



# **LifeKeeper for Windows**

LifeKeeper Microsoft SQL Server Recovery Kit  
Administration Guide

**May 2005**

---

The product described in this book is a licensed product of SteelEye™ Technology, Inc.

SteelEye Technology and LifeKeeper are registered trademarks and SteelEye is a trademark of SteelEye Technology, Inc.

Microsoft, Windows 2000, Windows 2003, and SQL Server are trademarks or registered trademarks of Microsoft in the U.S. and other countries.

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

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

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

Address correspondence to:

[ip@steeleye.com](mailto:ip@steeleye.com)

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

# Table of Contents

LifeKeeper Microsoft SQL Server Recovery Kit Administration Guide .....	4
Document Contents .....	4
LifeKeeper Documentation .....	4
Recovery Kit Requirements .....	4
Recovery Kit Installation.....	5
Kit Removal .....	5
SQL Server Recovery Kit Overview .....	6
Differences between Microsoft SQL Server 7.0 and 2000 with LifeKeeper.....	6
Microsoft SQL 7.0 .....	6
Microsoft SQL 2000 .....	6
SQL Server Configuration Considerations .....	7
SQL Active/Standby Configuration .....	7
Active/Standby Failover.....	8
Multiple SQL Instances (Microsoft SQL 2000 Only) .....	9
Installing and Configuring SQL Server with LifeKeeper.....	10
Setup Checklist.....	10
Setup Tasks .....	10
Set up shared/replicated storage.....	10
Set up and test your network.....	10
Install and Configure LifeKeeper Core and Recovery Kits.....	11
Create volume and IP resource hierarchies .....	11
Install Microsoft SQL Server software.....	11
Additional Setup Tasks for Extended Configurations .....	12
Create your SQL Server Hierarchy .....	12
Resource Configuration Tasks.....	13
Creating the SQL Hierarchy .....	13
Extending a SQL Hierarchy .....	16
Unextending an SQL Hierarchy .....	17
Deleting a SQL Hierarchy .....	17
Testing Your Resource Hierarchy .....	18
Hierarchy Administration .....	19
Access via protected communication paths.....	19
Use resource name for remote access.....	19
Reserve volumes for exclusive SQL use.....	19
Start and stop SQL Server only through LifeKeeper.....	19
Understand manual switchover limitations.....	19
Running Microsoft SQL Enterprise Manager .....	20
Monitoring Your SQL Hierarchy .....	20
Adding Microsoft SQL Server Volumes .....	20
For Active/Standby Configurations: .....	20
Recovering from Databases in “Suspect” State after a Switchover.....	20
Pausing MSSQLServer.....	21
Configuring SQL Server to Connect Using the Switchable IP Address.....	22
Reattaching a SQL Database .....	22
For Microsoft SQL 7.0:.....	22
For Microsoft SQL 2000:.....	22
Maintaining SQL Server Login/Passwords .....	23
Synchronizing the SQL Username/Password.....	23
Error Messages.....	23
Troubleshooting.....	25
SQLAgent service fails to start sometimes for named instances.....	25
Connecting ODBC clients to Named Instances of SQL Server.....	25

# LifeKeeper Microsoft SQL Server Recovery Kit Administration Guide

The LifeKeeper® Microsoft SQL Server Recovery Kit provides fault resilience for Microsoft SQL Server 7.0 or Microsoft SQL 2000 (8.0) in a LifeKeeper environment.

## Document Contents

This guide provides the following information topics:

- [SQL Server Recovery Kit Overview](#). This section provides a general overview of the product and the LifeKeeper environment.
- [SQL Server Configuration Considerations](#). This section describes basic configuration concepts that are important to a successful installation of Microsoft SQL Server with LifeKeeper.
- [Installing and Configuring SQL Server with LifeKeeper](#). Follow the pre-installation checklist and the special procedures to set up your servers for SQL Server.
- [SQL Resource Configuration Tasks](#). After you have completed the necessary setup tasks, use the steps in this section to create, extend and manage the SQL hierarchy.
- [Hierarchy Administration](#). This section describes how set up and administer a single database instance, add system dependencies, and understand manual switchover.

## LifeKeeper Documentation

The following documentation is associated with the LifeKeeper core:

- *Release Notes*
- *Online Product Manual*
- *Planning and Installation Guide*

This documentation, along with documentation associated with other LifeKeeper recovery kits and LifeKeeper Data Replication for Windows, is available online at:

[www.steeleye.com/support/documentation](http://www.steeleye.com/support/documentation)

## 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 2000 Server Standard, Advanced, Data Center Editions
- Windows Server 2003 Standard, Enterprise, Data Center, Web Editions
- **LifeKeeper software.** You must install the same version of LifeKeeper for Windows on *all* servers in the cluster.

- **LifeKeeper Data Replication for Windows (optional).** If you plan to use Microsoft SQL Server with replicated volumes rather than shared storage, you should install the LifeKeeper Data Replication software on each server.
- **Microsoft SQL Server RDBMS software.** The kit is compatible with Microsoft SQL Server 7.0 or Microsoft SQL 2000 (8.0). However, the same version of Microsoft SQL Server must be installed on all systems in the cluster.  
**Note:** If using Microsoft SQL Server 7.0, the Microsoft SQL Server 7.0 Service Pack 3 is required.
- **Communications 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.

## Recovery Kit Installation

The LifeKeeper Microsoft SQL Server Recovery Kit is distributed on CD-ROM or via ftp download. Installation is simple and quick, using InstallShield to provide a standard installation interface. For complete LifeKeeper installation instructions, refer to the *Planning and Installation Guide*.

Before installing the LifeKeeper Microsoft SQL Server Recovery Kit software, be sure you are familiar with the product prerequisites listed above, as well as the installation procedure outlined in the [Setup Checklist](#).

### Kit Removal

To remove the LifeKeeper Microsoft SQL Server Recovery Kit software, choose the "LifeKeeper Microsoft SQL Recovery Kit for LifeKeeper v5.0" in the Add/Remove programs 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.

## SQL Server Recovery Kit Overview

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 complete 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. The LifeKeeper Microsoft SQL Server Recovery Kit allows instances of all of the following SQL services to be protected:

- MSSQLSERVER (required)
- Distributed Transaction Coordinator (optional)
- SQLSERVERAGENT (optional)
- Microsoft Search (optional)
- MSSQLServerADHelper (optional)

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. The recovery can be completely transparent to database users. Once LifeKeeper switches all dependent resources to the backup server, it starts the Microsoft SQL service if it is not already running.

## Differences between Microsoft SQL Server 7.0 and 2000 with LifeKeeper

There are several differences in how the LifeKeeper Microsoft SQL Server Recovery Kit operates with Microsoft SQL Server 7.0 and Microsoft SQL Server 2000 (8.0). These differences are summarized below.

### Microsoft SQL 7.0

- The Microsoft SQL Server 7.0 Service Pack 3 is required.
- Reattaching a database for Microsoft SQL Server 7.0 must be done using the `sp_attach_db` stored procedure.

### Microsoft SQL 2000

- Microsoft SQL 2000 supports multiple instances of SQL Server.
- The switchable IP addresses should be created prior to SQL hierarchy creation. This is because SQL binds to known IP addresses when the SQL Server service starts. SQL is unable to make a client connection to an IP address that is defined after SQL Server service has started.
- LifeKeeper switchable IP addresses must have a corresponding DNS entry. If absent, then on startup the SQL Server may not bind to the switchable IP address.
- Reattaching a database for Microsoft SQL Server 2000 is done via menu options in the Microsoft SQL Enterprise Manager.

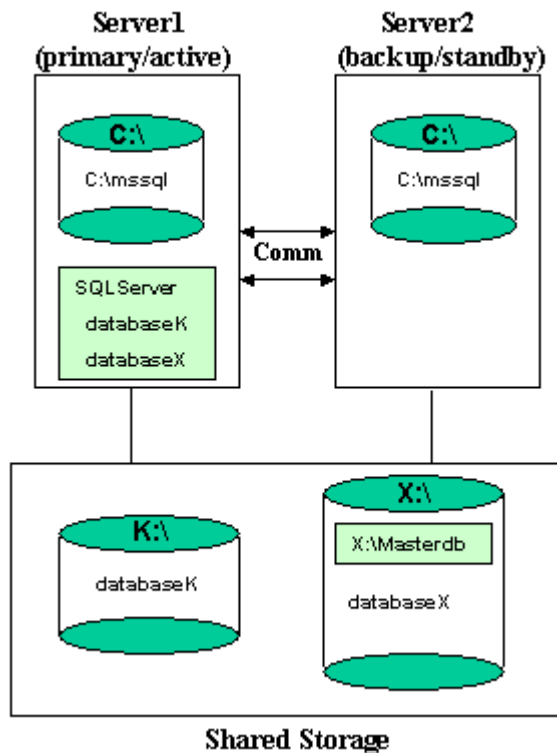
## SQL Server Configuration Considerations

Before you install and configure your clusters, it is important to understand the concepts of Active/Standby configurations, and how multiple instances can be setup in a SQL 2000 configuration.

### SQL Active/Standby Configuration

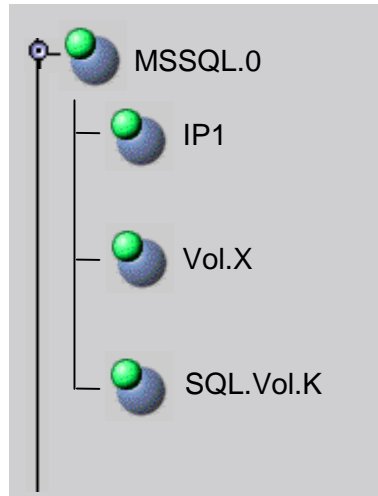
A configuration is **Active/Standby** when there is only one master database for each SQL Server, and it is located on a shared or replicated volume. The services run on only one system at a time. The servers are assigned priorities within LifeKeeper which determine the order of failover for a particular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



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.

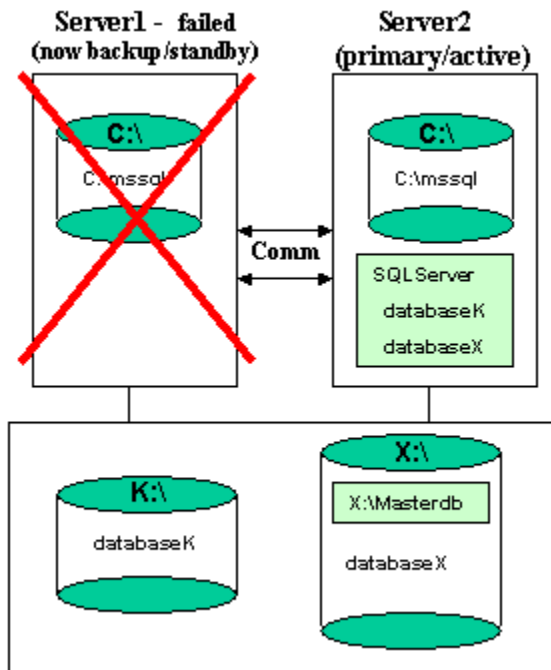


**Notes:**

- The hierarchy tag name *MSSQL.0* is the default offered by LifeKeeper.
- The volume tagname *Vol.X* indicates that the volume resource already existed before the SQL hierarchy was created.
- The volume tagname *SQL.Vol.K* indicates that this volume resource was created automatically during the SQL hierarchy creation.

**Active/Standby 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.

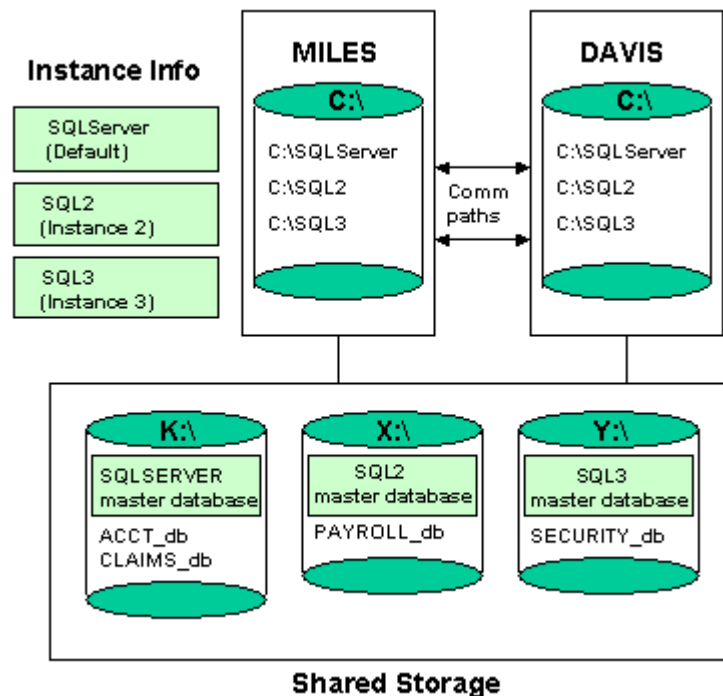


## Multiple SQL Instances (Microsoft SQL 2000 Only)

In Microsoft SQL 2000, 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:

- 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.
- The figure above depicts Active/Standby configurations because the master databases are located on the shared storage.
- 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.

# 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.

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.

## Setup Checklist

The installation and setup sequence should be performed in the following sequence. More detailed instructions for each of these steps are provided in the sections to follow. See the *Planning and Installation Guide* for additional details on the first three steps.

1. Set up your shared or replicated storage to reserve protected volumes for exclusive Microsoft SQL Server use.
2. Set up and test your network.
3. Install and configure the LifeKeeper Core, IP Recovery Kit, and Microsoft SQL Server Recovery Kit on each server.
4. Create volume and IP resource hierarchies.
5. Install Microsoft SQL Server software on each server.
6. Create the SQL Server resource hierarchy.

## Setup Tasks

During setup, first configure the primary server, then a backup server, and finally, any further servers you wish to include in the cluster.

### Set up shared/replicated storage.

Follow these steps to ensure the correct configuration.

1. Use the Windows Disk Management tool to configure your shared or replicated disk resources and **define the volumes that you want to use**. It is required that databases with different primary servers be placed on different volumes.
2. Use Windows Explorer to **unshare from the network all volumes to be used by the SQL Services**. A LifeKeeper-protected volume may fail to switchover if another application or process is using it.

### Set up and test your network.

Ensure your network is configured and working correctly before installing LifeKeeper, since the communications paths and LifeKeeper GUI depend upon the network.

## Install and Configure LifeKeeper Core and Recovery Kits

Install the LifeKeeper Core, IP Recovery Kit and LifeKeeper Microsoft SQL Server Recovery Kit on each server in the cluster. You must have the same version of LifeKeeper on all servers.

### Create volume and IP resource hierarchies

1. For each shared or replicated volume you plan to use, create a volume hierarchy to protect the volume, and ensure the volume hierarchy is In Service on the primary server. (It is possible - but not recommended - to skip this step, in which case LifeKeeper will create the volume(s) for you when you create the SQL Server hierarchy.)
2. Create your LifeKeeper IP resource hierarchies on the primary server, and extend them to the backup server. See the *Online Product Manual* for details on creating IP resource hierarchies. Note that LifeKeeper-protected IP addresses must have a corresponding DNS entry. If absent, then on startup the SQL Server may not bind to the protected IP address.

The IP resources must be defined, and should not be part of another hierarchy when the time comes for you to create your SQL hierarchy.

**Note:** If you create the IP resource *after* creating the SQL hierarchy, then you must stop and re-start the SQL service in order for SQL to recognize the IP resource.

### Install Microsoft SQL Server software.

How you install the software depends on what version of Microsoft SQL you are using, and on whether you are installing an Active/Standby or Active/Active configuration. Both methods are described below.

#### Active/Standby Installation (Microsoft SQL 2000 or Microsoft SQL 7.0)

1. **On the primary server**, install Microsoft SQL Server using the following guidelines:
  - In the *Computer Name* dialog, choose Local Computer.
  - In the *Setup Type* dialog, under Destination Folder be sure to specify a folder on the *local* disk for Program Files and a *shared* volume (protected by LifeKeeper) for Data Files.
  - In the *Services Accounts* dialog, select “Use the Local System Account”.
  - 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.
2. When installation is complete, use Microsoft SQL Enterprise Manager to verify that SQL Server can start properly on the primary server.
3. In LifeKeeper, bring the shared/replicated volume resource hierarchy In Service on the **backup** server.
4. **On the backup server**, open Explorer and access the drive associated with the shared/replicated volume.
5. Delete the directory where you previously installed the SQL data files. (You will re-install them in the next step).
6. Install Microsoft SQL Server on the backup server **EXACTLY** as you did on the primary server (program files on the local disk and data files on the shared volume).

7. When installation is complete, use Microsoft SQL Enterprise Manager to verify that SQL Server can start properly on the backup server.
8. In LifeKeeper, bring the shared volume resource hierarchy In Service on the **primary** server.

## Additional Setup Tasks for Extended Configurations

If your configuration uses a shared storage device, you may also choose an **Active/Standby** configuration that will be extended to a third (or more) server(s).

1. Configure two systems in an **Active/Standby** configuration per the steps given above.
2. Switch your protected volumes to the third server.
3. Install the Microsoft SQL Server software on the local drive, and the master database on the same shared volume as used by the other servers. This will permit you to extend the **Active/Standby** hierarchy and utilize LifeKeeper's cascading feature.

## Create your SQL Server Hierarchy

After you complete the setup and installation tasks, you are ready to use LifeKeeper to **create** and **extend** a SQL Server hierarchy. See the next section for details on configuring SQL Server resource hierarchies.

## Resource Configuration Tasks

Once you have completed the setup tasks as described in the previous section, you are ready to create and extend your SQL Server resource hierarchies.

The following four tasks are described in this guide, as they are unique to a SQL resource instance, and different for each recovery kit.

- **[Create a Resource Hierarchy](#)**. Creates an application resource hierarchy in your LifeKeeper cluster.
- **[Extend a Resource Hierarchy](#)**. Extends a resource hierarchy from the primary server to a backup server.
- **[Unextend a Resource Hierarchy](#)**. Unextends (removes) a resource hierarchy from a single server in the LifeKeeper cluster.
- **[Delete a Resource Hierarchy](#)**. Deletes a resource hierarchy from all servers in your LifeKeeper cluster.

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

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

**Note:** Throughout the rest of this section, configuration tasks are performed using the **Edit** menu. You can also perform most of these tasks:

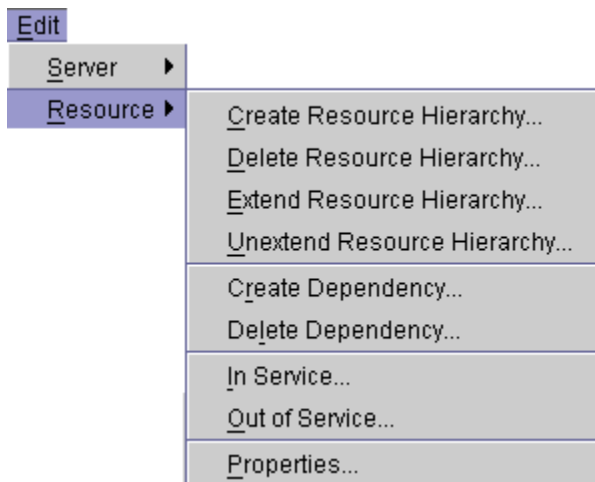
- from the toolbar
- by right clicking on a global resource in the left pane of the status display
- by right clicking on a resource instance in the right pane of the status display

Using the right-click method allows you to avoid entering information that is required when using the **Edit** menu.

### Creating the SQL Hierarchy

After you have completed the necessary setup tasks, use the following steps to define the SQL Server hierarchy to protect your database(s).

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



The *Create Resource Wizard* dialog box will appear with a drop down list box displaying all recognized recovery kits installed within the cluster.

2. Select *MS SQL Server* and click **NEXT**.
3. You will be prompted to enter the following information. When the **Back** button is active in any of the dialog boxes, you can go back to the previous dialog box. This is 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
Switchback Type	Choose either intelligent or automatic This dictates how the SQL instance will be switched back to this server when the server comes back up after a failover. The switchback type can be changed later from the General tab of the Resource Properties dialog box. <b>Note:</b> The switchback strategy must match that of the IP and volume resource to be used by the SQL resource, or else the create will fail.
Server	Select the Server on which you want to create the hierarchy.

Select MSSQL Server Instance ( <i>SQL 2000 only</i> )	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.
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).  <b>Note:</b> The IP address you select should not already be extended to another server. If you select an IP address that is already extended, the create process will fail in a later step.
Named Pipe Alias	Optional, but not recommended. If you intend to use Named Pipes (LAN Manager), select an alias name for the named pipe connection. Leave this blank if you choose not to use Named Pipes.  <b>Note:</b> The LAN Manager resource you select should not already be extended to another server. If you select a LAN Manager resource that is already extended, the create process will fail in a later step.
Microsoft SQL Server Tag	Enter a unique tag name, or you can accept the default tag name offered by LifeKeeper.
Quick Check Interval	Enter the interval (in minutes) between basic checks of the resource's availability. Different values can be specified for each system. The default value is 3 minutes. Value can be between 0 to 10080. Setting interval value to 0 will disable the quick check.
Deep Check Interval	Enter the interval (in minutes) between extensive checks of the resource's availability. This program utilizes Quickcheck for its Deepcheck implementation. Different values can be specified for each system. The default value is 5 minutes. Value can be between 0 to 10080. Setting interval value to 0 will disable the Deep Check.
Local Recovery	Select <b>Yes</b> to enable Local Recovery for this

	resource. Local recovery for a SQL resource means that if there is any of the protected services fail, LifeKeeper will attempt to restart the affected service. If the restart is unsuccessful, then LifeKeeper will failover the service to the backup server.
--	---

- After you click **Create**, the *Create Resource 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*. Refer to Step 2 under Extending a SQL Resource Hierarchy (below) for details on how to extend your resource hierarchy to another server.

## 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.

- On the **Edit** menu, select **Resource**, then Extend Resource Hierarchy. The Pre-Extend Wizard appears. If you are unfamiliar with the Extend operation, click **Next**. If you are familiar with the LifeKeeper **Extend Resource Hierarchy** defaults and want to bypass the prompts for input/confirmation, click **Accept Defaults**.
- 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.

Field	Tips
Template Server	Enter the server where your SQL resource is currently in service.
Tag to Extend	Select the SQL resource you wish to extend.
Target Server	Enter or select the server you are extending <i>to</i> .
Switchback Type	This dictates how the SQL instance will be switched back to this server when it comes back into service after a failover to the backup server. You can choose either intelligent or automatic. The switchback type can be changed later, if desired, from the General tab of the Resource Properties dialog box. <b>Note:</b> Remember that the switchback strategy must match that of the dependent resources to be used by the SQL resource.
Template Priority	(This field appears only if you did NOT extend directly from the Create function.) 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.
Target Priority	Enter a number between 1 and 999 to specify the target server's priority in the cascading failover sequence for this resource. A lower number means a higher priority. LifeKeeper offers a default of 10 for the first server to which a hierarchy is extended.

3. After receiving the message that the pre-extend checks were successful, click **Next**.
4. Depending upon the hierarchy being extended, LifeKeeper will display a series of information box showing the Resource Tags to be extended, which cannot be edited. Click **Extend**.
5. Select "Yes" to enable Local Recovery for the SQL resource on the target server; otherwise choose "No".
6. After receiving the message "Hierarchy extend operations completed" click **Next Server** to extend the hierarchy to another server, or click **Finish** if there are no other extend operations to perform.
7. After receiving the message "Hierarchy Verification Finished", click **Done**.

## Unextending an SQL Hierarchy

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

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 chosen 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.

## Testing Your 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.

# 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.

## Use 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 address). 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 switchover if another it is accessed by an application, process, or remote user.

## 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:

- **Consistent view.** When LifeKeeper stops and starts Microsoft SQL Server, it maintains a consistent view of the server on all nodes in the configuration.
- **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.
- **Protected** Microsoft SQL services should be set to **Manual** startup mode through the Control Panel "Services" tool.

## Understand manual switchover limitations.

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 Enterprise Manager

Open the Microsoft SQL Enterprise Manager only when needed, and do not run it constantly.

If the Microsoft SQL Enterprise Manager 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 and manually bring the SQL resource into service.

## Monitoring Your SQL Hierarchy

For every SQL resource, LifeKeeper monitors the MSSQLServer service and any optional services selected during the creation of the SQL hierarchy. 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.

## 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:

### For Active/Standby Configurations:

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 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

To connect to Microsoft SQL Server via an IP address, such as the LifeKeeper switchable IP address, you must make the following changes in SQL and in the client configuration:

1. Configure Microsoft SQL Server to use TCP/IP sockets network protocol in addition to named pipes (which is the default).
  - a. Make this change in SQL Setup under Change Network Support.
  - b. After selecting TCP/IP sockets, stop and restart the SQL Server service.
2. In the client configuration utility for SQL, do one of the following:
  - a. Make TCP/IP sockets the default network protocol (recommended), OR
  - b. Add a server in the Advanced window which specifies the IP address for the server name and uses TCP/IP sockets for the network library.

### Notes:

- Without these changes, the user receives an error indicating Microsoft SQL Server is not available or does not exist when trying to connect using the switchable IP address.
- For Microsoft SQL 2000, switchable IP addresses should be created prior to the SQL hierarchy creation. This is because SQL binds to known IP addresses when the SQL Server service starts. SQL is unable to make a client connection to an IP address that is defined after SQL Server service has started.

## Reattaching a SQL Database

A SQL database may need to be reattached when:

- You have deleted an out of service hierarchy and the database has become detached.
- You have tried to recreate a hierarchy and the database is detached.

### For Microsoft SQL 7.0:

Use the stored procedure in Microsoft SQL 7.0 to reattach your database:

Go to Query Analyzer and type in the following

```
sp_attach_db 'dbname', 'filename1', 'filename2'...
```

for as many filenames as the database is composed of.

So for example, if you were going to attach DB1, and its two files were G:\MSSQL\DB1.mdf and G:\MSSQL\DB1.ldf, the command to reattach the database would be

```
sp_attach_db 'DB1', 'G:\MSSQL\DB1.mdf', 'G:\MSSQL\DB1.ldf'
```

### For Microsoft SQL 2000:

Use the following menu option in the Microsoft SQL Enterprise Manager:

1. Right-click on the registered server name.
2. On the pop-up menu, click **All Tasks**, then **Attach Database**.

3. Select the database to be reattached.

## Maintaining SQL Server Login/Passwords

During the creation of a LifeKeeper SQL resource, the user must enter a SQL administrative username and password for that instance of Microsoft SQL Server. 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 leave the SQL resource and the information in the SQL security database out of sync and will prevent the SQL resource from coming in and out of service properly.

This issue is most critical in an Active/Active (SQL 7.0) configuration. This is because Microsoft SQL Server runs locally on each system, and it simply attaches or detaches the databases when LifeKeeper brings the resource in service or out of service. Prior to detaching or attaching a database, LifeKeeper checks the SQL security database to ensure the username and password associated with the SQL resource is still valid. If the username/password combination is no longer valid, you will receive an error message stating that the database cannot be attached. LifeKeeper provides a utility called **updateLKSQL** to resynchronize the SQL resource.

### Synchronizing the SQL Username/Password

Anytime the username or password is changed within SQL (typically through SQL Enterprise Manager), you will need to run the **updateLKSQL** utility on the system where the SQL resource is in service. The utility will update all systems to which the SQL resource is extended.

1. In Microsoft SQL Enterprise Manager, change the password associated with the user name.
2. On the system where the SQL resource is ISP, open a command prompt and execute the **updateLKSQL** utility as follows:

Change to the directory where the utility resides.

```
cd c:\<LKROOT>\admin\kit\sqlapp\bin
```

Run the **updateLKSQL** shell script:

```
sh updateLKSQL MSSQL.0 sqladmin sqladminpw
```

where:

- MSSQL.0 = the tagname of SQL resource to update on all systems that this resource has been extended to.
- sqladmin = "sa" or "sa" equivalent username defined for SQL Server on all systems. If the username is different than the username currently defined in the SQL resource, the new username will replace the existing username.
- sqladminpw = new password for the username specified above. This should match what you specified in STEP 1.

### Error Messages

There are three possible error messages associated with the **updateLKSQL** utility:

Error Message	Explanation
---------------	-------------

Cannot connect to <SERVERNAME>	Username and password combination specified by the utility does not exist or does not match the Microsoft SQL Server security database on <SERVERNAME>.
<SERVERNAME> is not available	LifeKeeper is not running on <SERVERNAME> so the utility cannot update the SQL resource on <SERVERNAME>.
<b>updateLKSQL</b> utility did not complete successfully.	<p>The utility could not update the SQL resource on all systems it is defined to for the reasons mentioned above. Resolve the above issue(s) and run the utility again and make sure the resource is updated successfully on all systems.</p> <p><b>Note:</b> Failure to do this will result in the SQL Resource being unable to come in service on the server where the SQL Resource has not been updated.</p>

# Troubleshooting

## 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 TCPIP 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.