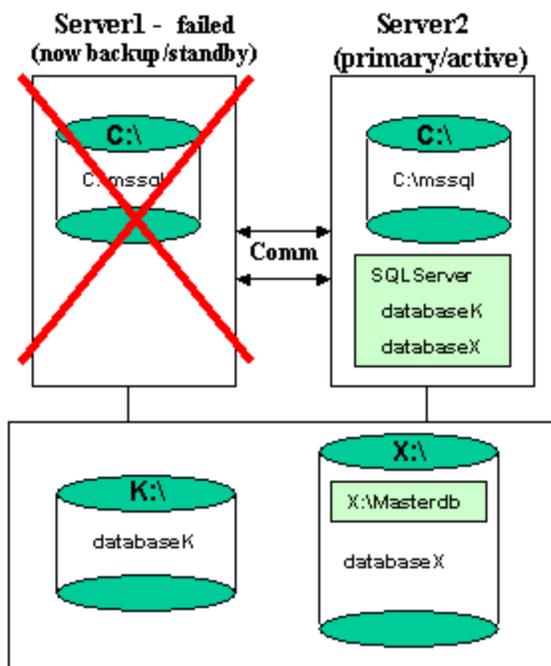
When you create the SQL hierarchy within LifeKeeper, you are asked to specify the SQL instance to be protected, and the IP resource that will be used to connect to the database. LifeKeeper then reads the configuration data for that instance and pulls the associated volumes into the hierarchy.

Once the hierarchy is created, it will appear as follows in the LifeKeeper GUI.



Failover

In the event of failure, LifeKeeper brings the SQL Server hierarchy In Service on the backup Server. SQL Server is started on the backup server and it takes over protection of all defined databases as depicted in the figure below.



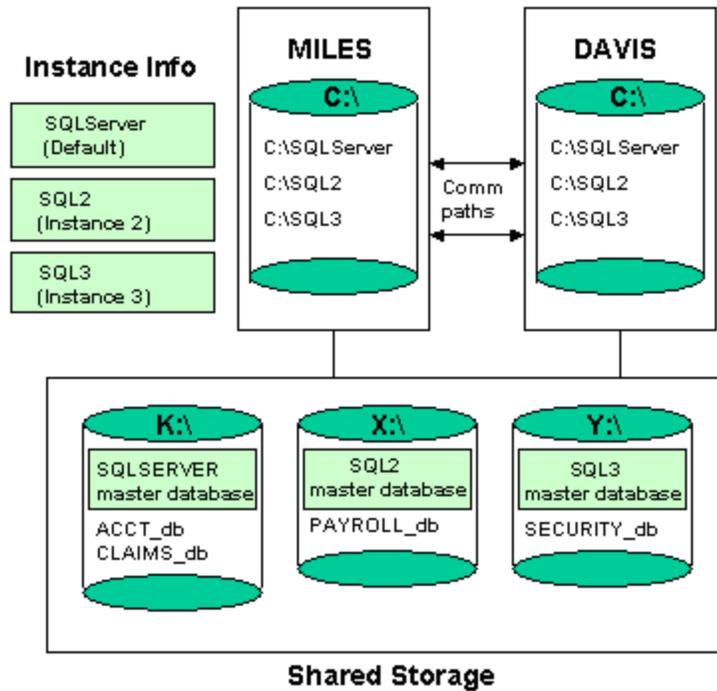
Multiple SQL Instances

SQL Server can be installed multiple times, which creates multiple SQL instances. LifeKeeper can protect multiple instances of SQL Server. LifeKeeper identifies each instance by the unique name

given during SQL installation.

One SQL instance may contain multiple SQL databases. Each instance is protected in a single LifeKeeper hierarchy. Thus, if the SQL instance contains two databases, the corresponding LifeKeeper hierarchy will protect two databases (along with the associated IP and volume resources).

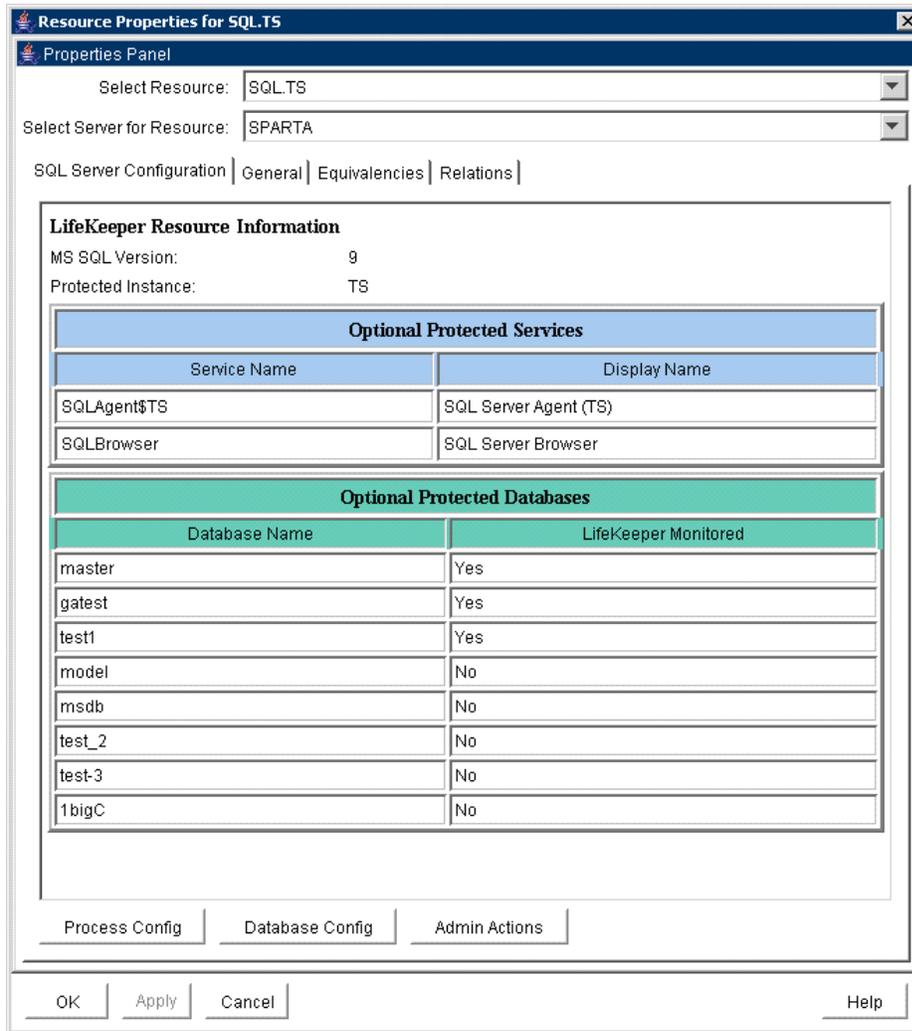
The figure below depicts three SQL instances: SQLServer (the default instance), SQL2, and SQL3. These are installed on a pair of servers, MILES and DAVIS.



Notes	<ul style="list-style-type: none"> • The databases are located on three different shared storage volumes, K, X and Y. Note that the default instance contains two databases and the other two instances contain one database each. • Each server can be the primary and backup server for multiple instances. • It would be possible for MILES to be the primary server for the default instance and DAVIS to be the primary server for the SQL2 and SQL3 instance.
-------	--

Managing a SQL Server Configuration

To administer a protected SQL Server resource from the LifeKeeper GUI, right-click on the SQL Server resource (on the right-hand side of the LifeKeeper GUI) and select **properties**, then select the **SQL Server Configuration** tab. Use the **SQL Server Configuration** page to view or change information about your SQL resource.



Process Config Menu

This menu allows users to modify the list of optional SQL Processes that are protected under the resource hierarchy. LifeKeeper will monitor all protected optional services (see [Monitoring Your SQL Hierarchy](#)).

Select **Action**:

- *Add Process* - Add an additional process to the protected configuration. LifeKeeper will start monitoring new SQL service
- *Delete Process* - Remove a process from the protected configuration.

Field	Tips
Service Name	Enter the service name for the process to Add or Delete from the protected configuration.
Update All Systems	Select Yes to update all systems in this cluster. Otherwise, select No to only update the current system. If you choose No , you must manually add the process to the backup servers.

Database Config Menu

This menu allows users to modify the list of optional SQL Databases that are protected under the resource hierarchy. LifeKeeper will monitor all protected optional SQL Databases by performing a SQL query to test the connection to the database (see [Monitoring Your SQL Hierarchy](#)).

Select **Action**:

- *Add Database* - Add an additional database to the protected configuration. **Note**: The option is available only on the server where the SQL resource is active (In Service).
- *Delete Database* - Remove a database from the protected configuration.

Field	Tips
Enter Database Name	Enter the database name for the process to Add or Delete from the protected configuration.
Update All Systems	Select Yes to update all systems in this cluster. Otherwise, select No to only update the current system. If you choose No , you must manually add the process to the backup servers.

Admin Actions Menu

This menu allows users to manage the SQL administrator user used during LifeKeeper operations or resolve ID conflicts between the primary and backup servers encountered during the extend operation.

Select **Administrative Action**:

- *Manage User* - Display or update the current user name used by the protected resource hierarchy.
- *ID Conflict Resolution (SQL 2005 only)* - Resolves Microsoft SQL ID mismatches encountered during extend of a SQL resource hierarchy. **Note**: This action will make registry modifications to your SQL configuration on the backup server to match the primary server. This will allow the extend operation of the SQL resource to the backup server.

Manage User Menu

Select **Management Action**:

Testing Your SQL Resource Hierarchy

- *Show Current User* - Display the current user name used by the protected resource hierarchy.
- *Change Password* - Update the user password for the current user associated with the protected resource hierarchy.
- *Change User and Password* - Update both the user and password to be used during LifeKeeper operations to administer and monitor the SQL instance. The user must have sql admin privileges for all databases under protection.

Note: The SQL instance must be running to change the user and/or password.

Field	Tips
Enter User Name	Enter the administrative user name. This user account must include SA permissions to all databases under LifeKeeper protection.
Enter Password	Enter the administrative password for the user account being updated.

ID Conflict Resolution (SQL 2005 only)

Field	Tips
Select Service Type	Choose the service to repair the registry to allow LifeKeeper protection: Choose Sql if only the IDs for the SQL Server service differ on the primary and backup servers. Choose Olap if only the IDs for the SQL OLAP Service differ on the primary and backup servers. Choose Rs if only the IDs for the SQL Reporting Service differ on the primary and backup servers. Choose All to update the Sql , Rs , and Olap services simultaneously.
Select Template Server	Choose the template (primary) server for the resource that needs to be repaired. The template server must have the resource in the in-service protected (ISP) state.
Select Target Server	Choose the target (backup) server for the resource to be repaired. The target server must not have the resource in the in-service protected (ISP) state.
Confirmation	Verify information is correct entered from above. Select Continue to perform the action, Back to change information or Cancel to cancel the action.

Testing Your SQL Resource Hierarchy

You can test your SQL resource hierarchy by initiating a manual switchover. This will simulate a

failover of a resource instance from the primary server to the backup server.

Selecting **Edit**, then **Resource**, then **In Service**. For example, an In Service request executed on a backup server causes the application hierarchy to be taken out of service on the primary server and placed in service on the backup 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.

Chapter 4: SQL Server Hierarchy Administration

Follow these guidelines when administering your SQL Server.

Access Via Protected Communication Paths

All remote access of the service should be done through the hierarchy's protected IP addresses. This will ensure that users can access the SQL service regardless of which server it is currently running on.

User Resource Name For Remote Access

Unless the application is cluster-aware, when using **Microsoft SQL Enterprise Manager** to administer the service, you should register it by the switchable resource name (the name by which users access the server using TCP/IP). This gives you a continuous monitor of the viability of this path.

If you register the SQL Server by the system name, you can also monitor the system.

Reserve Volumes For Exclusive SQL Use

The volumes containing the protected SQL files should be reserved for use by SQL exclusively.

A LifeKeeper protected volume may fail to switch over if it is accessed by an application, process or remote user.

Understand Manual Switchover Limitations in SQL Server Environment

Any manual action requires that all users be logged off of the SQL Server resources.

Local processes that have read-only access to volumes do not prevent removal of a resource from service but may cause a restore to fail when you try to switch back. Examples might be the Performance Monitor, which periodically polls each volume, or any running process which is installed on the shared or replicated volume. You can minimize your potential for this type of restore failure by installing the Microsoft SQL Server on local drives and putting only the database on shared or replicated volumes.

Running Microsoft SQL Management Tools

Open the **Microsoft SQL Enterprise Manager** (SQL 2000) or **Microsoft SQL Server**

Configuration Manager (SQL 2005) only when needed and do not run it constantly.

If the **Microsoft SQL Enterprise Manager** (SQL 2000) or **Microsoft SQL Server Configuration Manager** (SQL 2005) is open and active at the database level, it may prevent the SQL hierarchy from coming into service properly and the failover will not complete successfully. If this occurs, close the Microsoft SQL Enterprise Manager (SQL 2000) or Microsoft SQL Server Configuration Manager (SQL 2005) and manually bring the SQL resource into service.

Start and Stop SQL Server Only Through LifeKeeper

Although much of your administration of the Microsoft SQL Server is done through the Microsoft SQL Enterprise Manager, you derive two distinct benefits from bringing the Microsoft SQL Server in service and out of service using the LifeKeeper administration options:

1. **Consistent view.** When LifeKeeper stops and starts Microsoft SQLServer, it maintains a consistent view of the server on all nodes in the configuration.
2. **Configuration details saved.** If you change your Microsoft SQL Server configuration, you can stop and start the server through LifeKeeper or perform a manual switchover and LifeKeeper automatically replicates the configuration changes on the paired node.
Note: If you do not use these options to replicate new configuration information on the paired server, the backup server will use old configuration information in a failover situation.
3. **Protected Microsoft SQL services** should be set to **Manual** startup mode through the **Control Panel "Services"** tool.

Adding Microsoft SQL Server Volumes

As your environment grows, you may need to add new Microsoft SQL Server databases on new shared or replicated volumes. You should perform the following tasks to add the new volumes to the LifeKeeper hierarchy before administering the new databases within Microsoft SQL Server.

To add a new volume resource to an existing SQL Server hierarchy:

1. **Create the resource.** On the server where the SQL Server hierarchy is in service, create and extend the volume resource with the same priority order as that of the SQL hierarchy.
2. **Create dependency.** Right click on the SQL Server resource, and then select Create Dependency from the pop-up menu. For **Child Resource Tag**, select the new Volume resource.

When you have completed these LifeKeeper tasks, you can perform the administration tasks to add the SQL Server database. Adding new databases to volumes that are already part of the resource hierarchy requires no LifeKeeper specific administration.

Recovering From Databases in Suspect State After a Switchover

A database which gets marked as suspect may be caused by starting Microsoft SQL Server when

the volume(s) on which it resides is unavailable to this system. When LifeKeeper is used to protect SQL databases, the starting and stopping of SQL should be performed only as a function of bringing hierarchies in or out of service. In those situations where a database has been marked suspect and it is known that the database is fine, perform the following steps to correct the problem.

For those databases which are suspect on a primary/secondary server:

1. Make sure that the volume(s) where the database resides is actively (green) being protected by this server.
2. Use `sp_resetstatus` to change the state of the database. Execute the following commands from a query window to reset the status of the suspect database:

```
While in master database, execute sp_configure 'allow
updates', 1
```

```
Reconfigure with override
```

```
Sp_resetstatus 'dbname'
```

```
Sp_configure 'allow updates', 0
```

```
Reconfigure with override
```

3. **Stop** Microsoft SQL Server.
4. **Start** Microsoft SQL Server.

Pausing Microsoft SQL Server (MSSQLServer)

It is possible for the SQL Administrator to manually put Microsoft SQL Server into a PAUSED state whereby existing connections to the Microsoft SQL Server can continue processing, but no new connections are allowed. In this case, LifeKeeper detects that the MSSQLServer service is not RUNNING, but will NOT attempt to restart the service locally or fail the SQL hierarchy to the backup server. Neither option is the appropriate action, so monitoring of the SQL resource is essentially forfeited when Microsoft SQL Server is in the PAUSED state.

Because manual intervention was required to put Microsoft SQL Server into this state, the SQL administrator must manually move SQL Server out of this state. Once out of the PAUSED state, LifeKeeper can resume monitoring the SQL resource as outlined above.

Configuring SQL Server to Connect Using the Switchable IP Address

By default, TCP/IP sockets are set as the default network protocol when SQL 2000 and SQL 2005 are installed. If this setting is modified at any time, use the **SQL Server Configuration Manager** tool to re-enable the TCP/IP network setting.

Maintaining SQL Server Login and Passwords

During the creation of a LifeKeeper SQL resource, the user must enter a SQL administrative

username and password for that instance of Microsoft SQLServer. Should the password of this username change at some point in the future, the SQL resource must be updated on all systems in the cluster with this new password. Failure to do so will prevent LifeKeeper from fully monitoring the status of SQL Server on these systems. A warning message will be logged in the Application Event Log, but the SQL resource will not fail as a result of this.

The SQL administrative username and password associated with the LifeKeeper SQL resource can be changed using the **SQL Server Configuration** tab in the LifeKeeper GUI. To administer a protected SQL Server resource from the LifeKeeper GUI, right-click on the SQL Server resource (on the right hand side of the LifeKeeper GUI) and select **properties**, then select the **SQL Server Configuration** tab. Click the **Admin Actions** button and select Manage User on the **SQL Server Configuration** page to view or change information about your SQL resource.

See the section [Managing a SQL Server Configuration](#) for more details on the **SQL Server Configuration** page.

Monitoring Your SQL Hierarchy

For every SQL resource, LifeKeeper monitors the Microsoft SQL Server service and any optional services selected during the creation of the SQL hierarchy or added at a later time. Should any of these services stop, the monitoring process associated with the SQL resource will detect this and attempt to restart the service on the local server if Local Recovery is enabled. If Local Recovery is disabled, the resource will fail over to the backup server.

LifeKeeper will also perform a SQL query to test the connection to all protected SQL databases. If the SQL query fails to the master database, the monitoring process associated with the SQL resource will detect this and attempt to restart the services, if Local Recovery is enabled. If Local Recovery is disabled, the resource will fail over to the backup server. For other protected databases added after creation of the SQL hierarchy, an error will be logged to the **Application Event log**.

Chapter 5: Troubleshooting

This section is intended to provide suggestions and insights into occurrences that are not specifically related to the LifeKeeper software, but have a relationship with the total environment.

Create and Extend Fail if Master or User Databases on System Drive

Symptom:

The **Create** and **Extend** of a Microsoft SQL Server resource will fail if the Master or any user databases are located on the system drive or a volume that can not be protected by LifeKeeper. LifeKeeper does not require that the *tempdb* database be located on a LifeKeeper protected volume as this database is recreated each time SQL Server starts.

Suggested Actions:

The following web sites provide information on moving the Master and user databases to a volume that can be protect by LifeKeeper.

<http://support.microsoft.com/kb/224071/en-us>

<http://www.databasejournal.com/features/mssql/article.php/3379901>

Extend of a SQL Resource Fails (SQL 2005 only)

Symptom:

The Extend of a Microsoft SQL resource will fail if the primary and backup configurations are different (i.e. Microsoft SQL ID mismatch). Below is an example of the error message displayed in the LifeKeeper GUI during the can extend operation:

```
Process: canextend.ksh(1292)
```

```
*ERROR* (No. 14003) The target value for Database file location does not match the template value (target=-
```

```
DQ:\SQL-
```

```
DEFAULT\MSSQL.1\MSSQL\DATA\MASTER-
```

```
.MDF,template=DQ:\SQLDEFAULT\MSSQL.4\MSSQL\DATA\MASTER.MDF)
```

```
Process: canextend.ksh(1572)
```

Suggested Actions:

```
*ERROR* (No. 14003) The target value for Temp DB location does not match the template value (target=-EQ:\SQLDEFAULT\MSSQL.1\MSSQL\LOG\ERRORLOG,template=-EQ:\SQLDEFAULT\MSSQL.4\MSSQL\LOG\ERRORLOG)
```

Process: canextend.ksh(1156)

```
*ERROR* (No. 14003) The target value for Log file location does not match the template value (target=-LQ:\SQLDEFAULT\MSSQL.1\MSSQL\DATA\MASTLOG.LDF,template=-LQ:\SQLDEFAULT\MSSQL.4\MSSQL\DATA\MASTLOG.LDF)
```

Error - extmgr (HAWK, MSSQL.0, MSSQL.0, OSPREY) - canextend failed

Suggested Actions:

Use the ID Conflict Resolution option to resolve Microsoft SQL ID mismatches between the primary and backup servers. In the LifeKeeper GUI, right-click on the **SQL Server resource** (on the right-hand side of the LifeKeeper GUI) and select **properties**, then select the **SQL Server Configuration** tab. Click the **Admin Actions** button and select **ID Conflict Resolution** on the **SQL Server Configuration** page.

See the topic [Managing a SQL Server Configuration](#) for more details on the SQL Server Configuration page.

SQLAgent Service Fails to Start Sometimes for Named Instances

Symptom:

On named instances of Microsoft SQL Server where LifeKeeper is protecting the SQLAgent service, when the resource is originally brought in service, a SQL problem prevents this service from starting and forces a MAXWAIT situation (300 second delay) before the SQL gives up trying to start the SQLAgent service.

This message indicates that the INFO field of the SQL resource has become corrupted. You must delete and re-create the SQL resource. Note that you should remove any IP and volume dependencies prior to deleting the resource. Upon creating the new SQL resource, LifeKeeper will re-create the dependencies.

If the Microsoft SQL Server service is already started on the system where the SQLAgent service is trying to start, you'll likely see this scenario.

If the SQLAgent service and the Microsoft SQL Server service are both started when the SQL hierarchy creation occurs, you will not see this issue.

Suggested Actions:

Stopping and starting the Microsoft SQL Server service usually clears up the problem and the SQL Agent service then starts. However, stopping the SQL Server service is not a good option.

Both the MSSQLServer and the SQLServerAgent service should start up properly using the Local System Account on the default instance or a named instance. They will both start up using a Domain

Administrator account, provided you have added that Domain Admin account to the Local Administrator Group on each system.

Connecting ODBC Clients to Named Instances of SQL Server

Symptom:

After creating an ODBC connection to your LifeKeeper cluster via the protected IP address, the connection fails after switching the SQL resource to the backup server.

Suggested Action:

1. Take each instance out of service and bring it back into service on the PRIMARY. Examine the application event log to determine which IP:PORT that particular SQL Server instance is listening on.
2. Bring each hierarchy In Service on the SECONDARY server and note the IP:PORT each SQL instance is listening on.
3. To insure your clients can connect via ODBC to either server, make sure the PORT each instance is listening on is the same on both the PRIMARY and SECONDARY servers.
4. To do this, use the Microsoft SQL Server Network Utility. Select the SQL instance (it must be running on that machine), highlight the TCP/IP protocol and look at the properties to determine the current default port it is listening on.
5. Change this value so the default PORT is the same on both systems for this instance.
6. Create your ODBC connections for each instance using the protected IPs:PORTs you just set up in Step 5.

