



SIOS Protection Suite for Linux v8.3.2

DB2 Recovery Kit
Administration Guide

Nov 2014

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 © 2014
By SIOS Technology Corp.
San Mateo, CA U.S.A.
All rights reserved

Table of Contents

Introduction	3
Document Contents	3
LifeKeeper Documentation	4
DB2 Recovery Kit Requirements	5
Hardware Requirements	5
Software Requirements	5
Overview	6
LifeKeeper for Linux DB2 Recovery Kit	6
Configuring the LifeKeeper for Linux DB2 Recovery Kit	7
Using DB2 with Raw I/O	7
Running DB2	7
Configuration Considerations for DB2 Single Partition	8
Configuration Considerations for DB2 Multiple Partition	8
Issues Regarding DB2 EEE or multiple partition ESE and NFS	8
Configuration Requirements	9
Configuration Considerations for All DB2 configurations	11
Configuration Examples	12
LifeKeeper Configuration Tasks	17
Overview	17
Creating a DB2 Resource Hierarchy	18
Deleting a Resource Hierarchy	19
Extending Your Hierarchy	21
Unextending Your Hierarchy	23
Testing Your Resource Hierarchy	24
Performing a Manual Switchover from the LifeKeeper GUI	24
Troubleshooting	25
Error Messages	26
Appendix: Setting up DB2 to use Raw I/O	27
Requirements	27
Naming Conventions	27
Raw I/O Setup Steps	27

DB2 Recovery Kit Administration Guide

Introduction

The SPS for Linux DB2 Recovery Kit provides fault resilient protection for DB2 database instances. LifeKeeper, together with the DB2 Universal Database product family afford increased availability to DB2 operating environments by effectively recovering database server failures without significant down-time or human intervention.

Document Contents

This guide contains the following topics:

- [LifeKeeper Documentation](#). A list of LifeKeeper for Linux documentation and where to find them.
- [Requirements](#). A description of the hardware and software necessary to properly setup, install, and operate the DB2 Recovery Kit. Refer to the *LifeKeeper for Linux Planning and Installation Guide* for specific instructions on how to install or remove LifeKeeper for Linux software.
- [Overview](#). A description of the DB2 Recovery Kit's features and functionality.
- [Configuring the LifeKeeper for Linux DB2 Recovery Kit](#). A description of the procedures required to properly configure the DB2 Recovery Kit.
- [LifeKeeper Configuration Tasks](#). A description of the tasks for creating and managing your DB2 resource hierarchies using the LifeKeeper GUI.
- [Troubleshooting](#). A list of LifeKeeper for Linux error messages including a description for each.
- [Appendix](#). Steps for setting up DB2 to use raw I/O.

LifeKeeper Documentation

The following LifeKeeper product documentation is available from SIOS Technology Corp.:

- *LifeKeeper for Linux Release Notes*
- *LifeKeeper for Linux Online Product Manual* (available from the Help menu within the LifeKeeper GUI)
- *LifeKeeper for Linux Planning and Installation Guide*

This documentation, along with documentation associated with optional LifeKeeper Recovery Kits, is available on the SIOS Technology Corp. website at:

<http://us.sios.com/support>

DB2 Recovery Kit Requirements

Your LifeKeeper configuration must meet the following requirements **prior** to the installation of the LifeKeeper for Linux DB2 Recovery Kit. Please see the *LifeKeeper for Linux Planning and Installation Guide* for specific instructions regarding the configuration of your LifeKeeper hardware and software.

Hardware Requirements

- **Servers** - LifeKeeper for Linux supported servers configured in accordance with the requirements described in the *LifeKeeper for Linux Planning and Installation Guide* and the *LifeKeeper Release Notes*.
- **IP Network Interface Cards** - Each server requires at least one Ethernet TCP/IP-supported network interface card. Remember, however, that a LifeKeeper cluster requires two communications paths; two separate LAN-based communication paths using dual independent sub-nets are recommended for heartbeats, and at least one of these should be configured as a private network. Using a combination of TCP and TTY heartbeats is also supported.

Software Requirements

- **TCP/IP Software** - Each server in your LifeKeeper configuration requires TCP/IP software.
- **IBM Software** - Please refer to the *LifeKeeper Release Notes* for specific DB2 version requirements on certain Linux distributions and hardware architectures.
- **LifeKeeper Software** - It is imperative that you install the same version of the LifeKeeper software and apply the same versions of the LifeKeeper software patches to each server in your cluster.
- **LifeKeeper for Linux DB2 Recovery Kit** - The DB2 Recovery Kit is provided on a CD. It is packaged, installed and removed via the Red Hat Package Manager, rpm. The following rpm file is supplied on the LifeKeeper for Linux DB2 Recovery Kit CD:

steeleye-lkDB2

Please see the *LifeKeeper for Linux Planning and Installation Guide* for specific instructions on the installation and removal of the LifeKeeper for Linux software.

- **LifeKeeper for Linux NFS Recovery Kit - required for use of DB2 EEE and multiple partition ESE deployments.** This recovery kit is provided on CD in the **steeleye-lkNFS** package. **Important:** See the section [Issues Regarding DB2 EEE or multiple partition ESE and NFS](#) for important configuration information.

Overview

LifeKeeper for Linux DB2 Recovery Kit

In versions 8 and greater, DB2 UDB Enterprise Edition (EE) and Enterprise-Extended Edition (EEE) have been combined into a single product named DB2 UDB Enterprise Server Edition (ESE). Previous versions included two separate enterprise level database servers, the Enterprise Edition (EE) as a standard relational database management system and the Enterprise-Extended Edition (EEE) as an extension of the EE database server to support multi-partition databases.

The LifeKeeper for Linux DB2 Recovery Kit provides protection for the database manager in the EE, WE, and WSE environments, and for the database partition servers in an EEE environment. In a combined ESE environment, the recovery kit provides protection for both the database manager and the database partition servers.

Users may elect to define the DB2 Administration Server for each machine within the LifeKeeper cluster. When the DB2 Administration server is defined, LifeKeeper will attempt to start the DB2 Administration Server as a function of the DB2 hierarchy create and the DB2 hierarchy restore operations.

Configuring the LifeKeeper for Linux DB2 Recovery Kit

This section describes the LifeKeeper for Linux DB2 Recovery Kit configuration details. It also contains information you should consider before you start to configure and administer the DB2 Recovery Kit. Please refer to your *LifeKeeper Online Product Manual* for instructions on configuring LifeKeeper Core resource hierarchies.

Using DB2 with Raw I/O

If you plan to use DB2 with Raw I/O devices, you must install the LifeKeeper Raw I/O Recovery Kit from the LifeKeeper Core CD. You must also properly set up the Raw I/O devices prior to use. See the [Appendix](#) for instructions.

Running DB2

In some instances the startup times of the DB2 processes can be excessive when using DB2 8.x under LifeKeeper protection. Making the following change to the kernel network parameters can improve this situation. As above, add the following line to the */etc/sysctl.conf* file on each LifeKeeper clustered system that will be running DB2 8.x:

```
net.ipv4.tcp_syn_retries=1
```

Then running **sysctl -p** will cause this change to take effect.

Configuration Considerations for DB2 Single Partition

The following should be considered before operating the LifeKeeper for Linux DB2 Recovery Kit in the single partition or workgroup environment:

1. LifeKeeper requires the location of the DB2 instance home directory as well as associated databases, tablespaces, and resources be stored on shared drives. The shared drives are automatically protected at the time the hierarchy is created. During creation of the DB2 resource hierarchy, the DB2 database manager is created as the parent resource while the shared file systems containing instance home directories and actual databases are created as dependent resources. Consequently, if **after** the creation of your DB2 hierarchy you decide to create a database on a shared file system that is not protected by LifeKeeper, you will need to create a resource hierarchy for that file system and make it a dependency of your DB2 resource hierarchy.
2. When the database manager becomes inoperable on the primary system, the service fails over to a previously defined backup system. The database service on the backup system becomes available immediately after the dependent resources fail over and the database manager is brought into service. Previously connected DB2 clients are disconnected and must reconnect to the functioning server. Any uncommitted SQL statements are rolled back and should be re-entered.

Configuration Considerations for DB2 Multiple Partition

DB2 Multiple Partition RESTRICTIONS: All DB2 multiple database partition servers will be protected on a particular machine when the LifeKeeper DB2 resource hierarchy is created on that machine. The nodes to protect are determined by examining the following file:

<instance home>/sqllib/db2nodes.cfg

Future plans for this recovery kit include added functionality to allow for N-way failover.

Issues Regarding DB2 EEE or multiple partition ESE and NFS

If the NFS export point for the DB2 instance home directory becomes unavailable while the DB2 instances are running, the system will hang while waiting for the export point to become available again. Many system operations will not work correctly, including a system reboot. You should be aware that the NFS server for the DB2 multiple partitions cluster should be protected by LifeKeeper and should not be manually taken out of service *unless* all the partitions in the DB2 cluster are also taken out of service *before* shutting down the NFS resource. Additionally, the DB2 partitions cannot be brought into service unless the NFS resource is in service.

To avoid accidentally causing your cluster to hang by inadvertently stopping the NFS server, we make the following recommendations:

NFS Recommendations

Use additional servers: It is highly recommended that you have a separate cluster for the NFS export point from which the DB2 instance home is mounted. The NFS export point on this cluster should be protected with the LifeKeeper NFS Server Recovery Kit.

If you do not have at least *two* additional servers available, you can reduce the chances of experiencing the problem described above by adding *one* additional server to the DB2 cluster. This additional server would export the NFS hierarchy. One of the other nodes in the cluster would serve as a backup. In this configuration the symptoms could occur if the NFS hierarchy were to failover to the backup node. The NFS export point on this cluster should be protected with the LifeKeeper NFS Server Recovery Kit.

If you cannot use additional servers: This is the least desirable option. However, if you decide to run your NFS server in the same cluster as your DB2 multiple partitions, the NFS export point should be protected with the LifeKeeper NFS Server Recovery Kit. You should note that LifeKeeper currently is not aware of the relationship between the DB2 partitions and the NFS server managing the DB2 partitions. Therefore, you must follow these manual procedures before stopping or starting LifeKeeper on any node in the cluster

1. If you wish to stop LifeKeeper on a single server, you *must* make sure that the NFS server is active on another server in the cluster. Failure to do this may cause the LifeKeeper shutdown to hang trying to take the DB2 partitions out of service. Generally, you should make sure that all DB2 partitions are either switched to another server or manually taken out of service before you stop LifeKeeper to ensure you don't have problems trying to restart LifeKeeper.
2. To shut down the entire cluster, you should manually take all DB2 partition resources out of service. Next, take all the DB2 NFS server resources out of service, and finally shut down LifeKeeper.
3. If you remembered to take the DB2 resource out of service before shutting down LifeKeeper, you should be able to restart LifeKeeper normally. Then bring the NFS server resources into service, followed by any DB2 partitions you wish to restart.
4. If you forgot to take the DB2 partition out of service before shutting down LifeKeeper, you *must* make sure that the NFS server resources for that partition are active elsewhere in the cluster *before* you restart LifeKeeper.

Configuration Requirements

To ensure proper operation of the DB2 Recovery Kit in a multiple partition environment, LifeKeeper requires the following:

1. If you cannot use an additional cluster for your NFS hierarchy, be aware that the LifeKeeper for Linux DB2 Recovery Kit restricts the occurrence of active inodes on an underlying NFS-protected file system. Therefore, to prevent this condition, we recommend that users protect the top-level directory and export the instance home directory using the fully qualified directory name. The top-level directory is protected in order to prohibit users from changing directories directly into it (i.e. `cd <top level dir>`).
2. Verify the installation of IBM's latest Fix Pack (for EEE deployments) as described in the Software Requirements section of this document.
3. Ensure that the hostname value in your `db2nodes.cfg` file is the same as the value returned from issuing the **hostname** command.

Example:

`db2nodes.cfg` file:

```
0 server1.sc.steeleye.com 0
```

Additionally, the hostname value in your server's */etc/hosts* file must be the same as the hostname value in your *db2nodes.cfg* file.

You must also verify that your server's */etc/hosts* file contains both the local hostname and the fully qualified hostname for each server entry included in the file.

Example:

/etc/hosts file

```
127.0.0.1 localhost localhost.localdomain
9.21.55.53 server1.sc.steeleye.com server1
```

4. During execution of the *db2setup* script, **do not** opt to create the DB2 Warehouse Control Database (DWCNTRL) or the DB2 Sample Database at this time. The databases need to be created on a shared file system to ensure successful creation of the DB2 resource hierarchy. Electing to create either of these databases during execution of the *db2setup* script will cause the database to be created in the home directory and not on a shared file system. Users wishing to create these databases should do so external to the *db2setup* script in order to specify a shared file system.

In versions later than 8.1, the DB2 Tools Catalog should not be created during the setup script. This database must be placed on a shared file system and should be created after setup has completed and prior to hierarchy creation, if necessary.

5. In Active/Active or multiple partition server environments, each server in the configuration must be capable of running all database instances in a failover scenario. Please see the *IBM Getting Started Guide* for help determining the maximum number of DB2 instances or partition servers feasible for a given set of system resources.
6. Select or create a shared file system, then export this file system. (i.e. */export/db2home*). The file system will be used as the DB2 instance home.
7. Protect your exported file system by creating a LifeKeeper NFS resource hierarchy. The file system should be included as a dependent resource in your NFS hierarchy.
8. NFS mount the shared file system on each server in the cluster including the server where it is being exported. See the *DB2 Quickstart Guide* for mount options. When creating the DB2 instance, the home directory of the instance must be located on the NFS mounted file system. Make certain that the file system is mounted using the LifeKeeper protected switchable IP address used when creating the NFS hierarchy. Additionally, the mount point of the home directory must be specified in the */etc/fstab* file on all servers in your LifeKeeper cluster. Each server in your configuration must have the file system mounted on identical mount points (i.e. */db2/home*).

NOTE: We recommend that you create and test your NFS hierarchy prior to creating your DB2 resource hierarchy. Please see the *LifeKeeper for Linux NFS Server Recovery Kit Administration Guide* for complete instructions on creating and testing a NFS hierarchy

9. For all servers in your configuration, set the following DB2 environment variable to equal the total number of partitions in the instance. To set this variable, log on to the server as the instance owner and issue a **db2set** command. Adjusting this variable will accommodate all conceivable failover scenarios.

```
db2set DB2_NUM_FAILOVER_NODES=<partitions in the instance>
```

10. Update your existing DB2 instances and your DB2 Administration servers using the following DB2 utilities:

db2iupdt and *dasupdt*

11. A LifeKeeper DB2 hierarchy must be created on each server in the cluster that has a database partition server managing data for the instance. The databases and tablespaces must be on a shared file system. A separate LUN is required for each database partition server and for the NFS exported home directory. Dependent resources include the file systems where actual databases and tablespaces are located.
12. If you create a database on a non-protected LifeKeeper file system after the creation of your DB2 hierarchy, you will need to create a resource hierarchy for that file system and make it a dependency of your DB2 resource hierarchy. The hierarchy will protect all of the partition servers that the *db2node.cfg* file indicates should run on the server.
13. To ensure proper execution of a failover, it is imperative that the file system of each database partition server is uniquely numbered.

Example:

The mount point for your database partition server *node0* should be:

/<FSROOT>/<db2instancename>/NODE0000

The mount point for your database partition server *node1* should be:

/<FSROOT>/<db2instancename>/NODE0001

Note: In this example there are two partition servers, and the file system for each is mounted on a separate LUN.

14. All database partition servers for a given machine must be running in order to assure the successful creation of your DB2 hierarchy.
15. When the database partition server becomes inoperable on the primary system, the service fails over to a previously defined backup system. The database service on the backup system becomes available immediately after the dependent resources fail over and the database partition server(s) is brought into service. Previously connected DB2 clients are disconnected and must reconnect to the functioning server. Any uncommitted SQL statements are rolled back and should be re-entered.

Configuration Considerations for All DB2 configurations

1. DB2 instance names should contain alphanumeric characters only.
2. DB2 clients should be configured to connect to the database via a LifeKeeper protected IP address. Users can define:
“DB2SYSTEM=<Floating IP>” in \$instancehome/sqllib/profile.env
and catalog the floating IP address on the clients.
3. The */etc/services* file for each server in your configuration protecting a DB2 resource hierarchy must have identical service entries for the protected instance. Additionally, the User ID, Group ID and instance home directory for the protected DB2 instance must be the same on all servers where the resource will be protected.

DB2 adds the following entries as default in */etc/services*:

```
DB2_db2inst1      60000/tcp
DB2_db2inst1_1   60001/tcp
DB2_db2inst1_2   60002/tcp
DB2_db2inst1_END 60003/tcp
db2c_db2inst1    50001/tcp
```

4. A recovery is what takes place after DB2 is terminated abruptly, as with a system crash. Following are tips that will significantly reduce the amount of time it takes for DB2 to recover from a failure.
 - Limit the log records that DB2 will process. You can accomplish this by properly configuring the **SOFTMAX** and **LOGFILSIZ** configuration parameters. You should use log files with a size of 4MB (1000 4KB pages) and keep the amount of active log space at 25% of the size of one log file (1MB):

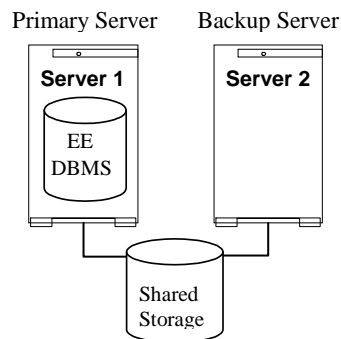

```
db2 UPDATE DB CFG FOR <dbname> USING SOFTMAX 25
db2 UPDATE DB CFG FOR <dbname> USING LOGFILSIZ 1000
```
 - Ensure that there is a sufficient number of page cleaners to accommodate your work load:


```
db2 UPDATE DB CFG FOR <dbname> USING NUM_IOCLEANERS <num>
```
5. DB2 Fault Monitor should be disabled in all servers.
6. DB2 should be installed in all servers.

Configuration Examples

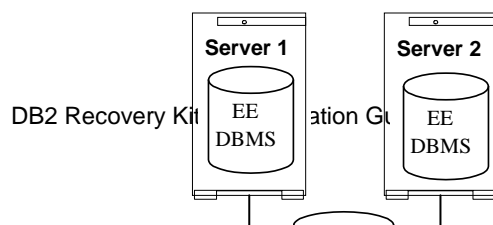
A few examples of what happens during a failover using LifeKeeper for Linux DB2 Recovery Kit are provided below. In the following pictures, EE and EEE are used to denote database configurations; ESE may be substituted wherever appropriate.

Configuration 1: DB2 Single Partition Active/Standby Configuration



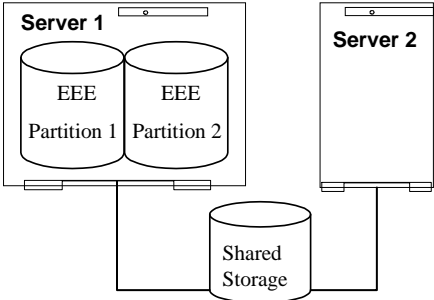
The DB2 instance is protected on Server 1. Server 2 will assume the DB2 resources when a failure occurs.

Configuration 2: DB2 Single Partition Active/Active Configuration



One DB2 instance is protected on Server 1 and another DB2 instance is protected on Server 2. Each server will assume the other's resources when a failure occurs.

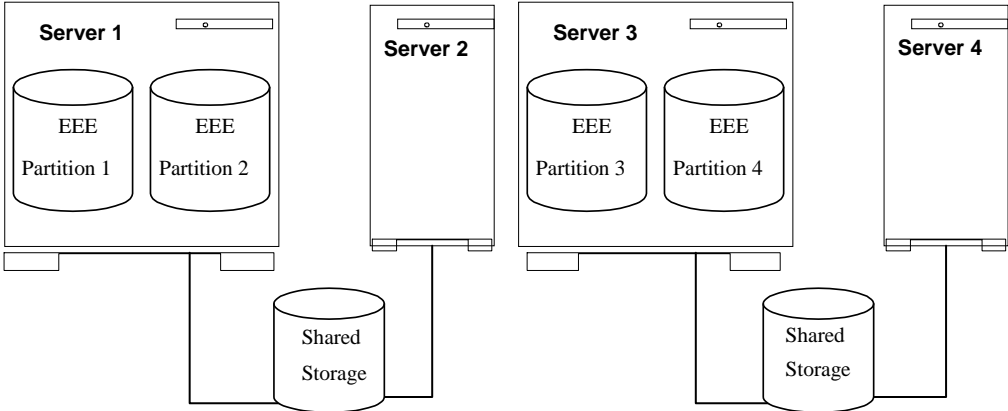
Configuration 3: DB2 Multiple Partition Active/Standby (1 Cluster)



One DB2 instance with two database partition servers is protected on Server 1 with one LifeKeeper DB2 resource hierarchy. Server 2 will assume ownership of the DB2 resource hierarchy when a failure occurs.

Note: For all cluster of cluster configurations listed in the following section, users should be aware that the cluster of cluster configuration is protecting only one DB2 instance with multiple partitions on multiple physical nodes.

Configuration 4: DB2 Multiple Partition Active/Standby (Cluster of Clusters)

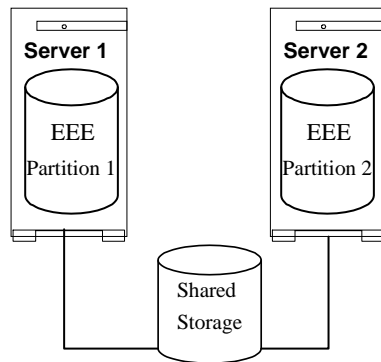


One DB2 instance with two database partition servers is protected on

Server 1 and two database partition servers protected on Server 3. There is one LifeKeeper DB2 resource hierarchy on Server 1, extended to Server 2, and another DB2 resource hierarchy on Server 3 extended to Server 4. When a failure occurs on Server 1, Server 2 will assume its resource. When a failure occurs on Server 3, Server 4 will assume its resource.

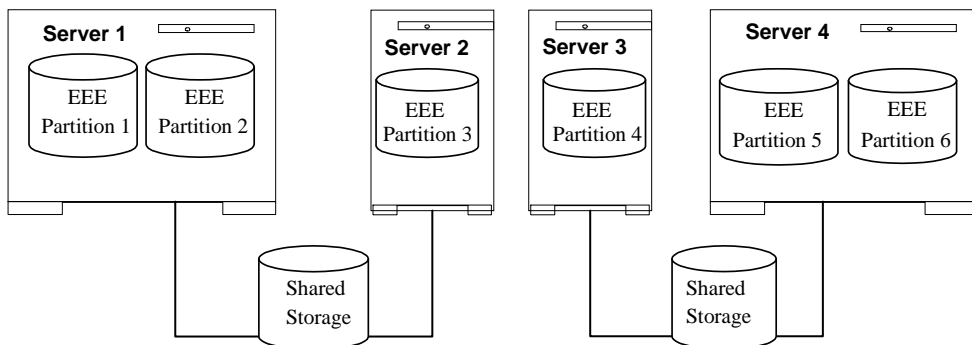
If the server that is exporting the DB2 instance home directory and its backup server become inoperable at once, the DB2 database is inaccessible. In addition, if the NFS hierarchy for the exported DB2 instance directory (primary and all backups) become inoperable at the same time, the DB2 database will be inaccessible until the NFS hierarchy can be restored.

Configuration 5: DB2 Multiple Partitions Active/Active (1 Cluster)



One DB2 instance with one database partition server is protected on Server 1 and one database partition server protected on Server 2. There is one LifeKeeper DB2 resource hierarchy on Server 1 and another DB2 resource hierarchy on Server 2. When a failure occurs each server will assume the other's resources.

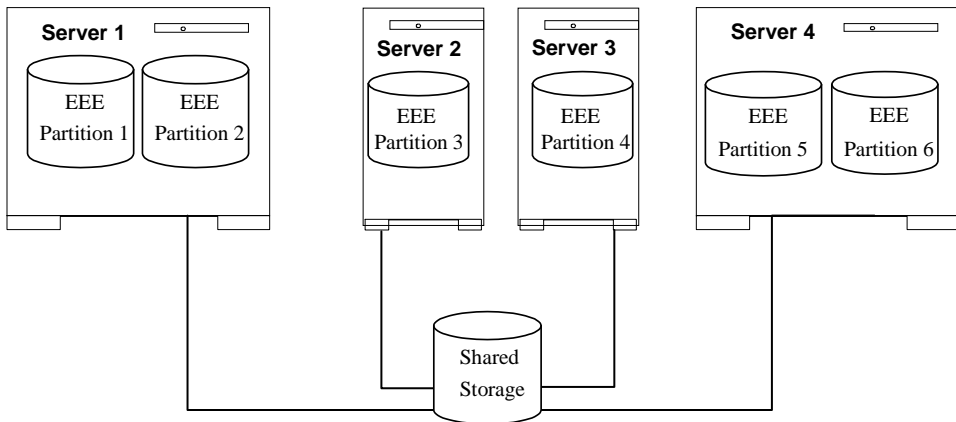
Configuration 6: DB2 Multiple Partitions Active/Active (Cluster of Clusters)



One DB2 instance with two database partition servers is protected on Server 1, one database partition server protected on Server 2, one database partition server protected on Server 3 and two database partition servers protected on Server 4. There is one LifeKeeper DB2 resource hierarchy on each server in the cluster. Upon failure, Server 1 and Server 2 assume each other's resources and Server 3 and Server 4 assume each other's resources.

If the server that is exporting the DB2 instance home directory and its backup server become inoperable at once, the DB2 database is inaccessible. In addition, if the NFS hierarchy for the exported DB2 instance directory (primary and all backups), become inoperable at the same time, the DB2 database will be inaccessible until the NFS hierarchy can be restored.

Configuration 7: DB2 Multiple Partition (4 Node Fibre Channel Cluster)



One DB2 instance with two database partition servers is protected on Server 1, one database partition server protected on Server 2, one database partition server protected on Server 3 and two database partition servers protected on Server 4. There is one LifeKeeper DB2 resource hierarchy on each server in the cluster. Each server in the cluster provides backup protection for the other in the event of failure.

If the server that is exporting the DB2 instance home directory and its backup server become inoperable at once, the DB2 database is inaccessible. In addition, if the NFS hierarchy for the exported DB2 instance directory (primary and all backups), become inoperable at the same time, the DB2 database will be inaccessible until the NFS hierarchy can be restored.

LifeKeeper Configuration Tasks

You can perform all LifeKeeper for Linux DB2 Recovery Kit administrative tasks via the LifeKeeper Graphical User Interface (GUI). The LifeKeeper GUI provides a guided interface to configure, administer, and monitor DB2 resources.

Overview

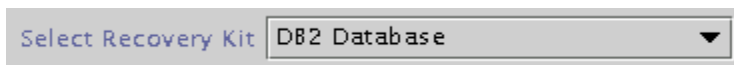
The following tasks are available for configuring the LifeKeeper for Linux DB2 Recovery Kit:

- **Create a Resource Hierarchy** - Creates a DB2 resource hierarchy
- **Delete a Resource Hierarchy** - Deletes a DB2 resource hierarchy
- **Extend a Resource Hierarchy** - Extends a DB2 resource hierarchy from the primary server to the backup server.
- **Unextend a Resource Hierarchy** - Unextends (removes) a DB2 resource hierarchy from a single server in the LifeKeeper cluster.
- **Create Dependency** - Creates a child dependency between an existing resource hierarchy and another resource instance and propagates the dependency changes to all applicable servers in the cluster.
- **Delete Dependency** - Deletes a resource dependency and propagates the dependency changes to all applicable servers in the cluster.
- **In Service** - Activates a resource hierarchy.
- **Out of Service** - Deactivates a resource hierarchy.
- **View/Edit Properties** - View or edit the properties of a resource hierarchy.

Creating a DB2 Resource Hierarchy

Perform the following on your primary server:

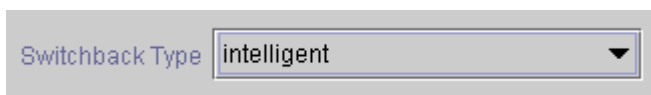
1. Select **Edit > Server > Create Resource Hierarchy**.
2. The “**Select Recovery Kit**” dialog appears. Select the **DB2 Database** option from the drop down list.



Click **Next** to continue.

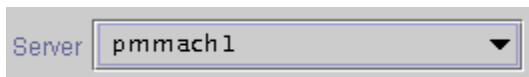
CAUTION: If you click the **Cancel** button at any time during the sequence of creating your hierarchy, LifeKeeper will cancel the entire creation process.

3. The “**Switchback Type**” dialog appears. The switchback type determines how the DB2 resource will be switched back to the primary server when it becomes in-service (active) on the backup server following a failover. Switchback types are either *intelligent* or *automatic*. Intelligent switchback requires administrative intervention to switch the resource back to the primary server while automatic switchback occurs as soon as the primary server is back on line and reestablishes LifeKeeper communication paths.



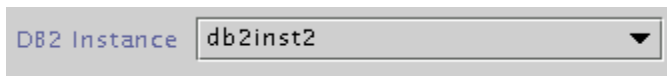
Click **Next** to continue.

4. The “**Server**” dialog appears. Select the name of the server where the DB2 resource will be created (typically this is your primary server). All servers in your cluster are included in the drop down list box.



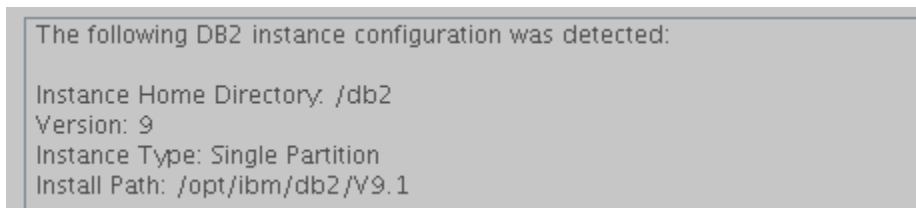
Click **Next** to continue.

5. The “**DB2 Instance**” dialog appears. Select or enter the name of the **DB2** instance that is being protected.



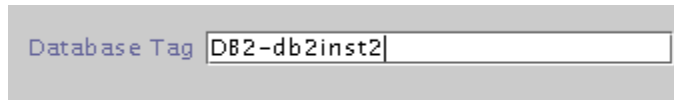
Click **Next** to continue.

6. An information box appears displaying information regarding the instance detected.



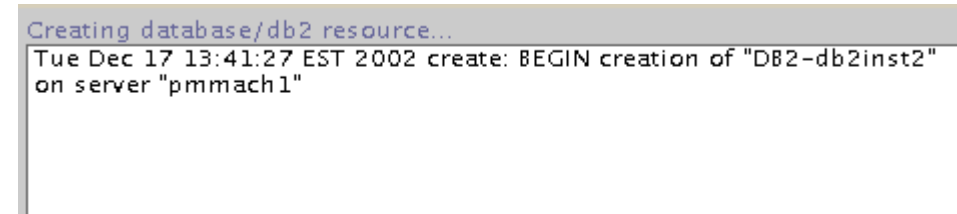
Click **Continue**.

- The “**Database Tag**” dialog appears. This dialog is populated automatically with a unique tag name for the new DB2 database resource instance.



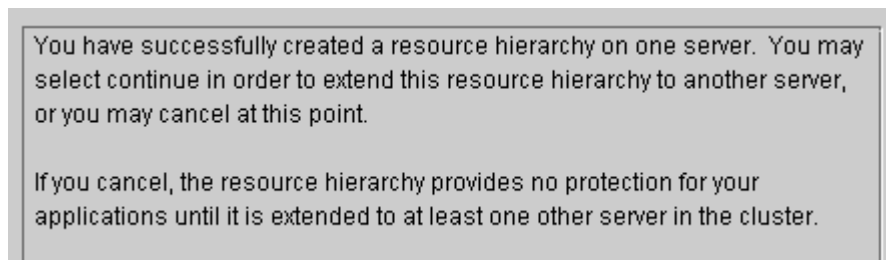
Click **Create** to continue.

- An information box appears indicating the start of the hierarchy creation.



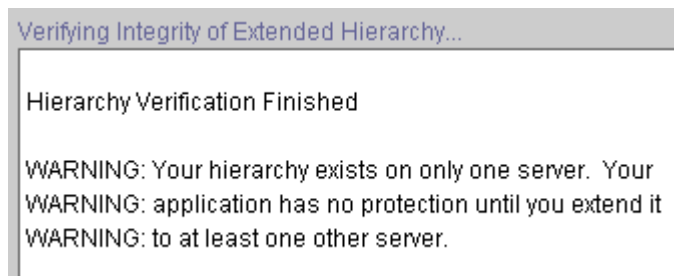
Click **Next** to continue.

- An information box appears announcing the successful creation of your DB2 resource hierarchy. You must **Extend** the hierarchy to another server in your cluster in order to place it under LifeKeeper protection.



Click **Continue** to extend the resource.

Click **Cancel** if you wish to extend your resource at another time.



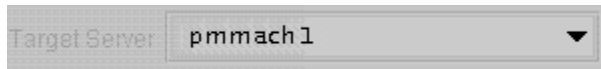
- Click **Done** to exit the Create Resource Hierarchy menu selection.

Deleting a Resource Hierarchy

To delete a DB2 resource from all servers in your LifeKeeper configuration, complete the following steps:

1. From the LifeKeeper GUI menu, select **Edit**, then **Resource**. From the drop down menu, select **Delete Resource Hierarchy**.
2. Select the name of the **Target Server** where you will be deleting your DB2 resource hierarchy.

Note: If you selected the Delete Resource task by right-clicking from either the left pane on a global resource or the right pane on an individual resource instance, this dialog will not appear.



Click **Next** to continue.

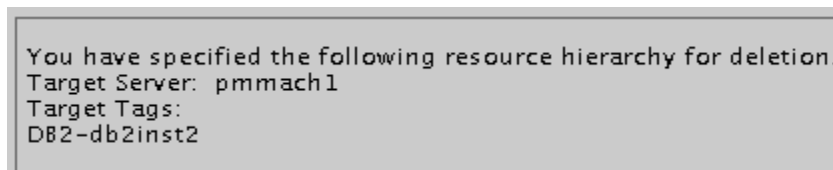
3. Select the **Hierarchy to Delete**. Identify the resource hierarchy you wish to delete, and highlight it.

Note: If you selected the Delete Resource task by right-clicking from either the left pane on a global resource or the right pane on an individual resource instance, this dialog will not appear.



Click **Next** to continue.

4. An information box appears confirming your selection of the target server and the hierarchy you have selected to delete.



Click **Delete** to continue.

5. An information box appears confirming that the DB2 resource instance was deleted successfully.

Deleting resource hierarchy...

```

Successfully removed
ins_remove[701,Iraci.C]Thu Jun 1 07:06:54 EDT 2000:
    fletch,priv_globact(1,delete): Running Post Global delete
    Machine cornfed
ins_remove[714,Iraci.C]Thu Jun 1 07:06:56 EDT 2000:
    fletch,priv_globact(1,delete): Post Global delete Scripts F
    Exiting 0 On Machine cornfed With Output Following:
lcrecover[701,Iraci.C]Thu Jun 1 07:12:15 EDT 2000:

```

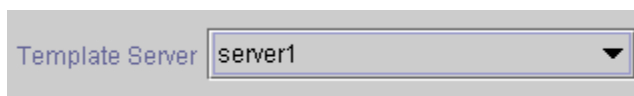
6. Click **Done** to exit the Delete Resource Hierarchy menu selection.

Extending Your Hierarchy

After you have created a hierarchy, you should extend that hierarchy to another server in the cluster. There are three possible ways to extend your resource instance:

1. When you successfully create your DB2 resource hierarchy you will have an opportunity to select **Continue** which will allow you to proceed with extending your resource hierarchy to your backup server.
2. Right-click on an unextended hierarchy in either the left or right pane on the LifeKeeper GUI.
3. Select the “Extend Resource Hierarchy” task from the LifeKeeper GUI by selecting **Edit, Resource, Extend Resource Hierarchy** from the drop down menu. This sequence of selections will launch the Extend Resource Hierarchy wizard. The **Accept Defaults** button that is available for the **Extend Resource Hierarchy** option is intended for the user who is familiar with the LifeKeeper **Extend Resource Hierarchy** defaults and wants to quickly extend a LifeKeeper resource hierarchy without being prompted for input or confirmation. Users who prefer to extend a LifeKeeper resource hierarchy using the interactive, step-by-step interface of the GUI dialogs should use the **Next** button.
 - a. The first dialog box to appear will ask you to select the **Template Server** where your DB2 resource hierarchy is currently in service. Remember that the **Template Server** you select now and the **Tag to Extend** that you select in the next dialog box represent an in-service (activated) resource hierarchy. An error message will appear if you select a resource tag that is not in service on the template server you have selected. The drop down box in this dialog provides the names of all the servers in your cluster.

Note: If you are entering the Extend Resource Hierarchy task by continuing from the creation of a DB2 resource hierarchy, this dialog box will not appear because the wizard has already identified the template server in the create stage. This is also the case when you right-click on either the DB2 resource icon in the left pane or right-click on the DB2 resource box in the right pane of the GUI window and choose *Extend Resource Hierarchy*.

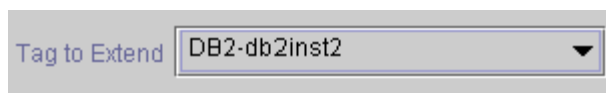


CAUTION: If you click the **Cancel** button at any time during the sequence of extending your hierarchy, LifeKeeper will cancel the extend hierarchy process. However, if you have already extended the resource to another server, that instance will continue to be in effect until you specifically unextend it.

Click **Next** to continue.

- b. Select the **Tag to Extend**. This is the name of the DB2 instance you wish to extend from the template server to the target server. The wizard will list in the drop down box all of the resources that you have created on the template server.

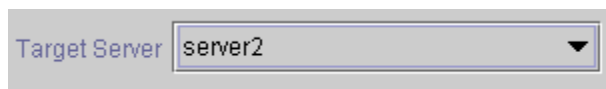
Note: Once again, if you are entering the Extend Resource Hierarchy task immediately following the creation of a DB2 hierarchy, this dialog box will not appear because the wizard has already identified the tag name of your resource in the create stage. This is also the case when you right-click on either the DB2 resource icon in the left pane or on the DB2 resource box in the right pane of the GUI window and choose *Extend Resource Hierarchy*.



A screenshot of a GUI window showing a dropdown menu labeled 'Tag to Extend'. The selected item is 'DB2-db2inst2'.

Click **Next** to continue.

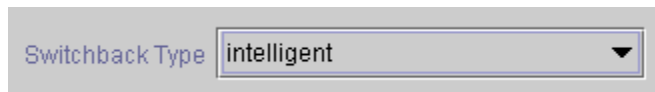
- c. Select the **Target Server** where you will extend your DB2 resource hierarchy.



A screenshot of a GUI window showing a dropdown menu labeled 'Target Server'. The selected item is 'server2'.

Click **Next** to continue.

- d. The **Switchback Type** dialog appears. The switchback type determines how the DB2 resource will be switched back to the primary server when it becomes in service (active) on the backup server following a failover. Switchback types are either *intelligent* or *automatic*. Intelligent switchback requires administrative intervention to switch the resource back to the primary server while automatic switchback occurs as soon as the primary server is back on line and reestablishes LifeKeeper communication paths.



A screenshot of a GUI window showing a dropdown menu labeled 'Switchback Type'. The selected item is 'intelligent'.

Click **Next** to continue.

- e. Select or enter a **Template Priority**. This is the priority for the DB2 hierarchy on the server where it is currently in service. Any unused priority value from 1 to 999 is valid, where a lower number means a higher priority (1=highest). The extend process will reject any priority for this hierarchy that is already in use by another system. The default value is recommended. **Note:** This selection will appear only for the initial extend of the hierarchy.

Click **Next** to continue.

- f. Select or enter the **Target Priority**. This is the priority for the new extended DB2 hierarchy relative to equivalent hierarchies on other servers. Any unused priority value from 1 to 999 is valid, indicating a server's priority in the cascading failover sequence for the resource. A lower number means a higher priority (1=highest). Note that LifeKeeper

assigns the number “1” to the server on which the hierarchy is created by default. The priorities need not be consecutive, but no two servers can have the same priority for a given resource.

Target Priority 10

Click **Next** to continue.

- g. An information box appears explaining that LifeKeeper has successfully checked your environment and that all requirements for extending this resource have been met. If there are requirements that have not been met, LifeKeeper will disable the **Next** button, and enable the **Back** button.

```
Executing the pre-extend script...
Checking existence of extend and canextend scripts
Building independent resource list
Checking extendability for DB2-db2inst2

Pre Extend checks were successful
```

Click on the **Back** button to make changes to your resource extension.

Click **Cancel** to extend your resource another time.

Click **Next** to launch the Extend Resource Hierarchy configuration task.

Click **Finish** to confirm the successful extension of your DB2 resource instance.

- Click **Done** to exit the Extend Resources Hierarchy menu selection.

Note: Be sure to test the functionality of the new instance on *both* servers.

Unextending Your Hierarchy

- From the LifeKeeper GUI menu, select **Edit, Resource, and Unextend Resource Hierarchy**.
- Select the **Target Server** where you want to unextend the DB2 resource. It cannot be the server where the resource is currently in service (active).

Note: If you selected the Unextend task by right-clicking from either the left pane on a global resource or the right pane on an individual resource instance, this dialog will not appear.

Target Server pmmach2

Click **Next** to continue.

- Select the DB2 **Hierarchy to Unextend**.

Note: If you selected the Unextend task by right-clicking from either the left pane on a global resource or the right pane on an individual resource instance, this dialog will not appear.

Hierarchy to Unextend DB2-db2inst2

Click **Next** to continue.

4. An information box appears confirming the target server and the DB2 resource hierarchy you have chosen to unextend.

```
You have specified the following resource hierarchy for unextend.  
Target Server = pmmach2  
Target Tag = DB2-db2inst2
```

Click **Unextend**.

5. Another information box appears confirming that the DB2 resource was unextended successfully.
6. Click **Done** to exit the Unextend Resource Hierarchy menu selection.

Testing Your Resource Hierarchy

You can test your DB2 resource hierarchy by initiating a manual switchover that will simulate a failover of the resource instance from the primary server to the backup server.

Performing a Manual Switchover from the LifeKeeper GUI

You can initiate a manual switchover from the LifeKeeper GUI by selecting **Edit**, **Resource**, and **In Service**. For example, an in-service request executed on a backup server causes the DB2 resource hierarchy to be placed in-service on the backup server and taken out-of-service on the primary 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 resource hierarchy is taken out-of-service without bringing it in-service on the other server. The resource can only be brought in-service on the same server, if it was taken out-of-service during resynchronization.

Important: After bringing your resource hierarchy in service on the backup server, you should attempt to connect to the databases, especially when using raw devices as tablespace containers. This is necessary to ensure that all disk partitions are visible on the backup servers and the raw bindings are being established correctly.

If the raw bindings have not been established on the backup servers, it is most likely caused by the fact that new partitions were created on the primary server and added to the configuration, but the partition tables have not yet been updated on the backup servers.

The solution is to reboot the backup servers so that the partition tables are updated correctly.

Troubleshooting

Symptom	Possible Cause
One or more of your DB2 EEE partition servers fail to start	The <i>db2nodes.cfg</i> file's port number may have erroneously outgrown the range set in the <i>/etc/services</i> file. View the number of ports set in the <i>db2nodes.cfg</i> file and ensure that the ports range value in the <i>/etc/services</i> file is large enough to accommodate.
LifeKeeper "In-Service" or "Out-of-Service" operation hangs.	The DB2 environment variable: DB2_NUM_FAILOVER_NODES may not have been properly set. Ensure that for all servers in your configuration, this environment variable is set to equal the total number of partitions in the instance. EXAMPLE: db2set DB2_NUM_FAILOVER_NODES=<partitions in instance>
LifeKeeper "In-Service" operation hangs.	The dasupdt command may not have been executed on the DB2 Administration server. Ensure that the dasupdt command was successfully executed on the DB2 Administration server.
LifeKeeper First Switch over operation fails	The DB2 Fenced User may not have been created on the backup server. Verify the DB2 Fenced User for the specified instances exists with the same user and group id for the primary. Ensure that the protected instance is also a member of the Administration Server group.
You need to add a new node to your existing DB2 resource hierarchy	Please see the nodes utility man page for complete instructions on adding a new node to your currently existing LifeKeeper DB2 resource hierarchy.
Administration Server fails to start	Verify another Administration Server is not already running on specified port.

Error Messages

The **Message Catalog** (located on our Technical Documentation site under “Search for an Error Code” -- <http://docs.us.sios.com/Linux/8.1.1/LK4L/MessageCatalogs/MessageCatalog/>) provides a listing of all messages that may be encountered while using SIOS Protection Suite for Linux and, where appropriate, provides additional explanation of the cause of the errors and necessary action to resolve the error condition. This full listing may be searched for any error code received, or you may go directly to the **DB2 Kit Message Catalog** (<http://docs.us.sios.com/Linux/8.1.1/LK4L/MessageCatalogs/DB2MessageCatalog/>) which contains a listing of all messages that may be encountered while utilizing the DB2 Recovery Kit.

Appendix: Setting up DB2 to use Raw I/O

There are several requirements for configuring RAW I/O devices for DB2 so that the DB2 instance can be protected by LifeKeeper.

Requirements

- The Linux OS must support Raw I/O devices. For most distributions this support was included in the 2.4 kernel, but there are some distributions that support Raw I/O on a 2.2 kernel.
- All Raw I/O devices must be bound to a shared disk partition. A number of shared SCSI disk partitions is required. The exact number is determined by the number of tablespaces that will be located on Raw I/O devices. (Please see to DB2 documentation for guidelines for writing tablespaces on raw devices).
- DB2 Version 7.1 Fix Pack 3 or later OR DB2 Version 8 or higher is required.

Naming Conventions

The naming of the raw device and controller varies by Linux distribution.

- On Red Hat the device name is `/dev/raw/raw<number>` and the controller is `/dev/rawctl`.
- On SuSE the name of the device is `/dev/raw<number>` and the controller varies between `/dev/raw`, `/dev/rawctl`, and `/dev/raw/rawctl`

Raw I/O Setup Steps

The following steps 1-4 were taken from Section 7.3.1.1 (“Using Raw I/O on Linux”) of the *IBM DB2 Universal Database Release Notes Version 7.2/Version 7.1 Fix Pack 3*. In this example, the raw partition to be used is `/dev/sda5`. It should not contain any valuable data.

Note that step 4 or 5 will vary depending upon whether you are using Multiple Logical Nodes.

1. Calculate the number of 4 096-byte pages in this partition, rounding down if necessary. For example:

```
# fdisk /dev/sda
Command (m for help): p

Disk /dev/sda: 255 heads, 63 sectors, 1106 cylinders
Units = cylinders of 16065 * 512 bytes

Device Boot   Start     End   Blocks   Id  System
/dev/sda1      1        23   4200997   83  Linux
/dev/sda2     524     1106   4682947+    5  Extended
/dev/sda5     524     1106   4682947   83  Linux

Command (m for help): q
#
```

The number of pages in `/dev/sda5` is:

```
num_pages = floor( ((1106-524+1)*16065*512)/4096 )
num_pages = 11170736
```

2. Bind an unused raw device node to this partition. Since this needs to be done every time the machine is rebooted, and requires root access, you may want to add the raw bindings to a system initialization file (i.e. `rc.local` or `boot.local`). **These bindings must be removed once the hierarchy is under LifeKeeper protection.** LifeKeeper will re-establish the raw bindings for Raw I/O devices that are under LifeKeeper protection.

Use `raw -qa` to see which raw device nodes are already in use:

```
raw /dev/raw/raw1 /dev/sda5
/dev/raw/raw1: bound to major 8, minor 5
```

3. Set global read permissions on the raw device controller and the disk partition. Set global read and write permissions on the raw device:

```
# chmod a+r /dev/rawctl
# chmod a+r /dev/sdb1
# chmod a+rw /dev/raw/raw1
```

4. **Important:** This step only applies if you are using DB2 EE OR your DB2 EEE configuration will *never* run Multiple Logical Nodes (MLNs) even after failover. If the configuration *may* run MLNs at some point, proceed to step 5.

Create the tablespace in DB2, specifying the raw device, not the disk partition. For example:

```
CREATE TABLESPACE dms1
MANAGED BY DATABASE
USING (DEVICE '/dev/raw/raw1' 11170736)
```

Tablespaces on raw devices are also supported for all other page sizes supported by DB2.

5. **Important:** This step must be followed if the configuration is running MLNs or will run MLNs at some point after failover.

Create the table space in DB2, specifying the raw device, not the disk partition, and specify a different raw I/O device node for each DB2 instance partition. For example:

```
CREATE TABLESPACE dms1
MANAGED BY DATABASE
USING (DEVICE '/dev/raw/raw1' 11170736) on NODE
(NODENUM)
USING (DEVICE '/dev/raw/<different raw device node>'
##### ) on NODE (NODENUM)
```

Note: ON NODE must be used because each DB2 node (database partition server) must use a different raw I/O device. This must be specified even if the node is running on a different machine so that the failover will work correctly.