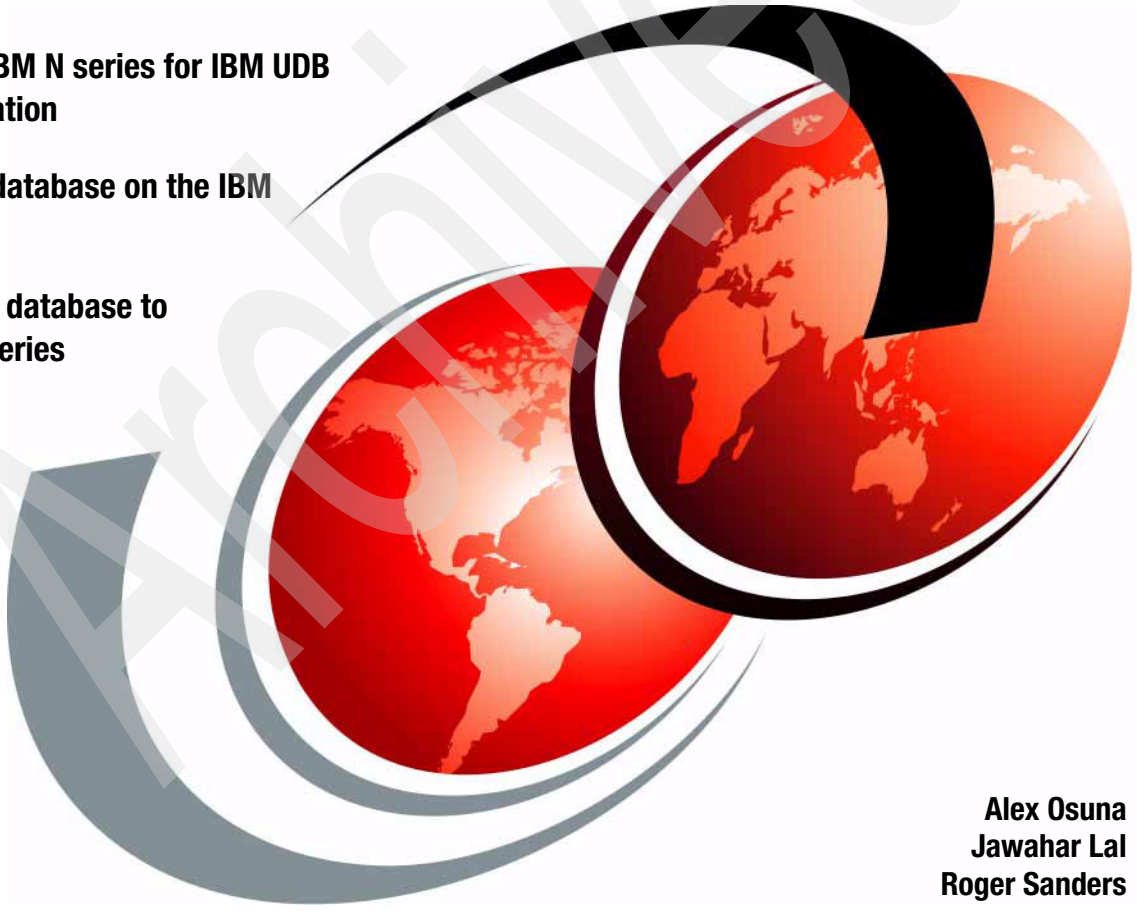
IBM

Integrating IBM DB2 with the IBM System Storage N series

- Preparing IBM N series for IBM UDB DB2 installation
- Creating a database on the IBM N series
- Migrating a database to the IBM N series



Alex Osuna
Jawahar Lal
Roger Sanders

Redbooks

ibm.com/redbooks

International Technical Support Organization

Integrating IBM DB2 with the IBM System Storage N series

September 2006

SG24-7329-00

Note: Before using this information and the product it supports, read the information in “Notices” on page v.

First Edition (September 2006)

This edition applies to Data ONTAP version 7 and above.

Contents

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

Redbooks (logo)  ™	DB2 Universal Database™	Redbooks™
developerWorks®	DB2®	System Storage™
AIX 5L™	IBM®	Tivoli®
AIX®	OS/2®	TotalStorage®

The following terms are trademarks of other companies:

QLogic, ©2004-2005 QLogic Corporation. Specifications are subject to change without notice. All rights reserved worldwide. QLogic, the QLogic logo SANbox, SAN Pro and SAN Express are registered trademarks of QLogic Corporation. SANblade is a registered trademark in the United States. All other brands and product names are trademarks or registered trademarks of their respective owners.

Oracle, JD Edwards, PeopleSoft, Siebel, and TopLink are registered trademarks of Oracle Corporation and/or its affiliates.

Snapshot, Network Appliance, WAFL, SnapRestore, SnapMirror, FilerView, Data ONTAP, NetApp, and the Network Appliance logo are trademarks or registered trademarks of Network Appliance, Inc. in the U.S. and other countries.

Snapshot, NOW, Data ONTAP, Network Appliance, WAFL, SnapRestore, SnapMirror, FilerView, NetApp, NetApp, and NetApp logo are trademarks or registered trademarks of NetApp Corporation or its subsidiaries in the United States, other countries, or both.

NetApp, Snapshot, Data ONTAP, WAFL, SnapRestore, SnapMirror, FilerView, Network Appliance, The Network Appliance logo, the bolt design, Camera-to-Viewer, Center-to-Edge, ContentDirector, ContentFabric, NetApp Availability Assurance, NetApp ProTech Expert, NOW, NOW NetApp on the Web, RoboCache, RoboFiler, SecureAdmin, Serving Data by Design, Smart SAN, The evolution of storage, Virtual File Manager, and Web Filer are trademarks of Network Appliance, Inc. in the U.S. and other countries. All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such.

Solaris, Sun, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Preface

This IBM® Redbook describes the steps carried out to configure and install an IBM DB2® database in a N series storage system UNIX® environment. It particularly elaborates the configuration steps for the UNIX host as well as the IBM N series storage system. The UNIX host types covered in this IBM Redbook are Linux®, IBM AIX®, Sun™ Solaris™, and HP-UX.

The team that wrote this redbook

This IBM Redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

Alex Osuna is a Project Leader at the International Technical Support Organization, Tucson Arizona Center. He writes extensively and teaches IBM classes on all areas of storage. He has over 28 years in the I/T industry working for United States Air Force, IBM, and Tivoli® in computer maintenance, field engineering, service planning, Washington Systems Center, business and product planning, advanced technical support, and Systems Engineering, focusing on storage hardware and software. He has over 10 certifications from IBM, Microsoft® and Red Hat.

Jawahar Lal is a Alliance engineer for IBM DB2 at Network Appliance™ Inc., RTP, North Carolina. He holds a Masters degree in Computer Science from University of Rajasthan, Jaipur (India). Currently he is working on his MBA degree from Kenan Flagler Business School UNC Chapel Hill, North Carolina. He has been involved in the storage and IT industry for over 11 years, in areas of database programming, modeling, designing, administration, performance tuning and storage integration. He writes extensively on storage and database integration. He is an IBM DB2 Certified DBA and holds four other certifications from IBM and Oracle®.

Roger Sanders is a Senior Manager - IBM Alliance Engineering with Network Appliance, Inc. He has been designing and developing databases and database applications for more than 20 years and has been working with DB2 Universal Database™ and its predecessors since it was first introduced on the IBM PC (as part of OS/2® 1.3 Extended Edition). He has written articles for IDUG Solutions Journal and Certification Magazine, authored DB2 tutorials for the IBM developerWorks® Web site, presented at several International DB2 User's Group (IDUG) and regional DB2 User's Group (RUG) conferences, taught

classes on DB2 Fundamentals and Database Administration (DB2 8.1 for Linux, UNIX, and Windows®), and is the author of nine books on DB2 and one book on ODBC. Roger is also a member of the DB2 Certification Exam development team and the author of a regular column (Distributed DBA) in DB2 Magazine.

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You will have the opportunity to team with IBM technical professionals, Business Partners, and Clients.

Your efforts will help increase product acceptance and client satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our Redbooks™ to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- Use the online **Contact us** review redbook form found at:

  ibm.com/redbooks

- Send your comments in an e-mail to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

1

Introduction

IBM DB2 UDB Enterprise Server Edition Version V8 is validated by Network Appliance (NetApp®) and certified by IBM to run on all the platforms that IBM supports for DB2 and for all the protocols that NetApp and the IBM Storage System N series supports. This IBM Redbook describes the steps to configure and install an IBM DB2 UDB database in an N series storage system UNIX environment. It particularly elaborates the configuration steps for the UNIX host as well as the IBM N series storage system. The UNIX host types covered in this document are Linux, IBM AIX, Sun Solaris, and HP-UX. The topics covered will include:

- Preparing the UNIX server for a DB2 UDB installation
- Preparing the IBM N series storage system for DB2 UDB database storage
- Installing DB2 UDB on a UNIX server and creating a database on an N series storage system
- Storing database transaction log files on an N series storage system
- Migrating an existing database from a local disk to an N series storage system

1.1 The configuration used for this IBM Redbook

For this IBM Redbook, IBM DB2 UDB V8.x was already deployed in a functioning UNIX and N series storage system environment. Information on the installation of UNIX-type OSes can be found on the respective software manufacturers' Web sites. The installation documentation for the IBM N series storage system can be found on the IBM support site. The examples provided are on an *as is* basis and may require modifications before they can be run under your version of UNIX.

We recommend that you read the DB2 UDB installation documentation for your particular operating system and follow its steps to install and configure the DB2 UDB instance and database.

The information presented in this IBM Redbook has been tested by Network Appliance using only a limited set of hardware and software options. Therefore, your experience and results may differ from that presented here.

1.2 Sample scripts

The sample scripts in this IBM Redbook are based on the following information:

- The name of the IBM N series storage system is itsotuc.
- The name of the UNIX host, henceforth known as a database server, is db2srv1.
- The user name for the DB2 instance owner is db2inst1.
- The DB2 Instance on the UNIX server is db2.
- The home directory for the DB2 Instance is /home/db2inst1.
- The location of the database data files on the IBM N series storage system is itsotuc:/vol/db2data.
- The location of the database transactions log files on the IBM N series storage system is itsotuc:/vol/db2logs.
- The mount point on the UNIX server for the volume itsotuc:/vol/db2data is /mnt/db2data.
- The mount point on the UNIX server for the volume itsotuc:/vol/db2logs is /mnt/db2logs.

You may need to make the appropriate changes to one or more of these settings in order to make these scripts work in your environment.

2

Requirements

In order to use DB2 UDB ESE for UNIX in conjunction with an IBM N series storage system, the following elements are required:

- IBM DB2 UDB Enterprise Server Edition V8 or higher
- One or more UNIX Host servers (database servers) with DB2 Administrator and user accounts
- One or more IBM N series storage systems running Data ONTAP® 7G (or later)
- Network connectivity between database server and IBM N series storage system
- FCP or iSCSI HBA (not required when NFS or an iSCSI software initiator is used)

2.1 DB2 UDB ESE V8

In order to perform the steps outlined in this IBM Redbook, you need to have IBM DB2 UDB V8.x ESE installed and running on a UNIX server or workstation. Before proceeding with the installation of IBM DB2 UDB ESE, check the system requirements described in the DB2 installation documentation for your host platform.

2.2 Database server with DB2 administrator and user accounts

In order to perform the steps outlined in this IBM Redbook, you will need to have both a DB2 administrator and one or more DB2 user accounts on the UNIX server. If you already have DB2 UDB installed on your UNIX server, chances are that the appropriate DB2 user accounts already exist. If, however, you are installing DB2 UDB for the first time, a DB2 administrator and one or more user accounts can be created as part of the DB2 UDB installation process. Refer to the IBM DB2 UDB documentation for detailed instructions on setting up the appropriate accounts during the installation process.

2.3 IBM N series storage system

IBM DB2 UDB stores data in tablespace containers. A tablespace container, which can be a file or a raw device, can physically reside on a IBM N series storage system. Any IBM N series storage system can be used to store a DB2 UDB database and its transaction log files.

2.4 Network

In order to store a DB2 UDB database on a IBM N series storage system, you must first establish a network connection between the DB2 database server and the IBM N series storage system. The type of connection you establish is dependant upon the type of environment you are using (see Figure 2-1 on page 5). If your environment is SAN with Fibre Channel (FC), then you will need to establish a FC connection between the database server and the IBM N series storage system; otherwise, an Ethernet connection will work for both IP SAN and NAS. We used two Gigabit Ethernet connections between our AIX server and the IBM N series storage system (connections were made using two cross-over fiber optic cables).

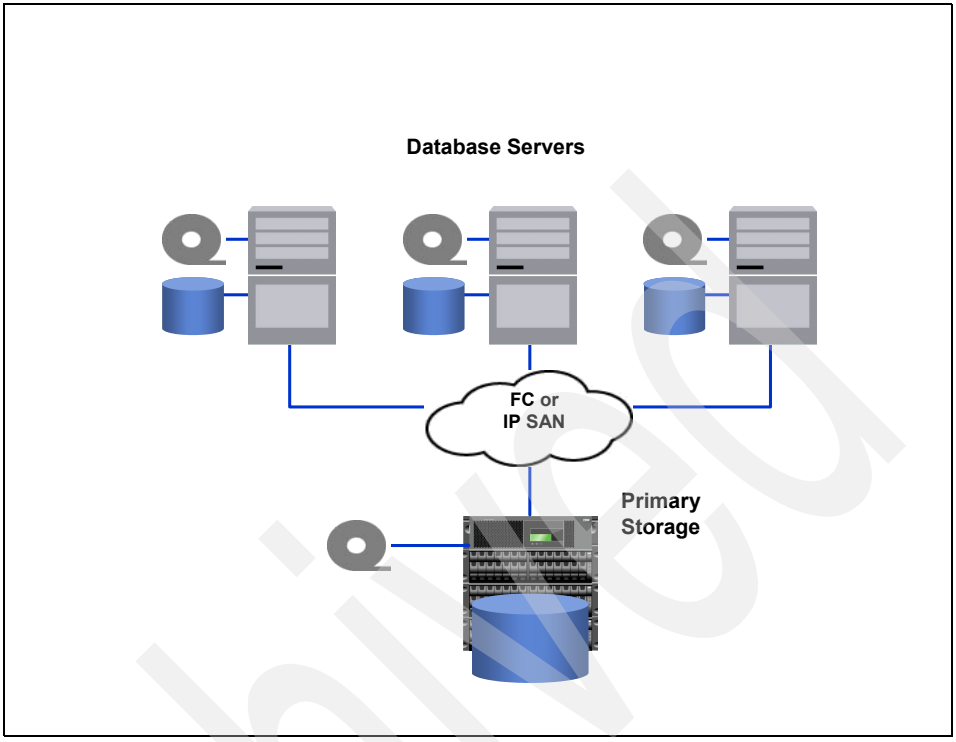


Figure 2-1 Network connections

2.5 Network performance notes

- Flow control and full-duplex operation should be enabled on the Ethernet interfaces on both the UNIX server and the IBM N series storage system.
- When using Gigabit Ethernet adapters, consider turning on jumbo frames.
- Consider using local locking for all mount points that communicate with the IBM N series storage system.
- If possible, dedicate at least one private network for the connection between the DB2 server and the IBM N series storage system. This can be done using a switch or a cross-over cable on most networking topologies (we used Gigabit Ethernet and a cross-over cable).
- To increase the occurrence of parallel I/O between the database server and the IBM N series storage system, create separate volumes on the IBM N series storage system for database data and transaction logs and use separate mount points to mount them to the database server.

- Dedicated or private network connections between the DB2 server and the IBM N series storage system are recommended for the following reasons:
  - Any issues of contention or latency are eliminated if the DB2 server and the IBM N series storage system are the only two nodes in the network.
  - Creating a private network connection ensures security. There is no need to protect DB2-specific files from tampering, as would be the case if a shared network were used.
  - As additional IBM N series storage systems are added to the network to increase storage capacity, they will adhere to the same security levels as the database server and the existing storage system do.
- Each dedicated network connection, using a high-end networking protocol, such as Gigabit Ethernet, provides a large amount of bandwidth between the IBM N series storage system and the database server. These connections are identical to SCSI or FC-AL I/O connections, except they use the NFS service instead of local I/O services.

3

Configuration overview

In the past, installation of DB2 UDB on a UNIX platform called for the storage of the databases' data and corresponding transaction log files on local disk(s). When an IBM N series storage system is used, a Database Administrator (DBA) can take full advantage of IBM N series technologies, such as Snapshot™, SnapMirror®, RAID-DP, and FlexClone (Figure 3-1 on page 8) by placing the database's overhead files, data, and transaction log files on an IBM N series storage system.

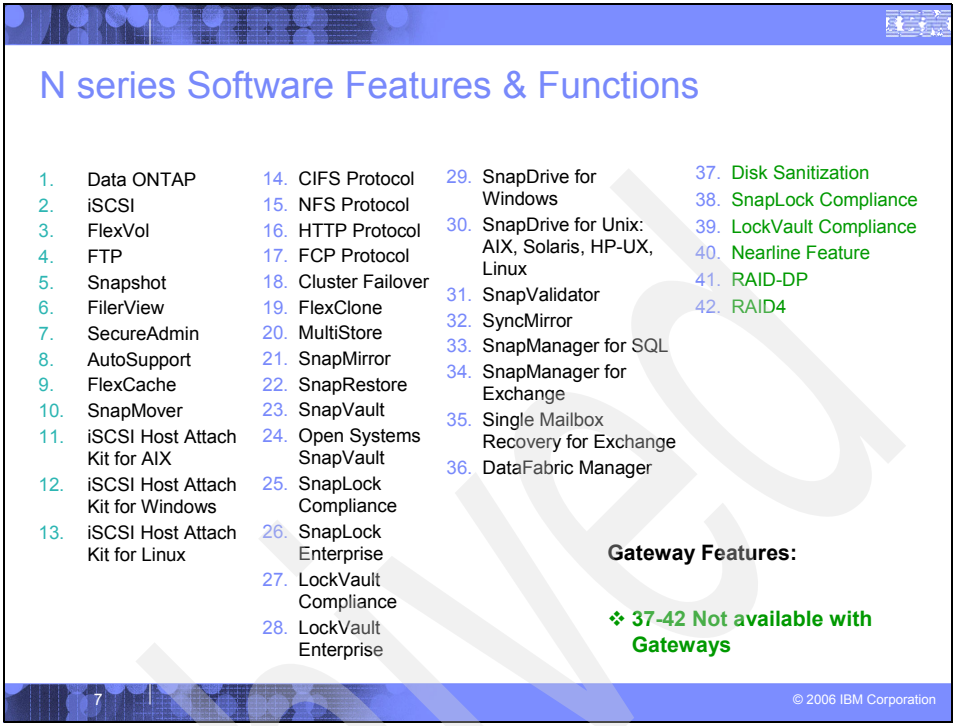


Figure 3-1 N series features

3.1 Designing a database with recovery in mind

When designing a new database, it is important to think about recovery needs from the start. As part of the database design process, you should identify any relationships that exist between the various database objects. These relationships can be at an application level, where transactions can work with one or more tables, as well as at a database level, where referential integrity constraints can exist between tables, or where events against one table can activate triggers that affect other tables. With these relationships in mind, you can group related database objects together on the same logical IBM N series storage system volume or group of volumes. Placing database objects with similar backup requirements or functions on the same logical volume(s) will simplify the use of Snapshot copies and typically make recovery that much easier (Figure 3-2). In addition, it goes without saying that the archive logs should be stored on a separate volume from that on which the data is stored. A documented model of your database design can be a tremendous benefit when making these kinds of decisions.

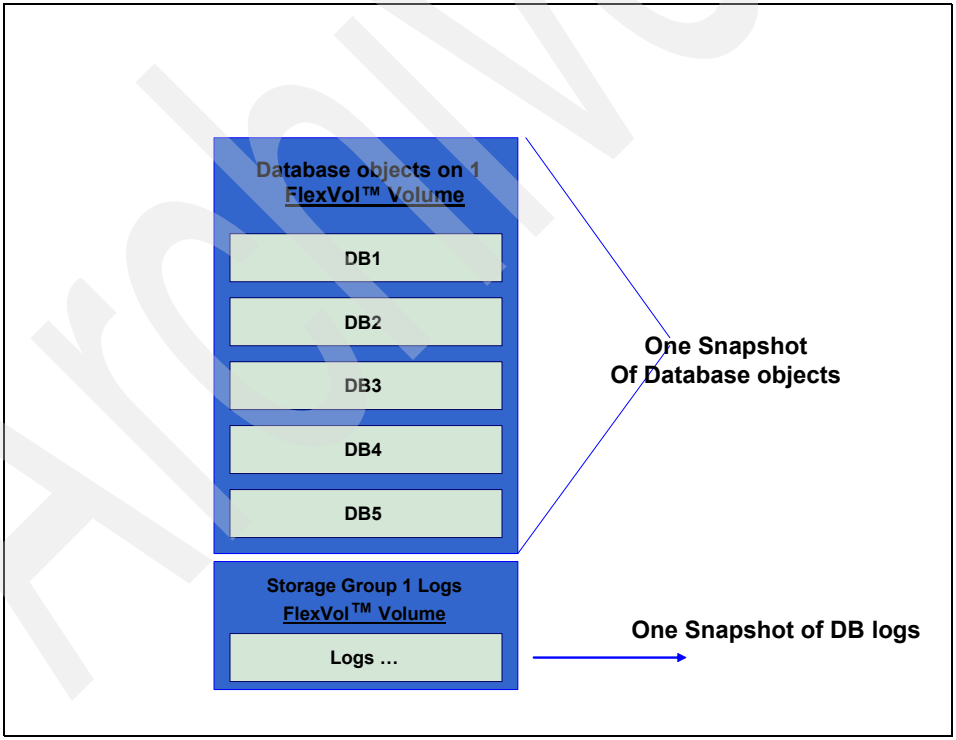


Figure 3-2 Database layout

3.2 Aggregate and volume configuration

The basic data container in DB2 UDB is the tablespace; a tablespace defines the physical relationship of the database objects and the storage on which they reside. Tablespaces also allow you to assign the location of database objects and data directly to one or more containers. A container can be a directory, a file, or a raw device. If a single tablespace spans more than one container, the DB2 Database Manager will attempt to balance the data load across all the containers used.

Data ONTAP 7G and above supports a new virtual storage layer called *flexible volumes*, hereafter known as FlexVol volume or volume. A FlexVol volume is created within an aggregate and is loosely coupled with its containing aggregate. The FlexVol volume provides greater performance than traditional volumes and can grow and shrink as needed by executing one simple command. Data ONTAP makes it easy to control the placement of related database objects on a FlexVol volume. For more information about creating and managing FlexVol volumes, refer to the **vol** command in the appropriate IBM N series documentation available online at the IBM support Web site.

Key to the ease of use and manageability of a DB2 UDB ESE database stored on an IBM N series storage system is the notion that the entire database can be stored on one or more FlexVol volumes. This configuration requires minimal attention by DBAs and System Administrators; allowing the IBM N series storage system to manage the physical data storage ensures high performance and availability. However, there are a few IBM N series storage system physical design considerations that need to be taken into consideration to ensure that these benefits are not compromised:

- The root volume should be its own volume. The root volume can be a traditional volume or a FlexVol volume.
- We strongly recommend that all the database object files be stored on a volume on the IBM N series storage system that is not the root volume.
- We strongly recommend that the database data and transaction logs (archive as well as active) be kept on separate FlexVol volumes. This allows the database to be backed up on a different schedule from the logs. This helps decrease the loss of data after a database failure, due to the availability of more current logs. If the transaction logs and the database reside on the same FlexVol volume, the recovery of the volume would cause overwriting of the active and archive logs, resulting in a loss of all the log data that was generated after the Snapshot copy was created. As a result, no transaction log data would exist for a roll-forward recovery operation.
- We strongly recommend that a large aggregate be used whenever possible for performance reasons. Multiple FlexVol volumes required for a database

can be created within a single large aggregate. In Figure 3-3, we show the aggregate dbagr.

Note: Throughout this IBM Redbook, the terms volume and FlexVol volume are interchangeable unless otherwise noted.

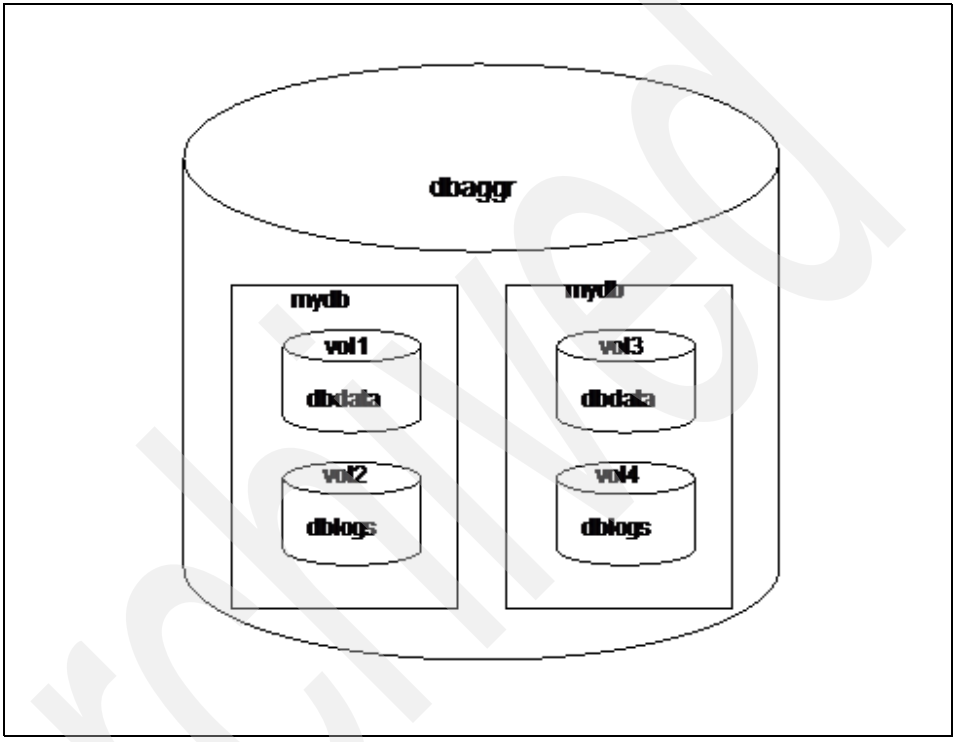


Figure 3-3 One large aggregate

4

IBM DB2 UDB databases on IBM N series storage system

This chapter will discuss the advantages of storing DB2 UDB databases on an IBM N series storage system.

Running a DB2 database with data and transaction log files stored on an IBM N series storage system has several advantages, which will be discussed in the sections that follow.

4.1 Extremely fast backup

Backup performance can be significantly improved using Snapshot copies in conjunction with DB2's **`set write suspend`** and **`set write resume`** commands. Snapshot copies can be created in a matter of seconds, regardless of the size of the database or the level of activity on the IBM N series storage system used. This reduces the database backup window from hours to seconds and allows DBAs to take frequent and full database backups without having to take the database offline. Figure 4-1 shows an overview of the Snapshot process.

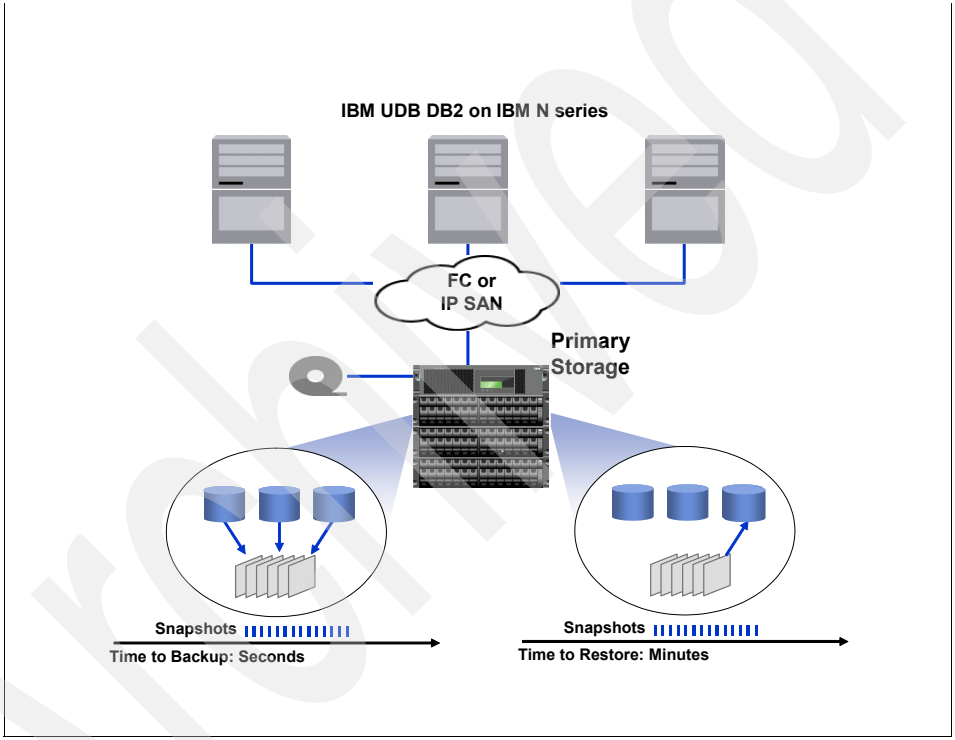


Figure 4-1 Snapshots

4.2 Quick recovery

Using the Data ONTAP SnapRestore® command, an entire database can be restored in a matter of seconds. Since there is no data copying involved, an incredible amount of time is saved, as the file system is put back to the original state it was in at the time the Snapshot was created. Data ONTAP 7G supports 255 Snapshot copies per WAFL® volume. The ability to store a large number of

low-impact, frequently created Snapshot copies brings the time needed to perform a roll-forward recovery operation down to minutes or seconds and in many circumstances it allows the DBA to restore the database immediately without the necessity to restore from tape.

4.3 High availability

The need for 24x7 availability is fast becoming a reality for organizations of all sizes. Organizations cannot tolerate scheduled downtime or afford extended periods of slow system response caused by traditional database backup methods. Database backups using N series Snapshot technology can be created in matter of seconds without bringing a database down.

4.4 High reliability

The RAID architecture used for IBM N series storage systems is unique and provides greater reliability than many traditional RAID implementations. If a disk in an IBM N series RAID group fails, it is reconstructed automatically without any user intervention. Additionally, IBM N series supports RAID-DP architecture. RAID-DP is considered approximately 4000 times more reliable than traditional RAID.

4.5 No impact on system response time

Because a Snapshot copy is simply a picture of the file system at a specific point-in-time, and creating a database backup using Snapshot does not involve actual data I/O, the process of backing up a database has virtually have no performance impact on system response time, as shown in Figure 4-2.

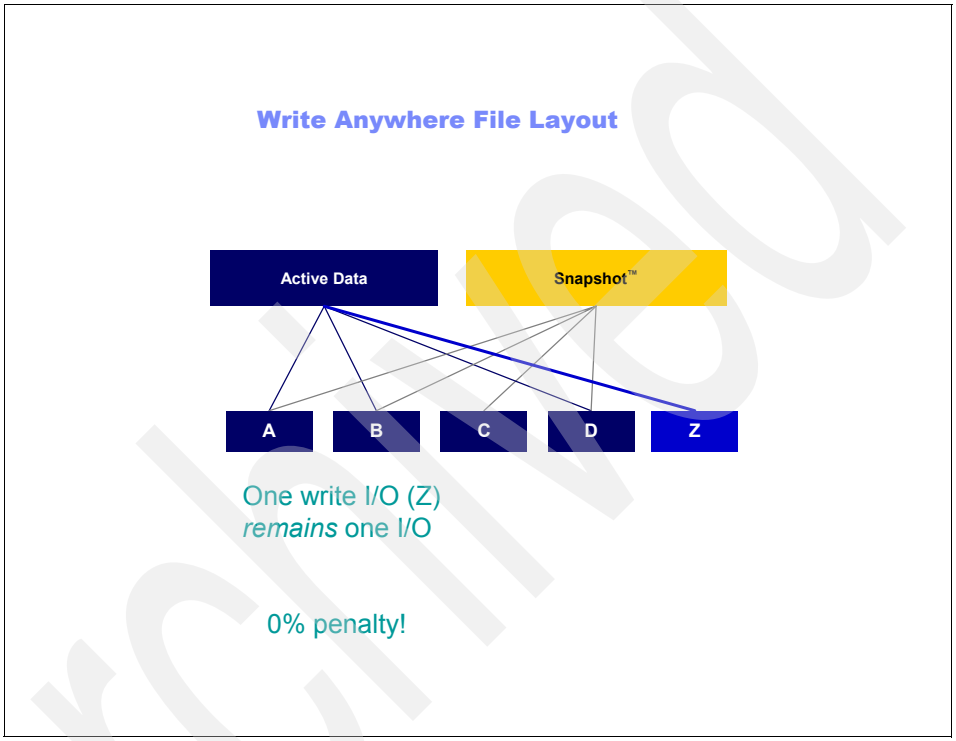


Figure 4-2 Snapshot I/O

4.6 Minimum storage requirement

Two Snapshot copies created in sequence differ from each other by the blocks added or changed in the time interval between the two. This block-incremental behavior limits associated storage capacity consumption.

4.7 Tablespace container load balance

Many of the tasks associated with balancing the load between multiple tablespace containers can be eliminated. Because of the high performance of the IBM N series storage system Data ONTAP operating system, only one container needs to be defined for each tablespace used.

5

Configuring the environment

This chapter will discuss configuring the database server and IBM N series storage system and describe the steps involved in installing the DB2 software on a database server and creating a DB2 database on an IBM N series storage system.

Note: The items described in Chapter 6, “Creating a new database on an IBM N series storage system” on page 69 must exist in order for this information to be valid.

Before you begin the installation process, you must ensure that the appropriate operating system kernel updates have been applied, and all the kernel parameters have been updated as per the installation documentation. Additional information about the DB2 installation procedure can be found in *Quick Beginnings for DB2 Servers*, found at:

ftp://ftp.software.ibm.com/ps/products/db2/info/vr82/pdf/en_US/db2ise81.pdf

Before you create a database on an IBM N series storage system, you need to complete a series of steps to configure your environment. These steps can be broken down into two sets: one set of steps must be performed on the IBM N series storage system, while the other needs to be performed on the database server.

5.1 Configuring the IBM N series storage system

1. Activate the appropriate license keys

   On the IBM N series storage system, the file access protocols FCP, iSCSI, NFS, and CIFS are licensed services. You need to enable the appropriate service by activating license keys for the protocol you intend to use. The license keys can be activated by executing the following command from the IBM N series storage system:

   ```
   license add <license code>
   ```

   For example, to activate the license code for NFS, you would execute the following command on the IBM N series storage system:

   ```
   license add 123XYZABCD
   ```

   You can verify your action by issuing the **license** command, as shown in Figure 5-1 on page 21.

```
itsotuc2> license
               cifs site TNXZHUO
            cluster not licensed
     cluster_remote not licensed
  disk_sanitization not licensed
                fcp site RGZOV1L
         flex_cache not licensed
         flex_clone site IPMK2CN
            gateway not licensed
    gateway_hitachi not licensed
               http not licensed
              iscsi site RGF3RGF
         multistore site DF4XLWH
                nfs site R5IEGMK
      rapid_restore not licensed
           smdomino not licensed
              smsql site 6BAEGMK
           snaplock not licensed
snaplock_enterprise site FITZKF7
snapmanagerexchange site NGQKA8N
          snapmirror site TTOX9WH
    snapmirror_sync site NYJABTI
           snapmover not licensed
        snaprestore site XJBGCKE
      snapvalidator site ZCUZHUB
       sv_linux_pri not licensed
       sv_ontap_pri not licensed
       sv_ontap_sec site DSQMWZG
        sv_unix_pri not licensed
 sv_windows_ofm_pri not licensed
     sv_windows_pri not licensed
   syncmirror_local not licensed
                vld site DTWKACE
```

Figure 5-1 license command

2. Update the /etc/host file on the IBM N series storage system used.

   The IBM N series storage system must be able to communicate with the database server and vice versa. The IBM N series storage system can communicate to the database server if there is an entry in its /etc/hosts file for the database server or, alternatively, if it uses some other host name resolution techniques, such as NIS or DNS. By default, the /etc/hosts file is checked first for host name resolution. The easiest way to update the

/etc/hosts file on the storage system is by using FilerView® (see Figure 5-2). Entries made in the /etc/hosts file should look similar to the following one:

```
<db server IP address>  <host name>
```

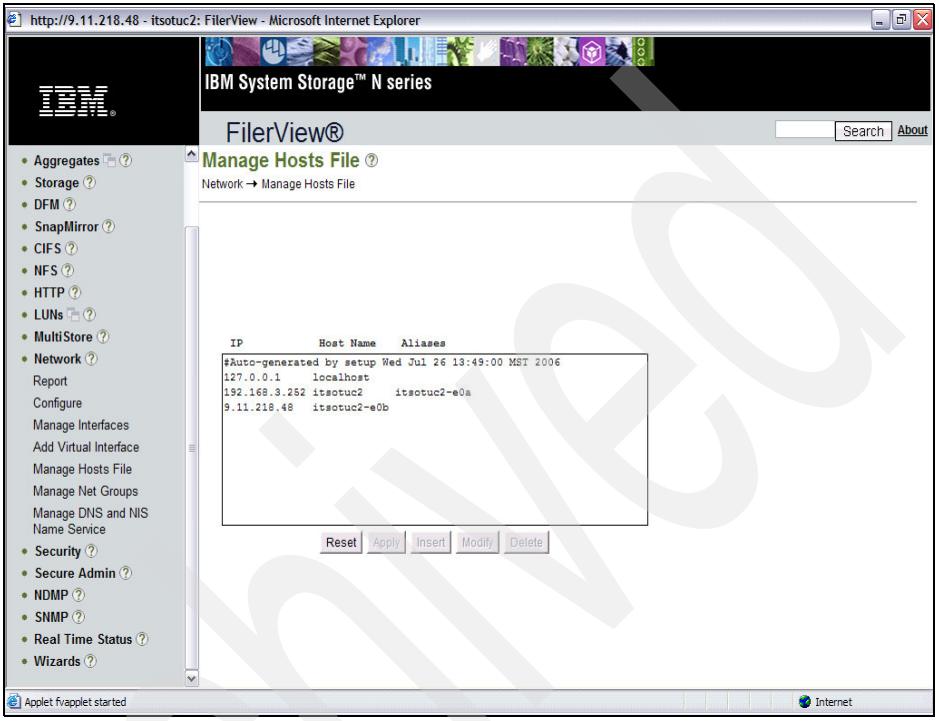


Figure 5-2 Filerview

For example, to add an entry for a database server named db2srv1 that has an IP address of 172.17.32.112, you would add the following line to the /etc/hosts file on the IBM N series storage system:

```
172.17.32.112       db2srvr1
```

3. Enable **rsh** access for the database server.

   In order to use the **rsh** (remote shell) command from the database server, you need to perform two steps. First, enable the rsh option on the IBM N series storage system by executing the following command:

   ```
   options rsh.enable on
   ```

   Then, add the database host and user name entry to the /etc/hosts.equiv file found on the IBM N series storage system. The entry to this file looks somewhat similar to the following one:

   ```
   < hostname >  < username >
   ```

For example, to allow **rsh** command execution from a database server named db2srv1 by a user named db2inst1, you would add the following line to the /etc/hosts.equiv file on the IBM N series storage system:

```
db2srvr1 db2isnt1
```

4. Create space to store data and transaction log files.

   In order to create a database on the IBM N series storage system, you need to create aggregates, flexible volumes and, in some cases, LUNs. An aggregate is a physical pool of storage at a RAID level (Figure 5-3) and is created using the following command:

   ```
   aggr create <aggrname> -r <raidsize> <ndisks@disksize> -t <raidtype>
   ```

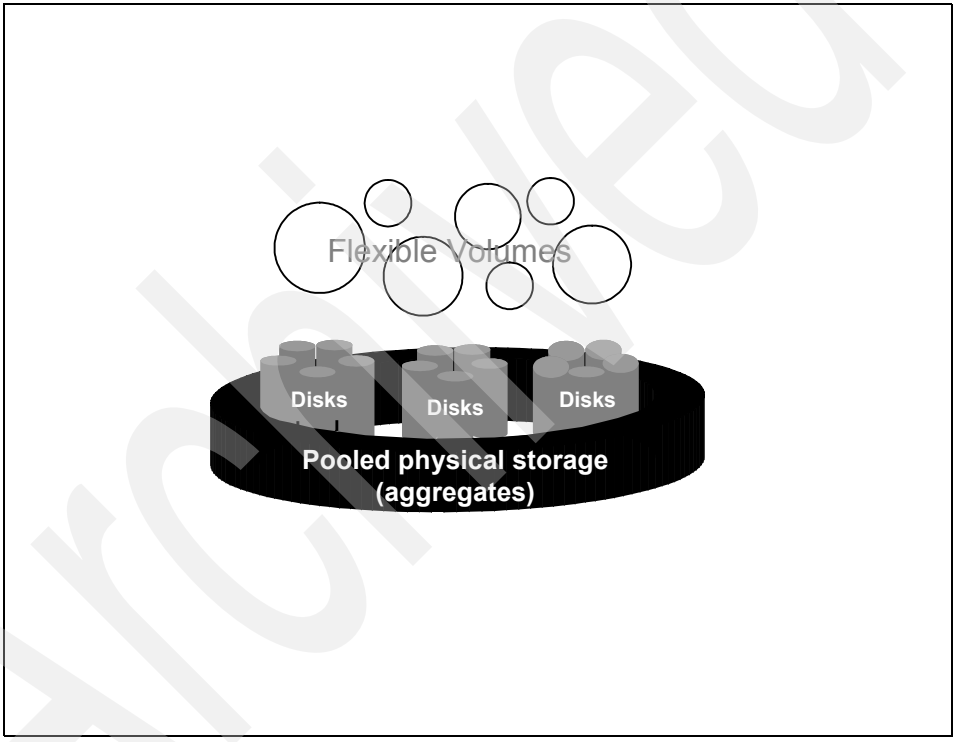


Figure 5-3 Aggregates

For example, to create an aggregate named db2aggr1 that has eight 72 GB disks in it and that uses RAID-DP, you would execute the following command on the IBM N series storage system:

```
aggr create db2aggr1 -r 8 8@72g -t raid_dp
```

A FlexVol volume is logical storage container inside an aggregate. A FlexVol volume can be as small as a few megabytes and as large as the aggregate itself. A FlexVol volume can be created using the following command:

```
vol create <volname> <aggrgate> size[k|m|g|t]
```

For example, to create a flexible volume named db2data that is 100 GB in size and that resides in the aggregate named db2agg1, you would execute the following command on the IBM N series storage system:

```
vol create db2data db2agg1  100GB
```

5. Specify the volume security settings to use.

   You must ensure that the volume(s) on the IBM N series storage system that are to be used for the DB2 UDB database support either the UNIX or the mixed security style. The security setting of a volume can be set using the following command:

   ```
   qtree security <vol name> [ unix | mixed ]
   ```

   For example, to change the security style to UNIX on an IBM N series storage system volume named db2data, you would execute the following command on the IBM N series storage system:

   ```
   qtree security /vol/db2data unix
   ```

   Repeat this step and change the security style for all the volumes to be used for the database.

6. Disable the automatic Snapshot feature.

   Normally, a database is backed up based on a user defined schedule. Therefore, we recommend that you turn off the automatic Snapshot feature for any volume to be used for the database and its transaction log files. The automatic Snapshot feature can be turned off using the following command:

   ```
   vol options <vol name> nosnap on
   ```

   For example, to turn the automatic Snapshot feature off for a volume named db2data, you would execute the following command on the IBM N series storage system:

   ```
   vol options db2data nosnap on
   ```

   Repeat this step and turn the auto Snapshot off for all the volumes to be used for the database.

7. Export the volumes if NFS is used (Skip this step if FCP or iSCSI is used).

   In order to access the volumes on an IBM N series storage system using the NFS protocol, you must export them. A volume can be exported by adding an entry to the /etc/exports file found on the IBM N series storage system and giving the database server appropriate access to it. You can use FilerView

(see Figure 5-4) or the **exportfs** command for this purpose. NFS exports are managed using the following command:

```
exportfs -p rw=<hostname>,root=<hostname> <pathname>
```

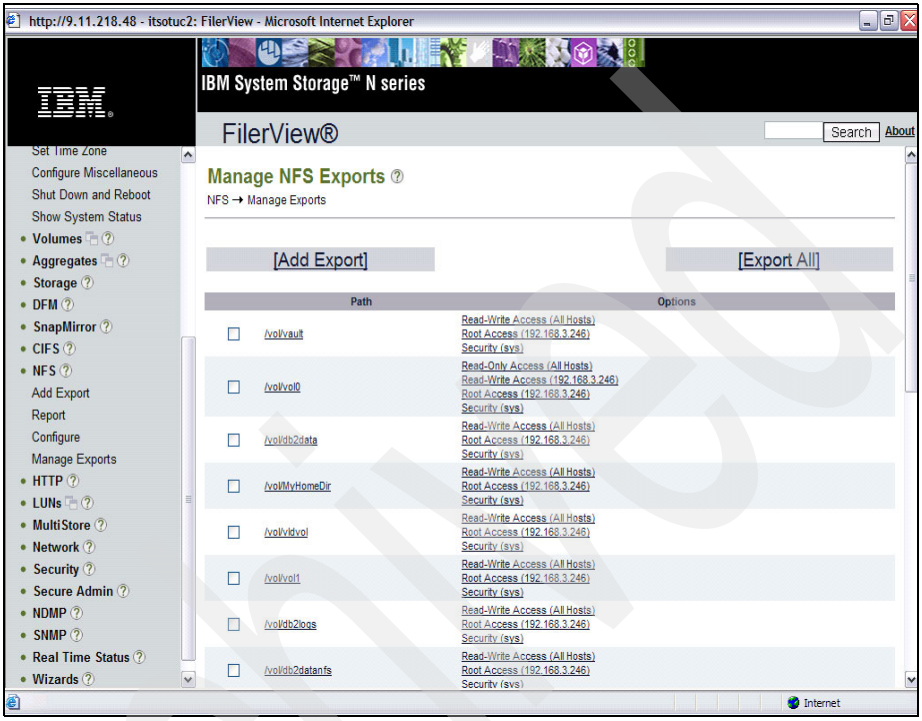


Figure 5-4 FilerView

For example, to create an export entry for a volume named db2data and allow root access to it from a database server named db2srv1, you would execute the following command on the IBM N series storage system:

```
exportfs -p rw=db2srv1,root=db2srv1 /vol/db2data
```

Repeat this step and create an export entry for all the volumes to be used for the database.

8. Set up Initiators and LUNs if FCP or iSCSI is used (Skip this step if NFS is used).

   In a SAN or IP SAN environment, UNIX hosts serve as initiators and IBM N series storage systems serve as targets. The logical storage layer created on an IBM N series storage system is referred as a LUN. Fibre Channel SAN and IP SAN (iSCSI) are block based access protocols that access LUNs on a storage system using a hardware or software initiator installed on the

database server. The initiator has a unique network identification know as World Wide Port Number (WWPN) or World Wide Node Name (WWNN).

a. Create an Initiator Group (igroup) for the database server.

   Before attempting to create an igroup, you need to find the WWPN or WWNN for the initiator. In order to determine the WWPN/WWNN, refer to 5.2, “Configuring a database server in a SAN environment” on page 33. Use the WWPN or WWNN to create an igroup on the IBM N series storage system by executing the following command:

   ```
   igroup create -[f|i] -t [linux|aix|solaris|hpux] <igroupname>
   <wwpn|wwnn>
   ```

   For example, to create an igroup named db2srv1_igp for an initiator on a Linux host that has a WWPN value of 10000000c93a069b, you would execute the following command on the IBM N series storage system:

   ```
   igroup create -f -t linux db2srv1_igp 10000000c93a069b
   ```

b. Create the appropriate LUNs and LUN Mappings.

   LUNs are used by initiators to identify SCSI devices on the target IBM N series N storage system. A LUN is created within a volume or qtree by executing the following command:

   ```
   lun create -s size[k|m|g|t] -t [linux|aix|solaris|hpux] lun_path
   ```

   For example, to create a 75 GB LUN within a FlexVol volume named db2data, you would execute the following command on the IBM N series storage system:

   ```
   lun create -s 75G -t linux /vol/db2data/data1
   ```

   In order to make LUNs accessible to a host server, you must map them to an igroup, which is associated to an initiator installed on the host. A LUN mapping can be created by executing the following command:

   ```
   lun map <lun_path>  <igroup> <lun id>
   ```

   For example, to create a LUN mapping for a LUN named /vol/db2data/data1 to an igroup named db2srv1_fcp_igp, you would execute the following command on the IBM N series storage system:

   ```
   lun map /vol/db2data/data1  db2srv1_fcp_igp 0
   ```

c. Obtain a node name for the IBM N series storage system used.

   The target IBM N series storage systems are identified by a unique node name that is used for creating persistent binding for the storage devices on the host. You can obtain the node name assigned to a IBM N series storage system by executing the following command:

   ```
   < fcp | iscsi > nodename
   ```

For example, to find the node name assigned to an iSCSI target, you would execute the following command on the IBM N series storage system:

```
iscsi nodename
```

The output will look similar to the one in Figure 5-5.

```
itsotuc2> iscsi nodename
iSCSI target nodename: iqn.1992-08.com.netapp:sn.84187691
```

Figure 5-5 iscsi command

5.1.1 Configuring database server in an NAS environment

After creating volumes on the IBM N series storage system and updating the /etc/exports/ file, the database server must be configured to make use of these volumes. The steps carried out to configure the database server to use the IBM N series storage system volumes over NFS depends on the UNIX flavor. The following are the basic steps needed to configure the database server for most popular UNIX flavors:

1. Log in to the database server as the root user.
2. Add the IP address and name of the IBM N series storage system to the /etc/hosts file found on the database server. The entry in the /etc/hosts file should look somewhat similar to the following:

   ```
   <IP Address>  <IBM N series storage system name>
   ```

   For example, to add a IBM N series storage system that has the IP Address 10.60.175.57 and the name nseries to the /etc/hosts file found on a database server, you would enter the following line:

   ```
   10.60.175.57 nseries
   ```

3. In order to install and work with the DB2 UDB database, you need to create database user groups and user accounts. User and group authentication is managed in a facility external to DB2 UDB, such as the operating system, a domain controller, or a Kerberos security system.

   We used the operating system's authentication mechanism and we created a group named db2adm and a user account named db2inst1. This was done by executing the following commands on the database server:

   ```
   groupadd -g 700 db2adm       # group of users to be granted SYSDBA
   authority
   useradd -c "DB2 Inst Owner" -u 700 -g db2adm -G db2adm db2inst1
   passwd db2inst1
   ```

You can create additional database groups and user accounts using the **groupadd** and **useradd** commands.

4. Create mount points to mount IBM N series storage system volumes.

   Create mount points by executing the following command on the database server:

   ```
   mkdir -p <mount point>
   ```

   For example, to create a mount point named /mnt/db2data, you would execute the following command on the database server:

   ```
   mkdir -p /mnt/db2data
   ```

   Repeat this step for each IBM N series storage system volume to be used for the DB2 database.

5.1.2 IBM AIX

1. The IBM N series storage system volumes to be used for the database need to be mounted on the database server. You need to modify the /etc/filesystems file found on the database server and add an entry for each storage system volume. Open the /etc/filesystems file with a text editor and add entries for each IBM N series storage system volume. The entry should look similar to the one in Figure 5-6.

```
 <mount point>:
dev      = <IBM N series storage system volume>
mount    = true
vfs      = nfs
nodename = nseries
options  = bg,nointr,rw
type     = nfs_mount
account  = false
```

Figure 5-6 /etc/filesystems file

For example, to add an entry for a IBM N series storage system volume named db2data to be mounted on a mount point named /mnt/db2data, you should append the lines shown in Figure 5-7 on page 29 to the /etc/filesystem file on the database server.

```
 /mnt/db2data:
dev      = /vol/db2data
mount    = true
vfs      = nfs
nodename = nseries
options  = bg,nointr,rw
type     = nfs_mount
account  = false
```

Figure 5-7 /etc/filesystem/ modified

In the above entry, `mount = true` makes a particular volume persistent across the system reboots.

After adding the entries to the /etc/filesystems file, you need to mount the IBM N series storage system volumes by executing the following command on the database server:

```
mount <mount point>
```

For example, to mount a IBM N series storage system volume to a mount point named /mnt/db2data, you would execute the following command on the database server:

```
mount /mnt/db2data
```

Repeat this step for each IBM N series storage system volume to be used for the DB2 database.

2. In order to create a database on the IBM N series storage system volumes, the database instance owner must have ownership of the mounted file systems. You need to change ownership for each mounted IBM N series storage system volume by executing the following command on the database server:

   ```
   chown -R db2inst1:db2adm <mount point>
   ```

   For example, to change the ownership of a IBM N series storage system volume mounted on the mount point named /mnt/db2data to the DB2 instance owner named db2inst1, you would execute the following command on the database server:

   ```
   chown -R db2inst1:db2adm /mnt/db2data
   ```

   Repeat this step for each mount point to be used for the DB2 database.

Note: You can also use SMITTY or SMIT to mount the IBM N series storage system volumes.

5.1.3 Linux host

1. Append an entry for each IBM N series storage system volume to the /etc/fstab file found on the database server. The entry should look somewhat similar to the following one:

   ```
   <IBM N series storage system name>:<volume name> <mount point> nfs
   hard, rw,nointr,rsize=32768,wsize=32768,bg,vers=3,tcp 0 0
   ```

   For example, in order to mount a IBM N series storage system volume named db2data on a mount point named /mnt/db2data, you would add the following entry to the /etc/fstab file found on the database server:

   ```
   nseries:/vol/db2data /mnt/db2data nfs hard,rw,nointr, rsize=32768,
   wsize=32768,bg,vers=3,tcp 0 0
   ```

2. After adding the entries to the /etc/fstab file, you need to mount and make the IBM N series storage system volumes available by executing the following command on the database server:

   ```
   mount <mount point>
   ```

   For example, to mount a IBM N series storage volume named db2data on a mount point named /mnt/db2data and make it available, you would execute the following command on the database server:

   ```
   mount /mnt/db2data
   ```

   Repeat this step for each IBM N series storage system volume to be used for the DB2 database.

3. In order to create a database on the IBM N series storage system volumes, the database instance owner must have ownership of the mounted file systems. You need to change ownership for each mounted IBM N series storage system volume that is used for the database by executing the following command on the database server:

   ```
   chown -R db2inst1:db2adm <mount point>
   ```

   For example, to change the ownership of the IBM N series storage system volume mounted on the mount point named /mnt/db2data to the DB2 instance owner named db2inst1, you would execute the following command on the database server:

   ```
   chown -R db2inst1:db2adm /mnt/db2data
   ```

5.1.4 Sun Solaris

1. Append an entry for each IBM N series storage system volume to be used for the database to the /etc/vfstab file on the database server. The entry should look somewhat similar to the following:

   ```
   <IBM N series storage system name>:<volume name> <mount point> nfs -
   yes rw,bg,hard,intr,rsize=32768,wsize=32768,[forcedirectio| llock],
   tcp,vers=3
   ```

   For example, in order to add an entry for a IBM N series storage system volume named db2data to be mounted on a mount point named /mnt/db2data, you would append the following line to the /etc/vfstab file on the database server:

   ```
   nseries:/vol/db2data  /mnt/db2data nfs - yes rw,bg,hard,intr,
   rsize=32768,wsize=32768,forcedirectio,tcp,vers=3
   ```

2. After appending the entries to the /etc/fstab file, you need to mount and make the IBM N series storage system volumes available by executing the following command on the database server:

   ```
   mount <mount point>
   ```

   For example, to mount an IBM N series storage system volume named db2data on a mount point named /mnt/db2data and make it available, you would execute the following command on the database server:

   ```
   mount /mnt/db2data
   ```

   Repeat this step for each IBM N series storage system volume to be used for the DB2 database.

3. In order to create a database on the IBM N series storage system volumes, the database instance owner should have ownership of the mounted volumes. You need to change ownership for each mounted file system to be used for the database, by executing the following command on the database server:

   ```
   chown -R db2inst1:db2adm <mount point>
   ```

   For example, to change the ownership of an IBM N series storage system volume mounted on a mount point named /mnt/db2data to the DB2 instance owner named db2inst1, you would execute the following command on the database server:

   ```
   chown -R db2inst1:db2adm /mnt/db2data
   ```

5.1.5 HP-UX

1. Append an entry for each IBM N series storage system volume to be used for the database to the /etc/fstab file on the database server. The entry should look somewhat similar to the following one:

   ```
   <IBM N series storage system name>:<volume name>   <mount point> nfs
   bg 0 0
   ```

   For example, in order to add an entry for a IBM N series storage system volume named db2data to be mounted on a mount point named /mnt/db2data, you would append the following line to the /etc/fstab file on the database server:

   ```
   nseries:/vol/db2data /mnt/db2data nfs bg 0 0
   ```

2. After appending the entries to the /etc/fstab file, you need to mount and make the IBM N series storage system volumes available by executing the following command on the database server:

   ```
   mount <mount point>
   ```

   For example, to mount a IBM N series storage system volume named db2data on a mount point named /mnt/db2data and make it available, you would execute the following command on the database server:

   ```
   mount /mnt/db2data
   ```

   Repeat this step for each IBM N series storage system volume to be used for the DB2 database.

3. In order to create a database on the IBM N series storage system volumes, the database instance owner must have ownership of the mounted volumes. You need to change ownership for each mounted IBM N series storage system volume to be used for the database by executing the following command on the database server:

   ```
   chown -R db2inst1:db2adm <mount point>
   ```

   For example, to change the ownership of the IBM N series storage system volume mounted on the mount point named /mnt/db2data to the DB2 instance owner named db2inst1, you would execute the following command on the database server:

   ```
   chown -R db2inst1:db2adm /mnt/db2data
   ```

5.2 Configuring a database server in a SAN environment

After creating the initiator group (igroup) and LUN mappings, the database server must be configured to utilize the space available on the IBM N series storage system LUNs. The steps carried out to configure the database server depend on the flavor of UNIX being used. The following section describes the basic steps used to configure the most common UNIX hosts.

5.2.1 Linux

Here we describe the configuration of a database server for Linux.

Set up kernel parameters

The default kernel parameters value may not be good enough to run DB2 UDB successfully on your Linux server. You may be required to update some of the kernel parameters' default values. Some of the recommended Linux kernel parameter values for DB2 UDB are:

- Kernel.shmmax=268435456 (for 32-bit)
- kernel.shmmax=1073741824 (for 64-bit)
- kernel.msgmni=1024
- fs.file-max=8192
- kernel.sem="250 32000 32 1024

DB2 UDB Version 8.2 has a new feature that checks the values of the kernel.semmni, kernel.msgmni, and kernel.shmmax parameters when you enter the **db2start** command, and changes them to the recommended values if the current values are not optimal. However, some of the other parameters may need to be changed manually.

To check your current shared memory segment, semaphore array, and message queue limits, enters the following command on your database server:

```
ipcs -l
```

You should see something similar to Figure 5-8.

```
[root@itso itso]# ipcs -l

------ Shared Memory Limits --------
max number of segments = 4096
max seg size (kbytes) = 2097152
max total shared memory (kbytes) = 8388608
min seg size (bytes) = 1

------ Semaphore Limits --------
max number of arrays = 128
max semaphores per array = 250
max semaphores system wide = 32000
max ops per semop call = 100
semaphore max value = 32767

------ Messages: Limits --------
max queues system wide = 16
max size of message (bytes) = 8192
default max size of queue (bytes) = 16384
```

Figure 5-8 ipcs output

In order to make the kernel parameter changes permanent after a system reboot, complete the following steps:

1. Open up /etc/sysctl.conf in a text editor and add the following entries:

   ```
   kernel.shmmax=268435456
   kernel.msgmni=1024
   kernel.sem="250 32000 32 1024"
   ```

2. Enter the **sysctl -p** command to load the sysctl settings from /etc/sysctl.conf.
3. Enter the **ipcs -l** command to view the updated kernel parameters in sysctl.

- The entries from the sysctl.conf file are read by the network initialization script during startup. On some distributions, you may be required to add **sysctl -p** in one of the system initialization files (for example, rc.local) so that the kernel parameters are set after each reboot. For a detailed list of recommended kernel parameters please, refer to Chapter 9, “Preinstallation tasks”, in *Quick Beginnings for DB2 Servers*, found at:

  ftp://ftp.software.ibm.com/ps/products/db2/info/vr82/pdf/en_US/db2ise81.pdf

Configuring a Linux Host for FCP

1. Install the IBM host Attach kit.

   We recommend that you download and install the SAN (FCP) host attach kit for Linux provided by IBM. This product simplifies the configuration of a Linux machine as the host component of an IBM N series SAN environment. In order to install this product, you need to complete the following basic steps:

   1. IBM System Storage™ N series FCP Linux Host Attach Kit 3.0 binaries are provided to licensed customers only and can be obtained by contacting IBM support.

   b. After obtaining and copying the downloaded attach kit tar file to a directory (/tmp/nseries) on the database server and extract the software by executing the following command:

      ```
      tar -xvzf /tmp/nseries netapp_linux_tools_3_0.tar.gz
      ```

      For example, in order to extract the tar file named ntap_linux_tools_1_0.tar.gz, you would execute the following command on the database server:

      ```
      tar -xvzf /tmp/nseries/netapp_linux_tools_3_0.tar.gz
      ```

      The extracted software is placed in the /tmp/nseries/netapp_linux_tools_3_0 directory.

   c. Change the directory to netapp_linux_tools_3_0, where you have extracted the attach kit software, and execute the following command on the database server:

      ```
      ./install
      ```

   d. Add the following lines to your /.bash_profile file:

      ```
      export PATH=$PATH:/opt/NetApp/santools/bin
      ```

   For detailed installation steps please refer to *Installation and Setup Guide for Fibre Channel Protocol on Linux* on the IBM support Web site at:

   http://www-03.ibm.com/servers/storage/support/nas/index.html

2. Install the HBA and driver.

   Before you install the HBA on the database server, you need to check for product compatibility on the IBM Web site. The compatibility matrix is available on the IBM support site. After confirming compatibility, install the HBA and its driver on the database server. For the HBA and its driver installation steps, please refer to product's installation and configuration guide on IBM support site, found at:

   http://www-03.ibm.com/servers/storage/support/nas/index.html

3. Obtain the WWPN for the initiator.

Each initiator installed on the database server is uniquely identified by WWPN or WWNN. The WWPN is required to create an igroup for FC on the IBM N series storage system. The WWPN for the HBA installed on the database server can be obtained by completing the following steps:

a. After installing the IBM N series attach kit, the HBA, and the required driver, you would execute the following command on the database server to obtain the WWPN:

```
sanlun fcp show adapter -c
```

Upon execution with -c option, the **sanlun** command generates the **igroup create** command. The output of the above command looks somewhat similar to the one below:

```
igroup create -f -t linux db2srv1 10000000c93a069b
```

b. The WWPN for adapter in the above example is 10000000c93a069b. Execute the above generated command to create the igroup on the IBM N series storage system.

c. Refresh the FC adapter by executing the following commands on your database server:

```
modprobe -r qla2300
modprobe -v qla2300
```

Configuring a Linux Host for iSCSI

1. Installing the IBM N series iSCSI Support Kit.

We recommend that you download and install the IBM N series provided iSCSI Host Support Kit for Linux. This product simplifies the configuration of a Linux host in an IBM N series IP SAN environment. In order to install this product, you need to complete the following basic steps:

a. Go to http://www-03.ibm.com/servers/storage/support/nas/index.html.

b. Under the Host Based Software section, select the **iSCSI Host Support Kit** product link of the page and select the **Download** tab.

c. Select the **Downloadable files** link. Select the **IBM System Storage N series iSCSI Initiator Support Kit Download** link. Select the **Linux Initiator Support Kit 1.4**, accept the terms, sign in, and then select **Download the iSCSI Red Hat Linux Initiator Support Kit Version 1.4**. Click on the **Continue** button at the bottom of the page. Accept the license and download the software tar file to a directory on your local machine.

d. Copy the downloaded support kit file to a directory (/tmp/nseries) on the database server and extract it using the following command:

```
tar -xvzf <file name>
```

For example, in order to extract the tar file named ibm_linux_tools_1_5.tar, you would execute the following command on the database server:

```
tar -xvf /tmp/nseries/ibm_linux_tools_1_5.tar
```

e. Change the directory to ibm_linux_tools_1_5, where you have extracted the software, and install the product by executing the following command on the database server:

```
./install
```

f. Add the following lines to your /.bash_profile file:

```
export PATH=$PATH:/opt/ontap/santools/bin
export MANPATH=/usr/share/man:/opt/ontap/santools/man
```

2. Install the iSCSI initiator.

IBM N series supports the iSCSI software initiator for Linux. In order to install the iSCSI initiator, complete the following steps:

a. If you are using RHEL 3.0 Update 4 or higher, then the iSCSI Software initiator is contained in Red Hat Linux distribution; otherwise, download it from http://sourceforge.net/projects/linux-iscsi/.

b. After the download completes, create a work directory named /tmp/nseries on the database server and ftp the downloaded initiator tar file to the work directory on the database server. Uncompress the tar file by executing the following command:

```
tar -xvzf <file name>
```

For example, to uncompress and extract the initiator tar file named linux-iscsi-4.0.2.gz, you would execute the following command on the database server:

```
tar xvzf /tmp/nseries/linux-iscsi-4.0.2.gz
```

c. Change the directory to linux-iscsi-4.0.2, where you have extracted the software and compile the iSCSI driver by executing the following command on the database server:

```
make
```

If your kernel sources are not in the usual place, add `TOPDIR=/path/to/kernel` or edit the definition of TOPDIR in the makefile.

If your kernel configuration file is not in the usual place, add `KERNEL_CONFIG=/path/to/.config` or edit the definition of KERNEL_CONFIG in the makefile.

d. Install the driver by executing the following command on the database server:

```
make install
```

e. In order to make the initiator discover the LUN devices on the IBM N series storage system, you need to add the following lines to the iSCSI configuration file named /etc/iscsi.conf that is found on the database server:

```
continuous=no
headerDigest=never
dataDigest=never
discoveryAddress=10.32.90.100
```

Where 10.32.90.100 is the IBM N series storage system IP address.

f. After updating the /etc/iscsi.conf file, you need to start the iSCSI service and load the driver by executing the following command on the database server:

```
/etc/init.d/iscsi start
```

Every time you make any LUN mapping changes, the iSCSI driver needs to be refreshed. You can do so by executing the following command on the database server:

```
/etc/init.d/iscsi restart
```

For additional information and troubleshooting please read the README file in the directory where you extracted the initiator software.

3. Obtain WWNN for the initiator.

The iSCSI initiator has a unique identification known as the World Wide Node Name or WWNN. To access LUNs on the IBM N series storage system, you must associate the initiator on the database server with an igroup on the IBM N series storage system. The WWNN for an initiator on the database server is required to create an igroup. The Linux software initiator stores its WWNN information in the /etc/initiatorname.iscsi file. You can obtain the WWNN for the initiator by executing the following command on the database server:

```
cat /etc/initiatorname.iscsi
```

Making LUNs accessible on the database server

IBM DB2 UDB supports two types of tablespace containers, file system containers and raw devices containers. The following sections describe the steps required to configuration both types of containers.

1. Accessing LUNs as a regular file system tablespace container

   Once the FCP or iSCSI driver is refreshed, the database server should be able to discover target LUNs on the IBM N series storage system as new devices. In order to use these devices as file system tablespace containers, complete the following steps:

   a. Get the assigned names for the newly added devices by executing the following command on the database server:

      ```
      sanlun lun show
      ```

   b. Before you use the newly added devices as the tablespace containers, you need to create partition tables and format them by executing the following command on the database server:

      ```
      fdisk < devicename >
      ```

      For example, to create partition tables and format a device named /dev/sdb, you would execute the following command on the database server:

      ```
      fdisk  /dev/sdb
      ```

      The **fdisk** command invokes the format wizard. You need to answer a series of questions in order to complete the formatting.

   c. After formatting, you need to create file system on each device by executing the following command on the database server:

      ```
      mkfs <devicename>
      ```

      For example, to create a file system on the device named /dev/sdb, you would execute the following command on the database server:

      ```
      mkfs /dev/sdb
      ```

   d. Add an ext3 journal to the file system, if it was not created as a journal file system, by executing the following command on the database server:

      ```
      tune2fs -j  < DeviceName >
      ```

      For example, to added the ext3 journal to a device named /dev/sdb, you would execute the following command on the database server:

      ```
      tune2fs -j /dev/sdb
      ```

e. After adding the file system, the LUNs can be mounted as a shared file system by executing the following commands on the database server:

```
mkdir -p <dir name>
mount <dir name> <mount point>
```

In order to mount a device named /dev/sdb on a mount point named /mnt/db2data, you would execute the following command on the database server:

```
mount /mnt/db2data /dev/sdb
```

f. In order to create a database on the IBM N series storage system volumes, the database instance owner must have the appropriate permissions on all the mounted volumes to be used for the database. You do so by changing the ownership of each mounted IBM N series storage system volume to the database instance owner by executing the following command on the database server:

```
chown -R db2inst1:db2adm <mount point>
```

For example, to change the ownership of the IBM N series storage system volume mounted on the mount point named /mnt/db2data, you would execute the following command on the database server:

```
chown -R db2inst1:db2adm /mnt/db2data
```

g. In order to make the file system mount persistent after a system reboot, you need to add an entry for each file system to the /etc/fstab.iscsi file, which is found on the database server. The entry in /etc/fstab.iscsi file should appear similar to the one indicated below:

```
<device name>  <mountpoint> <FS type> <mount option> <bkup freq>
<fsck pass>
```

For example, to make a mount point persistent for a device named /dev/sdb that is mounted on /mnt/db2data, you would add the following lines to the /etc/fstab.iscsi file found on the database server:

```
************************************************************
#device      mount        FS      mount      backup       fsck
#to mount    point        type    options    frequency    pass
/dev/sdb    /mnt/db2data  ext3    defaults      0         0
```

2. Accessing LUNs as raw devices

   If raw devices are desired as tablespace containers, then each LUN device used as a raw device needs to be converted to a character device. In order to use newly added LUN devices as raw device tablespace containers, complete the following steps on the database server:

   a. Convert newly added LUN devices to character devices by executing the following command on the database server:

      ```
      raw /dev/raw/raw[Number] [devicename]
      ```

      For example, to convert a block device named /dev/sdd to a character device, you would execute the following command on the database server:

      ```
      raw /dev/raw/raw1 /dev/sdd
      ```

   b. Add an entry for each raw device to the /etc/sysconfig/rawdevices file found on the database server. The entry should look somewhat similar to the following one:

      ```
      <rawdev> <blockdev name>
      ```

      For example, to add a raw device named /dev/raw/raw1, you would add the following line to the /etc/sysconfig/rawdevices file on the database server:

      ```
      /dev/raw/raw1 /dev/sdd
      ```

   c. After updating the raw device configuration file, you need to start the raw device daemon and reload the raw devices by executing the following commands on the database server:

      ```
      /sbin/service rawdevices restart
      /etc/init.d/rawdevices reload
      ```

   d. Set permission to the raw devices using the following command:

      ```
      chmod 777 <raw device>
      ```

      For example, to set permission to 777 for a raw device named /dev/raw/raw1, you would execute the following command on the database server:

      ```
      chmod 777 /dev/raw/raw1
      ```

      Sometimes permissions for a raw device controller named /dev/rawctl, which acts as the central repository of raw to block device binding information, are not set appropriately. You can change the permission for this controller by executing the following command on the database server:

      ```
      chmod 777 /dev/rawctl
      ```

e. Verify that you can write to the raw devices by executing the following command on the database server:

```
dd if=/dev/zero of=<raw device> bs=4096 count=10
```

For example, to perform a write test on the raw device named /dev/raw/raw1, you would execute the following command on the database server:

```
dd if=/dev/zero of=/dev/raw/raw1 bs=4096 count=10
```

5.2.2 AIX

Here we describe the configuration of a database server for AIX.

Configuring AIX Host for FCP

1. Install IBM N series host Attach Kit.

   We recommend that you download and install the IBM N series provided SAN (FCP) Host Attach Kit for AIX on the database server. This product simplifies the configuration and management of the AIX host in an IBM N series SAN environment. In order to install this product, you need to complete the following basic steps:

   a. IBM System Storage N series FCP AIX Host Attach Kit 4.0 binaries are provided to licensed customers only and can be obtained by contacting IBM support.

   b. Obtain and copy the downloaded software tar file to a work directory (/tmp/nseries) on the database server and uncompress the file using the following command:

   ```
   uncompress /tmp/nseries/ntap_aix_fcp_4.0.tar.Z
   ```

   c. Extract the file by executing the following command on the database server:

   ```
   tar -xvf <file name>
   ```

   For example, you would execute the following command on the database server to extract a file named ntap_aix_fcp_4.0.tar:

   ```
   tar -xvf /tmp/nseries/ntap_aix_fcp_4.0.tar
   ```

   The above command extracts the attach kit software in to the ntap_aix_fcp_1.2 directory.

   d. Change the directory to ntap_aix_fcp_4.0, where you have extracted the attach kit software, and execute the following command on the database server:

   ```
   ./install
   ```

Answer the prompt and complete the installation.

e. Add the following lines to the .profile file found on the database server in the user's home directory:

```
export PATH=$PATH:/opt/NetApp/santools/bin
export MANPATH=/usr/share/man:/opt/NetApp/santools/man
```

f. After installation is complete, reboot the system using the following command:

```
shutdown -F -r now
```

For detailed installation steps, please refer to the "N series FCP AIX Host Attach Kit 4.0 product documentation on the IBM Support for FCP Host Attach Kits site.

2. Install the HBA and driver.

   Before you install HBA on the database server, you need to check the product's compatibility on the IBM support Web site at http://www-03.ibm.com/servers/storage/support/nas/index.html. The compatibility matrix is available on the IBM System Storage N series and TotalStorage® NAS interoperability matrixes. After confirming compatibility, install the HBA and driver on the database server. For more information about the HBA and driver installation, please refer to the product's Installation and configuration guide.

3. Obtain the WWPN for the HBA.

   Each FC HBA attached to the database server is uniquely identified by a WWPN. In order to create an igroup on the IBM N series storage system, the WWPN for the HBA is required. In order to obtain the WWPN, complete the following steps:

   a. After installing the NetApp host Attach kit for AIX, HBA, and the drivers on the database server, you can find the WWPN by executing the following command:

```
sanlun fcp show adapter -c
```

Upon execution with the -c option, the **sanlun** command generates the **igroup create** command, which can be used to create an igroup on the IBM N series storage system. The output from the above command looks similar to the one below:

```
igroup create -f -t aix db2srv1 10000000c93a069b
```

b. The WWPN for adapter in the above example is 10000000c93a069b. Use the generated command to create the igroup.

c. Load the FC driver by executing the following command on the database server:

```
cfgmgr -l fcs0
```

Once the driver is loaded, the database server should able to see the target LUNs on the IBM N series storage system as new devices.

Configuring an AIX Host for iSCSI

1. Install IBM N series iSCSI host Attach Kit

   We recommend that you install the IBM N series provided iSCSI Host Support Kit for AIX. This product simplifies the configuration and management of the IBM N series IP SAN environment and it can be installed in a few simple steps:

   a. Go to the Support for Network Attached Storage (NAS) & iSCSI IBM Web site at http://www-03.ibm.com/servers/storage/support/nas/index.html. Under N series Host Based software, select **iSCSI Host Support Kits**.

   b. Select the **Download** tab, select the **Downloadable files** link and click the **IBM System Storage N series iSCSI Initiator Support Kit Download** link.

   c. Select the **Support kit for AIX**. Accept the terms and conditions.

   d. Select the **iSCSI AIX Initiator Support Kit Version 1.1** and select **Continue** at the bottom of the page

   e. Confirm the license agreement, and select **iSCSI AIX Initiator Support Kit ibm_aix_SAN_kit_1.1.tar.Z (943KB)** and select **Download Now**.

   f. Uncompress it using the following command:

   ```
   uncompress /tmp/nseries/ibm_aix_SAN_kit_1.1.tar.Z
   ```

   g. Extract the file by executing the following command on the database server:

   ```
   tar -xvf <file name>
   ```

For example, you would execute the following command on the database server to extract a file named ibm_aix_SAN_kit_1.1.tar:

```
tar -xvf /tmp/nseries/ibm_aix_SAN_kit_1.1.tar
```

The above command will extract the support kit software into the ibm_aix_SAN_kit_1.1 directory.

h. Change the directory to ibm_aix_SAN_kit_1.1, where you have extracted the support kit software and install the support kit by executing the following command on the database server:

```
./install -f -a /tmp/nseries/ibm_aix_SAN_kit_1.1
```

Answer the prompts and complete the installation.

i. After installing the support kit software, add the following lines to the user's .profile file, found on the home directory for the user on the database server:

```
export PATH=$PATH:/opt/NetApp/santools/bin
export MANPATH=/usr/share/man:/opt/NetApp/santools/man
```

j. Reboot the system by executing the following command on the database server:

```
shutdown -F -r now
```

For detailed installation steps, please refer to *iSCSI IBM AIX Support Kit 1.1 Installation and Setup Guide* on the IBM Web site at:

http://www-03.ibm.com/servers/storage/support/nas/index.html

2. Configure the AIX iSCSI software initiator.

IBM N series supports the iSCSI software initiator for AIX. The iSCSI initiator is included with the IBM AIX 5L™ V5.2 operating system in the maintenance level 3 (ML3) release and above. If you are using an earlier version, then download and install the iSCSI initiator from the IBM Web site at http://www-03.ibm.com/servers/storage/support/nas/index.html. The iSCSI software initiator for AIX can be configured using SMIT as described below:

a. Start SMITTY by executing the following command on the database server:

```
smit
```

b. Select **Devices** → **iSCSI** → **iSCSI Protocol Device**.

c. Select **Change / Show Characteristics of an iSCSI Protocol Device**.

d. Verify if the Initiator's WWNN is correct; it has to be in a specific format, as indicated below:

```
iqn.yyyy-mm.backward_naming_authority:unique_device_name
i.e. - iqn.1992-08.ibmp650.hostid.0xa3cafcd
```

The Initiator's WWNN is required to create an igroup on the IBM N series storage system.

a. Obtain the iSCSI target node name by executing the following command on the IBM N series storage system:

```
iscsi nodename
```

b. Now define the IBM N series storage system as an iSCSI target by adding a line to the /etc/iscsi/targets file found on the database server. The entry in the /etc/iscsi/targets file should look like the one below:

```
[storage system IP] 3260 [storage system nodename]
```

For example, to add an IBM N series storage system that has an IP of 10.32.90.100, as an iSCSI target for the initiator, you would add the following line to the /etc/iscsi/targets file on the database server:

```
10.32.90.100 3260 iqn.1992-08.com.netapp:sn.33606694
```

3. Obtain the WWNN for the initiator.

a. The iSCSI initiator is identified by a unique World Wide Node Name or WWNN. You can obtain the WWNN for an initiator by executing the following command on the database server:

```
lsattr -l iscsi0 -a initiator_name -E
```

b. Verify if the initiator's WWNN is correct. The initiator's WWNN is required for creating an igroup and it has to adhere to a naming format somewhat similar to the following one:

```
iqn.yyyy-mm.backward_naming_authority:unique_device_name
i.e. - iqn.1992-08.ibmp650.hostid.0xa3cafcd
```

If the initiator name is not in the correct format, it can be changed by executing the following command on the database server:

```
chdev  -l 'iscsi0' -a initiator_name=<'new WWNN'>
```

For example, to change the initiator name, you would execute the following command on the database server:

```
Chdev  -l 'iscsi0' -a
initiator_name='iqn.1992-08.ibmp650.hostid.0xa384509'
```

c. Load the iSCSI driver by executing the following command on the database server:

```
cfgmgr -l iscsi0
```

Making LUNs accessible on the database server

1. Once the initiator is refreshed, the database server should be able to discover the LUNs as new devices. You can see these newly added devices by executing the following command on database server:

   ```
   lsdev -Cc disk
   ```

   The output would look similar to Figure 5-9.

```
# lsdev -Cc disk
hdisk0  Available 1S-08-00-8,0 16 Bit LVD SCSI Disk Drive
hdisk1  Available 1S-08-00-9,0 16 Bit LVD SCSI Disk Drive
hdisk14 Available 1D-08-01     NetApp LUN
hdisk15 Available 1D-08-01     NetApp LUN
hdisk16 Available 1D-08-01     NetApp LUN
hdisk17 Available 1D-08-01     NetApp LUN
hdisk18 Available 1D-08-01     NetApp LUN
hdisk19 Available 1D-08-01     NetApp LUN
hdisk20 Available 1D-08-01     NetApp LUN
#
```

Figure 5-9 lsdev command

2. Get the device names by executing the following command:

   ```
   sanlun lun show
   ```

   Upon execution, the **sanlun lun show** command shows the devices along with the assigned names (see Figure 5-10). These devices can be used as tablespace containers for the DB2 database.

```
# sanlun lun show
    filer:         lun-pathname         device filename          adapter
protocol        lun size        lun state
  itsotuc2:     /vol/db2data/lun        hdisk14                     fcs1
FCP      39.9g (42858446848)    GOOD
  itsotuc2:  /vol/db2datanfs/lun        hdisk16                     fcs1
FCP       8.0g (8571060224)     GOOD
  itsotuc2:     /vol/db2logs/lun        hdisk15                     fcs1
FCP      23.9g (25715277824)    GOOD
  itsotuc2:   /vol/db2datacl/lun        hdisk18                     fcs1
FCP      39.9g (42858446848)    GOOD
  itsotuc2:  /vol/db2logsnfs/lun        hdisk17                     fcs1
FCP       8.0g (8571060224)     GOOD
  itsotuc2:   /vol/db2logscl/lun        hdisk19                     fcs1
FCP      23.9g (25715277824)    GOOD
#
```

Figure 5-10 sanlun lun show

3. Verify the attributes of the new disks by entering the following command:

   ```
   lsattr -El <device>
   ```

   For example, to verify the attribute of newly added device named hdisk3, you would execute the following command on the database server:

   ```
   lsattr -El hdisk3
   ```

4. Now create a volume group for each LUN device by issuing the following command on the database server:

   ```
   mkvg -f -y '< VgName >' < devicename >
   ```

   For example, to create a volume group named datavg1 for the device named hdisk3, you would execute the following command on the database server:

   ```
   mkvg -f -y 'datavg1' hdisk3
   ```

   Repeat this process until all the necessary volume groups are created.

5. You can verify the volume groups by executing the following command on the database server:

   ```
   lspv
   ```

If you assigned the wrong name to a Volume Group, it can be changed by executing the following commands:

```
varyoffvg < VolumeGroupName >
exportvg < VolumeGroupName >
mkvg -q -f -y '< NewVG_Name >' < Device >
```

For example, to change a volume group's name from dblogsvg1 to logsvg1, you would execute the following commands on the database server:

```
varyoffvg dblogsvg1
exportvg dblogsvg1
mkvg -q -f -y 'logsvg1' hdisk4
```

6. Create mount points on the database server and assign the appropriate permissions to the database instance by executing the following commands:

   ```
   mkdir -p [mountpoint]
   chown -R db2inst1:db2adm [mount point]
   ```

   For example, to create a mount point named /mnt/db2data and change the ownership to db2inst1 user, you would execute the following command on the database server:

   ```
   mkdir -p /mnt/db2data
   chown =R db2inst1:db2adm /mnt/db2data
   ```

7. Create a new file system on the volume group by executing the following command on the database server:

   ```
   crfs -v jfs -a bf=true -g '< VgName >' -a size=< size > < M|G > -m
   '< mountPoint > -p 'rw' -a nbpi=4096 -a ag=64
   ```

   For example, to create a file system on the volume group named data1vg that is mounted on a mount point named /mnt/db2data, you would execute the following command on the database server:

   ```
   crfs -v jfs -a bf=true -g 'data1vg' -a size=6000M -m
   '/mnt/db2data1/' -p 'rw' -a nbpi=4096 -a ag=64
   ```

   Note: The value for nbpi must be set to 8192, 16384, 32768, 65546, or 131072 if you need to create a file system that is greater than 64 GB in size.

8. To make the file system mount persistent on system reboots, you need to update the file /etc/filesystems. Open the file /etc/filesystems and locate the entries that correspond to the file systems you want to make persistent and set the option `mount = true`. For example, to make a file system named /dev/lvl1 persistent on database server reboot, you would modify the corresponding entry in the /etc/filesystems file; after the update, the entry should look somewhat similar to the entry shown in Figure 5-11.

```
/mnt/db2data:
        dev             = /dev/lv11
        vfs             = jfs
        log             = /dev/loglv10
        mount           = true
        options         = rw
        account         = false
```

Figure 5-11 /etc/filesystems file

Repeat this step for each mounted file system to be used for the database.

9. The tablespace containers that use raw devices reference the character device name assigned to a block device. The character device name corresponding to a block device can be obtained by prefixing the device name with the letter “r”. For example, the character device name for a block device named db2logs, shown in Figure 5-12, should be /dev/rlv12. This is the name that would be used to specify a raw device container for a tablespace.

```
/mnt/db2logs
        dev             = /dev/lv12
        vfs             = jfs
        log             = /dev/loglv10
        mount           = true
        options         = rw
        account         = false
```

Figure 5-12 Character device name

10. Make sure that you can write to the raw device. You can perform this test by executing the following on the database server:

```
dd if=/dev/zero of=/dev/raw/raw1 bs=4096 count=10
```

5.2.3 Sun Solaris

Here we describe the configuration of a database server for Sun Solaris.

Update the kernel parameter

The default values for the kernel configuration parameters might not be the right ones to run DB2 UDB efficiently. For example, the default values for the semsys:seminfo_semume and shmsys:shminfo_shmseg kernel parameters are not appropriate for most of the multi-threaded applications with a fair number of connections. DB2 provides a utility named db2osconf that makes recommendations for the kernel parameter values based on the size of a system. Complete the following steps to set up the appropriate values for the kernel parameters:

1. Use the **db2osconf** command with -t option to get the recommended values:

   ```
   db2osconf -t <number of threads>
   ```

   For example, you would execute the following command on the database server to get the recommended kernel parameter values for 500 threads:

   ```
   db2osconf -t 500
   ```

2. In order to set a kernel parameter value other than the default one, you need to add a line to the /etc/system file:

   ```
   set parameter_name=<new value>
   ```

   For example, to set the value of the parameter msgsys:msginfo_msgmax to 65535, you would add the following line to the /etc/system file found on the database server:

   ```
   set msgsys:msginfo_msgmax=65535
   ```

3. After updating the /etc/system file, you must restart your system.

- For detailed list of recommended kernel parameters, please refer to Chapter 9, “Preinstallation tasks“, in *Quick Beginnings for DB2 Servers*, found at:

  ftp://ftp.software.ibm.com/ps/products/db2/info/vr82/pdf/en_US/db2ise81.pdf

Configure Sun Solaris host for FCP

1. Install the IBM N series SAN (FCP) host attach kit.

   We recommend that you install the SAN (FCP) host attach kit for Solaris on the database server (provided by NetApp). This product simplifies the configuration of the Solaris host in the IBM N series SAN environment. In order to install this product, you need to complete the following steps:

   a. IBM System Storage N series FCP Solaris Host Attach Kit 4.0 binaries are provided to licensed customers only and can be obtained by contacting IBM support.

   b. Obtain and copy the downloaded software tar file to a work directory (/tmp/nseries) on the database server and uncompress the file by executing the following command on the database server:

      ```
      uncompress /tmp/netapp/NTAPasl.4.0.2.tar.Z
      ```

   c. After uncompressing, you need to extract the file by executing the following command on the database server:

      ```
      tar -xvf <file name>
      ```

      For example, you would execute the following command on the database server to extract the software tar file named NTAPasl.4.0.2.tar:

      ```
      tar -xvf /tmp/nseries/NTAPasl.4.0.2.tar
      ```

   d. To install the attach kit software, you need to change the directory to ntap_solaris_fcp_2_0, where you have extracted the software, and execute the following command on the database server:

      ```
      ./install
      ```

      Answer the prompt and complete the installation.

   For detailed installation steps, please refer to Support for FCP Host Attach Kits on the IBM Web site at:

   http://www-03.ibm.com/servers/storage/support/nas/index.html

2. Install the HBA and driver.

   Go to the IBM System Storage N series and TotalStorage NAS interoperability matrixes and check for your products compatibility. For the installation steps, please refer to *The FCP Solaris Host Attach Kit 4.0 Installation and Setup Guide* on the Support for Network Attached Storage (NAS) & iSCSI Web site:

   http://www-03.ibm.com/servers/storage/support/nas/index.html

   a. After installing the initiator, create an igroup for it and map the LUNs to the igroup. The database server must be configured to make use of the new resources. In order for LUN mappings to take effect, you need to augment the /kernel/drv/sd.conf file with additional entries. The entry should look somewhat similar to the following:

      ```
      name="sd" parent="lpfc" target=<targetID> lun=<lunID>;
      ```

      For example, to make LUN ID 0 and 1 accessible on the target ID 0, you would add the following lines to the /kernel/drv/sd.conf file on the database server:

      ```
      name="sd" parent="lpfc" target=0 lun=0;
      name="sd" parent="lpfc" target=0 lun=1;
      ```

   b. After any modification is made to the kernel/drv/sd.conf file, the system must be rebooted before it will take effect. A reconfiguration reboot will scan for and create the new device entries. Execute the following command at the database server:

      ```
      Reboot -- -r
      ```

   c. After adding LUN devices, you can execute the following command on the database server to refresh the driver:

      ```
      Devfsadm
      ```

3. Obtain the WWPN for the initiator.

   Each FC HBA attached to your database server is uniquely identified by WWPN. In order to create the igroup on the IBM N series storage system, you will require the WWPN for the HBA. After installing the FC adapter and driver, you can obtain the WWPN by executing the following **sanlun** command on the database server:

   ```
   sanlun fcp show adapter -c
   ```

   Upon execution with the -c option, the **sanlun** command generates the **igroup create** command, which can be used to create an igroup on the IBM N series storage system. The output from the above command should look like the following:

   ```
   igroup create -f -t solaris db2srv1 10000000c93a069b
   ```

   The WWPN for initiator in the above example is 10000000c93a069b.

Sun Solaris host configuration for iSCSI

1. Install IBM N series iSCSI Host Support Kit.

   We recommend that you install the IBM N series provided iSCSI Host Support Kit for Solaris. This product simplifies the configuration and management of the SAN environment. In order to install this product, you need to complete the following basic steps.

   a. Go to the Support for Network Attached Storage (NAS) & iSCSI IBM Web site at http://www-03.ibm.com/servers/storage/support/nas/index.html. Under N series Host Based software, select **iSCSI Host Support Kits**.

   b. Select the **Download** tab, select the **Dowloadable files** link, and click the **IBM System Storage N series iSCSI Initiator Support Kit Download** link.

   c. Select the **Support kit for Solaris**. Accept the terms and conditions.

   d. Select the **iSCSI Solaris Initiator Support Kit Version 1.0** and select **Continue** at the bottom of the page.

   e. Confirm the license agreement, select **iSCSI Initiator Support Kit Solaris (amd) santoolkit_solaris_amd_ibm_v1.0.0.tar.Z (198KB)** or **iSCSI Initiator Support Kit Solaris (sparc) santoolkit_solaris_sparc_ibm_v1.0.0.tar.Z (447KB)** and select **Download Now**.

   f. Uncompress it using the following command:

      ```
      uncompress /tmp/nseries/santoolkit_solaris_amd_ibm_v1.0.0.tar.Z
      ```

      Extract the file by executing the following command at the database server:

      ```
      tar -xvf <file name>
      ```

      For example, you would execute the following command on the database server to extract a tar file named santoolkit_solaris_amd_ibm_v1.0.0.tar:

      ```
      tar -xvf /tmp/nseries/ santoolkit_solaris_amd_ibm_v1.0.0.tar
      ```

   g. Change to the santoolkit_solaris_amd_ibm_v1.0.0 directory where you extracted the software and execute the following command on the database server:

      ```
      ./install
      ```

   This will complete the installation of Host Support Kit.

2. Install the HBA and driver.

   IBM N series supports Qlogic iSCSI HBA for iSCSI on Solaris. You need to download the iSCSI driver for the initiator and SANSurfer iSCSI Command line utility from the Qlogic web site:

   http://www.qlogic.com/products/iscsi_products_hba.asp

   We have used the QLogic™ SANsurfer iSCSI Command Line tools for configuring the initiator. Log in as root to your database server and perform the following basic installation steps:

   a. Download the HBA driver and SANSurfer CLI tools from the Qlogic Web site to a local directory.

   b. Copy the downloaded driver and SANSurfer CLI tools file to a work directory (/tmp/nseries) on the database server and uncompress the files by executing the following command on the database server:

      ```
      uncompress /tmp/nseries/qla4010.z
      uncompress /tmp/nseries/iscli-1.0.36-2_solaris_sparc_x86.Z
      ```

   c. If you already have a previous installation of the QLogic driver, remove it by executing the following command on the database server:

      ```
      pkgrm QLA4010-0
      ```

      Make sure you reboot the system after removing the old driver.

   d. Install the driver by executing the following command on the database server:

      ```
      pkgadd -d /tmp/nseries/qla4010
      ```

   e. After starting the **pkgadd** command, follow and answer the prompts displayed to finish the installation.

   f. Install the Qlogic SANSurfer iSCSI CLI tools by executing the following command on the database server:

      ```
      pkgadd -d /tmp/nseries/iscli-1.0.36-2_solaris_sparc_x86
      ```

      This completes the Installation for an add-on HBA Driver and the SANSurfer CLI tools.

   g. Now shut down the system and insert the QLogic HBA into an empty PCI slot on the database server and connect the adapter to your iSCSI target IBM N series storage system.

   h. Update the /kernel/drv/sd.conf file found on your database server by adding the following line for each target LUN:

      ```
      name="sd" class="scsi" class_prop="atapi" target=<N> lun=<M>;
      ```

      Where "N" is the target ID and "M" is the LUN ID.

For example, to add an entry for LUN ID 0 that is accessible on target ID 0, you would add the following line to the /kernel/drv/sd.conf file on the database server:

```
name="sd" class="scsi" class_prop="atapi" target=0 lun=0;
```

For detailed installation steps, please download and read the readme.txt file from the Qlogic Web site for the HBA you have installed.

i. Now, reboot the system by executing the following command at the database server:

```
reboot -- -r
```

3. Obtain the WWNN for the initiator.

Each iSCSI HBA attached to your database server is uniquely identified with a WWNN. In order to create an igroup on the IBM N series storage system, you will need a WWNN for the HBA. You can find the WWNN by executing the following Qlogic SANSurfer iSCSI CLI command on the database server:

```
iscli -i
```

The output from the above command should look somewhat similar to the following one:

```
iqn.1992-08.sunv20z.svl12.0xa3cafcd
```

To manage and configure Qlogic HBA on a Sun Solaris host in an IBM N series IP-SAN environment, you can also use the Qlogic SANSurfer iSCSI HBA Manager.

Making LUNs accessible on the database server

1. After LUNs are discovered, execute the **format** command to create a partition table on the new devices on the database server:

```
format
```

Upon execution of the **format** command, a numbered list of available disks is displayed.

2. Type the number for the disk from the list that you want to repartition:

```
Specify disk (enter its number): disk-number
```

For example, you would specify 7 in order to create the partition and format disk number 7:

```
Specify disk (enter its number): 7
```

3. This is first time the LUN is used as device, so you will be prompted to create a label for the disk. Enter `no` at the prompt.

```
Disk not labeled. Label it now? No
```

4. Select the partition from the menu and type `print` to display the current partition table.

```
format>part
Partition>print
```

5. Modify the predefined partition table, reserve one cylinder at the head of the disk for the partition table using the all free hog option for slice 6, as shown in Figure 5-13.

```
partition> modify
Select partitioning base:
        0. Current partition table (default)
        1. All Free Hog
Choose base (enter number) [0]? 1
Do you wish to continue creating a new partition
table based on above table[yes]? Yes
Free Hog partition [6]? 6
Enter size of partition '0' [0b, 0c, 0.00mb, 0.00gb]: 1c
Enter size of partition '1' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '3' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '4' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '5' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '7' [0b, 0c, 0.00mb, 0.00gb]:
Part      Tag    Flag     Cylinders        Size            Blocks
  0       root    wm       0 -    0        1.00MB    (1/0/0)           2048
  1       swap    wu       0               0         (0/0/0)              0
  2     backup    wu       0 - 6141        6.00GB    (6142/0/0)    12578816
  3 unassigned    wm       0               0         (0/0/0)              0
  4 unassigned    wm       0               0         (0/0/0)              0
  5 unassigned    wm       0               0         (0/0/0)              0
  6        usr    wm       1 - 6141        6.00GB    (6141/0/0)    12576768
7 unassigned    wm       0               0         (0/0/0)              0
```

Figure 5-13 modify command

6. Create the new partition table by typing `Yes` on the prompt and entering the table name, as shown in Figure 5-14.

```
Okay to make this the current partition table[yes]? Yes
Enter table name (remember quotes): db2data1
```

Figure 5-14 Table name

7. Create the label and complete the formatting operation on the database server, as shown in Figure 5-15.

```
Ready to label disk, continue? Y
partition> quit
format> quit
```

Figure 5-15 label disk

8. Create the file system on the disk by executing the following command at the database server:

   ```
   newfs [-b size] [-i bytes] /dev/dsk/< device-name >
   ```

   Where:

   -b size The block size for the file system, either 4096 or 8192 bytes per block. The default value is 8192.

   -i bytes The number of bytes per inode. The default varies depending on the disk size. This value should reflect the expected average size of files in the file system. If fewer inodes are desired, a larger number should be used. To create more inodes, a smaller number should be given. The default for nbpi is as follows:

Disk size	Density
Less than 1 GB	2048
Less than 2 GB	4096
Less than 3 GB	6144
3 GB to 1 TB	8192
Greater than 1 TB	1048576

   The number of inodes can increase if the file system is expanded with the **growfs** command.

   device-name The disk device name on which to create the new file system.

   For example, to create a file system on a device identified as c3t0d0s6, you would execute the following command on the database server:

   ```
   newfs -i 12000 -b 8192  /dev/dsk/c3t0d0s6
   ```

5.2.4 HP-UX

Here we describe the configuration of a database server for HP-UX.

Set up kernel parameters

Before installing DB2 UDB, we recommend that you update your system's kernel configuration parameters. IBM provides a utility, **db2osconf**, for DB2, which makes recommendations for kernel parameter values based on the size of a system. Complete the following setups to modify the kernel parameters' values:

1. Use the **db2osconf** command with the -t option to get the recommended kernel parameter values:

   ```
   db2osconf -t <number of threads>
   ```

   For example, you would execute the following command on the database server to get the recommended kernel parameter values for 500 threads:

   ```
   db2osconf -t 500
   ```

2. Log in to the database server as root and start the System Administration Manager (SAM) by executing the following command at the database server:

   ```
   sam
   ```

3. Select the **Kernel Configuration** parameter from the list.
4. Select the **Configurable Parameters** from the list.
5. Select the parameter that you want to change and press the Tab key.
6. Select the **Action** → **Modify Configuration Parameter** from the main menu and enter the new value in the Formula/Value field.
7. Press the Tab key and then **OK**.
8. Repeat these steps for each of the kernel configuration parameters that you want to change.
9. When you are finished setting all of the kernel configuration parameters, select **Action** → **Process New Kernel** from the action menu bar.
10. In the next window, confirm your modification by selecting **Yes**. This will update the kernel parameter and reboot your server automatically.

- For a detailed list of recommended kernel parameters, please refer to Chapter 9, "Preinstallation tasks", in *Quick Beginnings for DB2 Servers*, found at:

  ftp://ftp.software.ibm.com/ps/products/db2/info/vr82/pdf/en_US/db2ise81.pdf

Configure HP-UX host system for FCP

1. Install the IBM SAN (FCP) host Attach kit.

   We recommend that you install the IBM provided SAN (FCP) host attach kit for HP-UX. This product simplifies the configuration and management of the HP-UX host component of the IBM N series SAN environment. In order to install this product, you need to complete the following basic steps on the database server:

   a. IBM System Storage N series FCP HP Host Attach Kit 1.2 binaries are provided to licensed customers only and can be obtained by contacting IBM support.

   b. Obtain and copy the downloaded software tar file to a work directory (/tmp/netapp) on the database server and uncompress the file using the following command:

      ```
      gunzip /tmp/netapp/ntap_hpux_fcp_1_2.depot.gz
      ```

   c. Install the NetApp Host Attach kit by executing the following command on the database server:

      ```
      /usr/sbin/swinstall -s /tmp/netapp/ntap_hpux_fcp_1_2.depot
      ```

   d. Add the following lines to the .profile file found on the database server in the user's home directory:

      ```
      export PATH=$PATH:/opt/NetApp/santools/bin
      export MANPATH=/usr/share/man:/opt/NetApp/santools/man
      ```

For detailed installation steps, please refer to FCP HP-UX Host Attach Kit 1.2 Installation and Setup Guide on the IBM web site at:

http://www-03.ibm.com/servers/storage/support/nas/index.html

2. Install the HBA and driver

   In order to install the HBA, complete the following steps on your database server:

   a. Go to the IBM System Storage N series and TotalStorage NAS interoperability matrixes on the IBM support site at:

      ftp://service.boulder.ibm.com/storage/nas/nseries/nseries_interop.pdf

   b. Check the FCP Host Compatibility Matrix and verify the product compatibility for your database server products.

   c. Download the HBA driver from the HP Web site (http://software.hp.com). You should also download the HBA installation guide from the HP Web site (http://docs.hp.com/en/netcom.html).

d. Follow the instructions described in the HBA installation and configuration guide and install the card and driver on the database server.

3. Configure the initiator and the WWPN.

   a. Each FC HBA attached to your database server has a unique identification known as a WWPN. In order to create an igroup on the IBM N series storage system, you will need the WWPN for the HBA. After installing the HBA and NetApp host Attach kit for HP-UX, you can obtain the WWPN by executing the following command on the database server:

      ```
      sanlun fcp show adapter -c
      ```

      Upon execution with the -c option, the **sanlun** command generates the **igroup create** command. The output from the above command looks similar to the following:

      ```
      igroup create -f -t hpux db2srv1 50060b00002cf7c8
      ```

      The WWPN for initiator in the above example is 50060b00002cf7c8.

      You can use the **igroup create** command to create an igroup on the IBM N series storage system.

   b. Enter the following command on the database server to create device nodes for the IBM N series storage system LUNs:

      ```
      ioinit -i
      ```

      Use the **insf -e** command if the **ioinit -i** command does not create device nodes for all of the paths to the LUN.

   c. IBM N series storage system LUNs are not automatically visible to the database server. To perform LUN discovery, you need to execute the following command on the database server:

      ```
      ioscan -fn -C disk
      ```

Configure HP-UX host system for iSCSI

1. Install the ISCSi Host Support Kit.

   We recommend that you install the IBM N series provided iSCSI Host Support Kit for HP-UX. This product simplifies the configuration and management of the SAN environment. In order to install this product, you need to complete the following basic steps:

   a. Go to the Support for Network Attached Storage (NAS) & iSCSI IBM Web site at http://www-03.ibm.com/servers/storage/support/nas/index.html. Under N series Host Based software, select **iSCSI Host Support Kits**.

   b. Select the **Download** tab, select the **Dowloadable files** link, and click the **IBM System Storage N series iSCSI Initiator Support Kit Download** link.

   c. Select the **Support kit for HP-UX**. Accept the terms and conditions.

   d. Select the **iSCSI HP-UX Initiator Support Kit Version 1.2** and select **Continue** at the bottom of the page.

   e. Confirm the license agreement, select **iSCSI Initiator HP-UX Support Kit ntap_hpux_iscsi_1.2.depot.gz (185KB)**, and select **Download Now**.

   f. Copy the downloaded support kit file to a work directory (/tmp/nseries) on the database server and uncompress the file using the following command:

      ```
      gunzip /tmp/nseries ntap_hpux_iscsi_1.2.depot.gz
      ```

   g. Install the support kit by executing the following command on the database server:

      ```
      swinstall -s /tmp/nseries/ntap_hpux_iscsi_1.2.depot iscsitools
      ```

The diagnostic scripts are installed to the /opt/NetApp/iscsitools/bin directory.

   h. Add the following lines to the .profile file found on the database server in the user's home directory:

      ```
      export PATH=$PATH:/opt/NetApp/santools/bin
      export MANPATH=/usr/share/man:/opt/NetApp/santools/man
      ```

2. Install the iSCSI initiator.

   IBM N series supports a software initiator for HP-UX. In order to install the HP-UP iSCSI Software Initiator, complete the following steps.

a. Go to Support for Network Attached Storage (NAS) & iSCSI on the IBM Web site and go to the NAS interoperability and compatibility and IBM System Storage N series and TotalStorage NAS interoperability matrixes page:

   ftp://service.boulder.ibm.com/storage/nas/nseries/nseries_interop.pdf

   Check the iSCSI Host Compatibility Matrix and find the supported initiator details for your database server products.

b. Download the iSCSI software initiator from the HP Web site:

   http://software.hp.com

   Enter iSCSI Software Initiator in the search box. When the search results show the iSCSI Software Initiator, click the **Receive for Free** button.

c. Follow the prompts to reach the Software download page and download the software initiator file and installation instructions to a local directory.

d. Copy the downloaded initiator file to a work directory (/tmp/nseries) on the database server.

e. Install the initiator by executing the following command on the database server:

```
swinstall -x autoreboot=true -s
/tmp/nseries/iSCSI-00_B.11.23.03e_HP-UX_B.11.23_IA+PA.depot
iSCSI-00
```

   The `autoreboot=true` option will cause a system reboot after the installation is complete.

f. Add the following line to the ./profile file found in the home directory of the user on the database server:

```
export PATH=$PATH:/opt/iscsi/bin
```

g. Download the *HP-UX iSCSI Software Initiator Support Guide* from http://docs.hp.com and refer to the detailed installation and configuration steps.

3. Configure the initiator and obtain the WWNN.

   a. After installing the iSCSI initiator, execute the following command on the database server to obtain its WWNN:

```
/opt/iscsi/bin/iscsiutil -l
Initiator Name
:iqn.1980-03.com.hp:cx2600-1.d25ca127-6ec1-11d7-a3a1-9759e748a354
```

Use this value of WWNN to create an igroup on the IBM N series storage system. The initiator's WWNN has to be in a specific format, as indicated below:

```
iqn.yyyy-mm.backward_naming_authority:unique_device_name
```

If the WWNN is not in the required format, then you can change the WWNN value by executing the following command on the database server:

```
iscsiutil /dev/iscsi -i -N <iSCSI-initiator-name>
```

For example, to change initiator's WWNN from iqn.1980-03.com.hp:cx2600-1.d25ca127-6ec1-11d7-a3a1-9759e748a354 to iqn.1986-03.com.hp:hpfcs214.2000853943, you would execute the following command on the database:

```
iscsiutil /dev/iscsi -i -N iqn.1986-03.com.hp:hpfcs214.2000853943
```

b. After installing the initiator and its driver, the iSCSI service needs to be started on the database server. You would execute the following command on the database server to start the iSCSI service:

```
/sbin/init.d/iscsi start
```

c. Each iSCSI target used for the database need to be identified by the database server persistently over the system reboots. In order to make the target's IBM N series storage system identification persistent, you need to add its name or IP address to kernel configurations by executing the following command on the database server:

```
 iscsiutil [/dev/iscsi] -a -I <ip-address> [-P <tcp-port>] [-M
<portal-grp-tag>]
```

For example, you would execute the following command on the database server to use the static identification for the IBM N series storage system:

```
iscsiutil /dev/iscsi -a -I nseries
```

d. To make database server discover IBM N series storage system LUNs, you would execute the following command on the database server:

```
ioscan -H 255
```

You can check the newly discovered devices by executing the following command on the database server:

```
ioscan -funC disk
```

e. Enter the following command on the database server to create device nodes for the IBM N series storage system LUNs:

```
ioinit -i
```

Use the **insf -e** command if the **ioinit -i** command does not create device nodes for all of the paths to the LUN.

Making LUNs accessible on the database server

In order to create a database on the IBM N series storage system LUN devices, you need to create physical volumes, volume groups, and logical volumes on them. You need to complete the following steps to create the file systems and configure the newly added devices:

1. The database server assigns a device name to each LUN it discovers. Obtain the assigned device name for each LUN by executing the following **sanlun** command:

   ```
   sanlun lun show -p < storage system Name|IP >:< lunpath >
   ```

   For example, to find a device name for the LUN named /vol/db2data/data1 on an IBM N series storage system named nseries, you would execute the command shown in Figure 5-16 on the database server.

```
sanlun lun show -p nseries:/vol/db2data/data1

nseries:/vol/db2data/data1 (LUN 0)
     75g (80530636800)     lun state: GOOD
Storage system_CF_State: Cluster Enabled  Multipath_Policy: None
Multipath-provider: None
--------  --------- ---------------  -------- ---------- ----------
host path filer path      /dev/dsk   host      primary    partner
state     type            filename   HBA       filer port filer port
--------  --------- ---------------  --------  ---------- --------
up         FCP       /dev/dsk/c4t0d0 lan1     10.32.90.31
```

Figure 5-16 sanlun lun show

The block device name in the output from the above command is /dev/dsk/c4t0d0. The corresponding character device name will be /dev/rdsk/c4t0d0.

2. You can create a physical volume on the IBM N series storage system LUN by specifying its character device name as indicated below:

   ```
   pvcreate <character device name>
   ```

   For example, to create a physical volume on the character device named /dev/rdsk/c4t0d0, you would execute the following command on the database server:

   ```
   pvcreate /dev/rdsk/c4t0d0
   ```

3. Execute the following command on the database server to display a list of existing volume groups:

```
ls -l /dev/*/group
crw-r-----  1 root sys   64 0x000000 Mar 28 10:59 /dev/vg00/group
```

Pick an unused minor device number for the new volume group. For example, in the output of the previous command, the used minor device number is 0, so the next unused minor device number is 1. Set a shell variable containing the next unused minor device number:

```
VG=1
```

4. Before a volume group is created, a device node directory for the volume group must exist, the minor device number needs to be converted to hexadecimal base, and a device node needs to be created. Enter the following command on the database server to create a device node directory:

```
mkdir -p /dev/< vg_name >
```

For example, to create a device node directory for the volume group named vg01_db2data, you would execute the following command on the database server:

```
mkdir -p /dev/vg01_db2data
```

Enter the following command on the database server to convert the minor device number to hexadecimal:

```
VGH=$(echo 16o $VG p|dc)
```

Execute the following command to create a device node for the volume group:

```
mknod /dev/[vg_name]/group c 64 ox<hex minor device number>0000
```

For example, to create a device node for the volume group vg01_db2data, you would execute the following command on the database server:

```
mknod /dev/vg01_db2data/group c 64 0x${VGH}0000
```

Repeat steps 1 through 4 for each LUN you want to use for the database.

5. Now create a volume group by executing the following command on database server:

```
vgcreate [vg-name]  /dev/dsk/[device], …
```

For example, you would execute the following command on the database server to create a volume group named vg01_db2data:

```
vgcreate vg01_db2data /dev/dsk/c4t0d0, /dev/dsk/c4t0d1
```

A single volume group can have multiple devices . Depending on the size of volume group you are creating, you need to specify an extent size in the **vgcreate** command. The total storage in the volume group can be divided into

a maximum of 65535 extents. The extent size can be power of 2 between 1 and 256 MB.

6. Create a logical volume on the volume group by executing the following command on the database server:

   ```
   lvcreate -L size -n <logicalvolumename> <volumegroup>
   ```

   For example, to create a 1000 MB logical volume named db2data within the vg01_db2data volume group, you would execute the following command on the database server:

   ```
   lvcreate -L 1000 -n db2data vg01_db2data
   ```

   Repeat this step until all the necessary logical volumes have been created. You can create up to 255 logical volumes in one volume group.

7. To see the volume group detail, execute the following command on the database server:

   ```
   vgdisplay -v vg01_db2data
   ```

8. If database tablespace containers use raw devices, then skip this step and proceed with the next step.

   a. Create a new file system on each logical volume by executing the following command on the database server:

      ```
      newfs -F vxfs -o largefiles /dev/< vg_name >/< lv_name >
      ```

      For example, to create a file system on the logical volume named db2data, you would execute the following command on the database server:

      ```
      newfs -F vxfs -o largefiles /dev/vg01_db2data/db2data
      ```

      Repeat this step until the file systems have been created on all the required logical volumes.

   b. Create the mount points and change the ownership to the database instance owner by executing the following commands on the database server:

      ```
      mkdir -p /mnt/<dir name>
      chown -R db2inst1:db2adm <dir name>
      ```

      For example, to create a mount point named /mnt/db2data and change its ownership to the database instance owner named db2inst1, you would execute the following command on the database server:

      ```
      mkdir -p /mnt/db2data
      chown -R db2inst1:db2adm /mnt/db2data
      ```

      Repeat this step and create all the required mount points for your environment.

c. Now mount the file system using the newly created mount points by executing the following command on the database server:

```
mount -F vxfs /dev/[vg_name]/[lv_name] [mount point]
```

For example, to mount a file system to a mount point named /mnt/db2data, you would execute the following command on the database server:

```
mount -F vxfs /dev/vg01_db2data/db2data /mnt/db2data
```

d. In order to make the file system mounting persistent on the database server reboot, you need to add an entry for each file system you have created to the /etc/fstab file found on the database server. The entry in the /etc/fstab file should look somewhat similar to the following one:

```
/dev/[vg_name]/[lv_name] [mount point] vsfx delaylog 0 2
```

For example, you would add the following line to the /etc/fstab file on the database server to make the mount persistent for the logical volume named db2data on the mount point named /mnt/db2data:

```
/dev/vg01_db2data/db2data1 /mnt/db2data vsfx delaylog 0 2
```

9. If raw devices are desired as the tablespace containers, then reference the logical volume name prefixed with an “r” as the storage container name. For example, the raw device name for the logical volume /dev/vg01_db2data/db2data1 would be /dev/vg01_db2data/rdb2data1. This is the name that would be used to specify the container for a raw DMS tablespace.

6

Creating a new database on an IBM N series storage system

This chapter describes the steps needed to create a new DB2 UDB database whose data and transaction log files reside on the IBM N series storage system volume(s)/LUN(s). The creation of DB2 database is a very straightforward process that utilizes standard DB2 UDB commands. The sections in this chapter will allow you to create the DB2 database.

6.1 Install IBM DB2 UDB

Log in as the instance owner and install the IBM DB2 UDB on the database server. For installation instructions, please refer to *Quick Beginnings for DB2 Servers*, found at:

ftp://ftp.software.ibm.com/ps/products/db2/info/vr82/pdf/en_US/db2ise81.pdf

6.2 Create a new database

You can create a new database on an IBM N series storage system volume that is mounted to a mount point on the database server. You can do so by executing the following command on the database server:

```
db2 "CREATE DATABASE <database name> on <mount point>"
```

The mount point name has to be explicitly specified in the CREATE DATABASE command.

For example, to create a database named MYDB on an IBM N series storage system volume named db2data that is mounted on a mount point named /mnt/db2data, you would execute the following command on the database server:

```
db2 "CREATE DATABASE MYDB ON /mnt/db2data"
```

6.3 Transaction logs

By default, when a database is first created, circular logging is used and the primary log files are created in the subdirectory <dbpath>/<instancename>/<nodename>/*SQLnnnn*/SQLOGDIR/.

Where SQLnnnn is the number referring to the database in the instance. If you have only one database in the instance, then this value will be SQL0001.

We recommend that you move transaction log files to another volume on the IBM N series storage system. The new transaction log location can be specified by updating the registry parameter NEWLOGPATH. You would execute the following command on the database server in order to update this registry parameter:

```
db2 "UPDATE DB CFG FOR MYDB USING NEWLOGPATH <new log location>"
```

For example, to move transaction logs for a database named mydb to another volume named db2logs on the IBM N series storage system that is mounted on a

mount point named /mnt/db2logs, you would execute the following command on the database server:

```
db2 "UPDATE DB CFG FOR MYDB USING NEWLOGPATH /mnt/db2logs"
```

Keep in mind that, in most cases, changes made to a database configuration file do not take effect until the database has been restarted after terminating all the database connections.

6.4 Database verification

You can verify If the database is created correctly and exists on the IBM N series storage system by executing the following command on the database server:

```
db2 "LIST DB DIRECTORY"
```

If the database was created successfully, its name should appear in this list, along with information about where the overhead files for this database are stored.

6.5 Establishing a database connection

Assuming the database was created successfully, establish a connection to it by issuing the following command:

```
db2 "CONNECT TO [DB_NAME]"
```

Where DB_NAME is the alias assigned to the database that was created (if no alias was specified, the alias will be the same as the name assigned to the database).

For example, you would execute the following command on the database server to connect to the database named mydb:

```
db2 "CONNECT TO mydb"
```

6.6 List and verify tablespaces

List the tablespaces that were created with the database by executing the following command on the database server:

```
db2 "LIST TABLESPACES"
```

Verify that the tablespace container(s) associated with each tablespace refer to a file or directory located on an IBM N series storage system volume by executing the following command:

```
db2 "LIST TABLESPACE CONTAINER [TSPACE_ID] SHOW DETAIL"
```

Where TSPACE_ID is the unique identifier (0, 1, 2, and so on) that has been assigned to the tablespace that has container information retrieved. If the database was created correctly, the container information returned by this command should identify the file or directory that was assigned to the tablespace specified when the CREATE DATABASE command was executed.

For example, to verify the tablespace container for tablespace ID 2, you would execute the following command on the database server:

```
db2 "LIST TABLESPACE CONTAINER 2 SHOW DETAIL"
```

Additional tablespaces can be created on IBM N series storage system volume using the CREATE TABLESPACE command.

7

Migrating existing DB2 UDB databases to an IBM N series storage system

The migration of existing DB2 databases to an IBM N series storage system volume is also a fairly straightforward process that utilizes standard DB2 UDB commands. This chapter describes the steps needed to convert an existing DB2 UDB database where the data and transaction log files reside on local disk to a new DB2 UDB database where the data and transaction log files reside on an IBM N series storage system volume (see Figure 7-1 on page 74 for details).

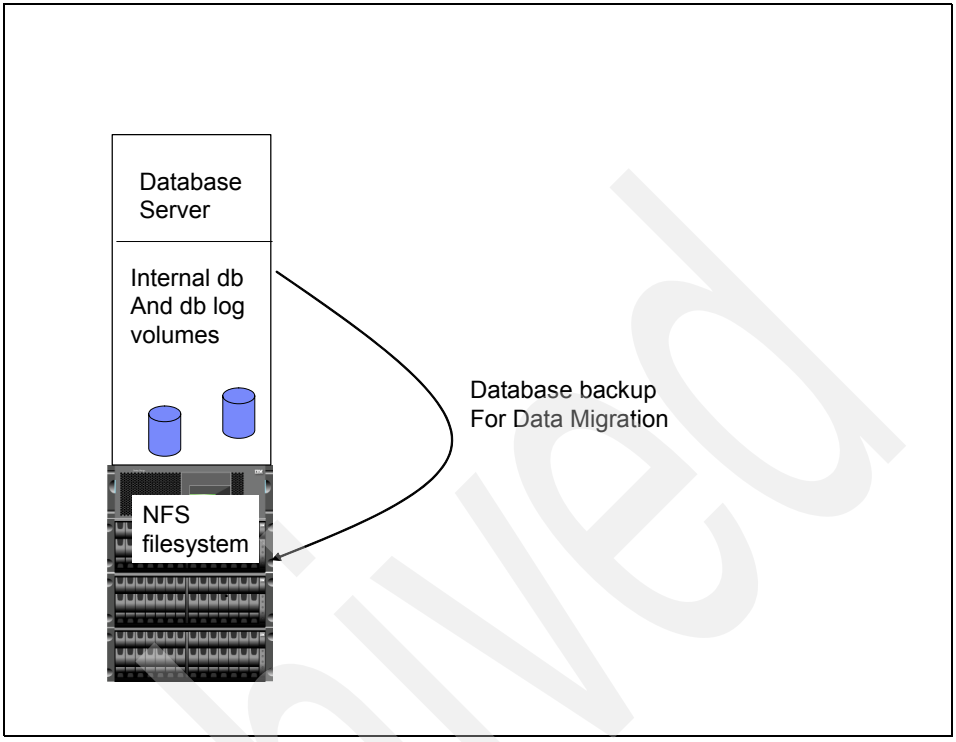


Figure 7-1 Database migration

Note: The items described in Chapter 2, “Requirements” on page 3 are required in order for this process to work correctly.

Important: Always ensure that you have fully recoverable backup copies of your database and related files before attempting to migrate an existing database to an IBM N series storage system volume.

7.1 Migrating a DB2 database using a redirected restore operation

The easiest way to physically move a DB2 database from one storage location to another is by backing up the database with DB2's BACKUP DATABASE command and then performing what is known as a *redirected restore* operation to copy the database to its new location. Redirected restore operations are performed by executing a combination of DB2's RESTORE DATABASE and SET TABLESPACE CONTAINERS commands once a backup image of the database has been obtained. The steps needed to perform this type of DB2 database migration are:

1. Establish a connection to the database to be migrated by issuing the following command:

   ```
   db2 "CONNECT TO [DB_NAME]"
   ```

   Where - DB_NAME is the database alias or name.

   For example, you would execute the following command on the database server to connect to a database named mydb:

   ```
   db2 "CONNECT TO mydb"
   ```

2. List the tablespaces that have been created for the database to be migrated by executing the following command:

   ```
   db2 "LIST TABLESPACES" > tablespace.out
   ```

3. Stop and restart the DB2 Database Manager instance (to make the change take effect) by executing the following commands:

   ```
   db2 "FORCE APPLICATIONS ALL"
   db2stop
   db2start
   ```

4. Create a full backup image of the database to be migrated by issuing the following command on the database server:

   ```
   db2 "BACKUP DATABASE [DB_NAME] TO [BKUP_LOCATION]"
   ```

   Where- DB_NAME is the database alias or name and BKUP_LOCATION is the location where the database backup image is to reside.

   For example, to back up a database named mydb on a location named/dbbkup/mydb, you would execute the following command on the database server:

   ```
   db2 "BACKUP DATABASE mydb TO /dbbkup/mydb"
   ```

5. Create a script file that contains the commands needed to perform a redirected restore. This script file should contain the following commands:

```
db2 "RESTORE DATABASE [OLD_DB_NAME] FROM [BKUP_LOCATION] TO
[NEW_LOCATION] INTO [NEW_DB_NAME] REDIRECT"
db2 "SET TABLESPACE CONTAINERS FOR [TSPACE_ID] USING
('[DIRECTORY]')"
```

or

```
db2 "SET TABLESPACE CONTAINERS FOR [TSPACE_ID] USING (FILE
'[FILE_NAME]' [SIZE])"
```

or

```
db2 "SET TABLESPACE CONTAINERS FOR [TSPACE_ID] USING (DEVICE
'[DEVICE_NAME]' [SIZE])"
db2 "RESTORE DATABASE [OLD_DB_NAME] CONTINUE"
```

Where:

- OLD_DB_NAME is the alias assigned to the database that was backed up, and BKUP_LOCATION is the location where the database backup image resides.
- NEW_LOCATION is the location where the overhead files associated with the database are to reside.
- NEW_DB_NAME is the name that is to be assigned to the new database that will be created.
- TSPACE_ID is the unique identifier (0, 1, 2, and so on) that has been assigned to the tablespace that has the new container information assigned to it.
- DIRECTORY is the new directory that stores the SMS tablespace data.
- FILE_NAME is the name of the file that stores the DMS tablespace data.
- DEVICE_NAME is the name of the device that stores the tablespace data.
- SIZE is the size, in 4, 8, 16, or 32 KB pages, of the file or device specified.

For example, to restore a database named mydb database from a backup location named /dbbkup/mydb and move the tablespace containers to a IBM N series storage system volume, your script should look somewhat similar to the following:

```
DumpDir=/dbbkup/mydb
NewDBDIR=/mnt/db2data/mydb
OLD_DB_NAME=mydb
NEW_DB_NAME=mydb
db2 "restore database $OLD_DB_NAME from $DumpDir TO $NewDBDIR INTO
$NEW_DB_NAME redirect without prompting" > restore.out
```

```
db2 "SET TABLESPACE CONTAINERS FOR 0 USING (PATH $NewDBDIR/catalog)"
db2 "SET TABLESPACE CONTAINERS FOR 1 USING (PATH $NewDBDIR/temp)"
db2 "restore db $OLD_DB_NAME continue"
```

Note: Using the information collected in Step 3, you will need to create **`set tablespace container for`** commands for each tablespace found in the database.

Execute the script file just created to perform a redirected restore operation.

6. Verify that the database was successfully migrated by performing a few simple tests (as outlined in 6.4, “Database verification” on page 71).

7.2 Migrating a DB2 database using db2look and db2move

Earlier, it was mentioned that the easiest way to physically move a DB2 database from one storage location to another is by backing up the database and then performing a redirected restore operation to copy the database to a new location. However, because a redirected restore operation allows you to change the storage location for one or more tablespaces but not the tablespace type (SMS or DMS), this process cannot be used for migrating a DB2 database if you wish to change the tablespace type (for example, to switch from DMS tablespaces to SMS tablespaces).

Instead, to perform this type of migration, a new DB2 database that uses the desired tablespace types must be created and data from the existing database must be copied to the new database. Data can be copied on a table-by-table basis using DB2's EXPORT and IMPORT commands, but a more efficient way to copy an entire DB2 database is by using DB2's db2move utility. This utility queries the system catalog tables for the specified database and compiles a list of all user tables found. It then exports the contents and table structure of each table found to a PC/IXF formatted file. The set of files produced can be used to populate a new a DB2 database that uses tablespaces that are associated with files stored on an IBM N series storage system.

The db2move utility can be run in one of three modes: EXPORT, IMPORT, or LOAD. When run in EXPORT mode, the db2move utility invokes DB2's EXPORT utility to extract data from one or more tables and externalize it to PC/IXF formatted files. It also creates a file named db2move.lst that contains the names of all tables processed, along with the names of the files that the table's data was written to. In addition, the db2move utility may produce one or more message

files that contain warning or error messages that were generated as a result of an export operation.

When run in IMPORT mode, the db2move utility invokes DB2's Import utility to recreate a table, and its indexes from data stored in PC/IXF formatted files. When run in this mode, the file db2move.lst is used to establish a link between the PC/IXF formatted files needed and the tables into which data will be imported.

When run in LOAD mode, the db2move utility invokes DB2's LOAD utility to populate tables that already exist with data stored in PC/IXF formatted files. Again, the file db2move.lst is used to establish a link between the PC/IXF formatted files needed and the tables into which data will be loaded. For the purpose of migrating a DB2 database, the db2move utility should never be run in LOAD mode.

Unfortunately, the db2move utility only works with table and index data objects. If the database to be migrated contains other data objects such as aliases, views, triggers, user-defined data types (UDTs), user-defined functions (UDFs), and so on, you must find another way to migrate those objects as well. This is where the DB2 utility db2look comes in. This utility analyzes an existing database and produces a set of Data Definition Language (DDL) SQL statements that can then be used to recreate all of the objects found in the database analyzed.

Now that we have seen how the tools that are needed work, let us take a look at how this type of migration is performed. The steps needed to perform this type of DB2 database migration are:

1. Stop and restart the DB2 Database Manager instance by executing the following commands:

   ```
   $ db2 "FORCE APPLICATIONS ALL"
   $ db2stop
   $ db2start
   ```

2. Generate DDL that can be used to recreate the data objects found in the DB2 database to be migrated by executing the following command:

   ```
   db2look -d  < DB_NAME > -e -o  < OUT_FILE >
   ```

   Where DB_NAME is the alias assigned to the database that is to be migrated and OUT_FILE is the name and location of the file that has the DDL information written to it.

   For example, to create DDL for the object database named mydb, you would execute the following command on the database server:

   ```
   db2look -d mydb -e -o /export/home/db2ins81/mydb.ddl
   ```

   When executed, the command will place the mudb.ddl file in the directory /export/home/db2ins81.

3. Edit the file produced in Step 2 and delete the first CONNECT statement found. Save the file when finished.
4. Use the **db2move** command to query the system catalog tables of the DB2 database to be migrated and export the contents and table structure of each table found to a PC/IXF format:

   ```
   db2move [DB_NAME] EXPORT
   ```

   where DB_NAME is the alias assigned to the database that is to be migrated.

   For example, to migrate a database named mydb, you would execute the following command on the database server:

   ```
   db2move mydb export
   ```
5. Create a new database on the IBM N series storage system volumes, as shown in Chapter 5, “Configuring the environment” on page 19.

   Note: If you want to assign the name of the database being migrated to the database being created, you must first drop or uncatalog the database being migrated. If you select to uncatalog the database being migrated, you can always recatalog it.
6. Establish a connection to the database created in Step 5 and create the database objects using the script created db2look utility in step 2:

   ```
   db2 "connect to mydb1"
   db2 -t -f /export/home/mydb.ddl
   ```
7. Import the contents and table structure of each table found in the DB2 database being migrated to the database created in Step 5 by executing the following command:

   ```
   db2move mydb IMPORT
   ```
8. Verify that the database was successfully migrated by performing the simple tests outlined in 6.4, “Database verification” on page 71.

8

Performance considerations

DB2 UDB provides two sets of configuration parameters (one for the DB2 Database Manager instance and one for the database itself) along with several registry and environment variables that can be used to tune a system for optimum performance. Two of these registry variables, DB2_PARALLEL_IO and DB2_STRIPED_CONTAINERS, should always be set when DB2 is used in conjunction with an IBM N series storage system. The system command **`db2set`** is used to display, set, or remove values for DB2 registry/environment variables; after setting any registry variable, the DB2 Database Manager must be stopped and restarted before the changes will take effect.

8.1 The DB2_PARALLEL_IO registry variable

When reading data from or writing data to tablespace containers, DB2 can use parallel I/O if the number of containers in the tablespace is greater than 1. However, there are situations when it would be beneficial to have parallel I/O enabled for single container tablespaces. For example, if the container is created on an IBM N series storage system volume or qtree (Figure 8-1), performance may be improved if read and write calls are issued in parallel.

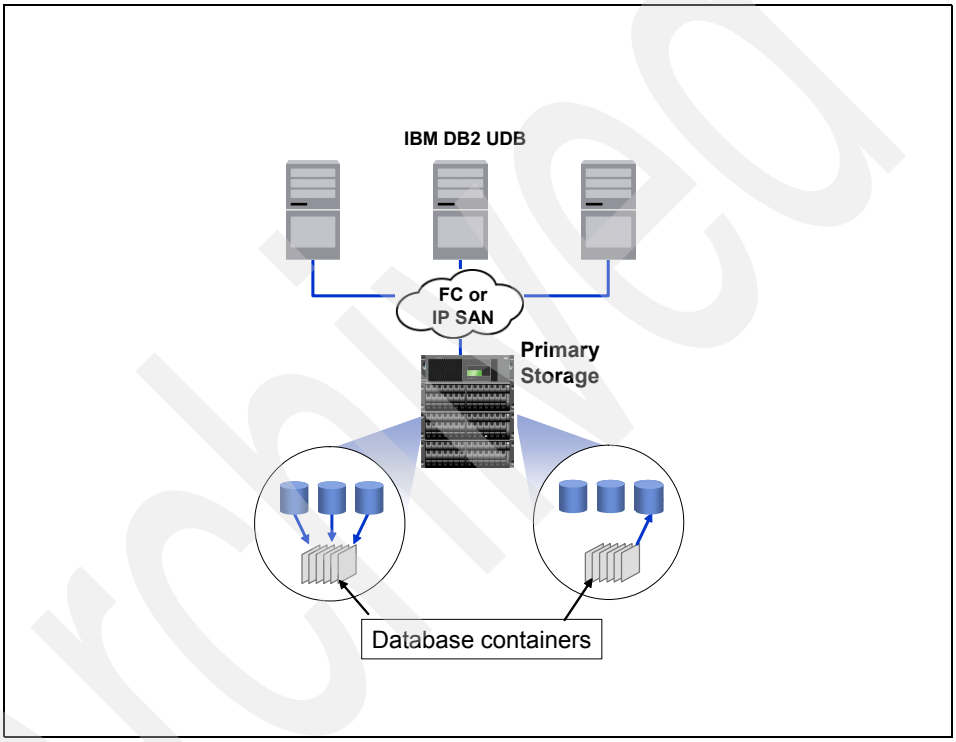


Figure 8-1 Database containers

To force DB2 UDB to use parallel I/O for a tablespace that has only one container, you use the DB2_PARALLEL_IO registry variable. This variable can be set to asterisk (*), meaning every tablespace in every database for the database instance is to use parallel I/O, or it can be set to a list of tablespace IDs that are separated by commas. Execute one of the following commands on your database server:

```
db2set DB2_PARALLEL_IO=*
```

or

```
db2set DB2_PARALLEL_IO=1,2,4,8
```

In the above example, the first command will turn parallel I/O on for all tablespaces, while the second command will turn parallel I/O on only for tablespaces 1, 2, 4, and 8.

8.2 The DB2 Configuration Advisor

When a database is created using the CREATE DATABASE command, it is created with a set of a default database configuration parameter values. These configuration parameter values may not be appropriate for your environment. To obtain a good performance, you may need to tweak these parameter values. If your experience with DB2 UDB is limited, you can use the DB2 Configuration Advisor to tune these configuration parameters.

The Configuration Advisor is a tool that asks you a series of questions about your database environment and, using the answers provided, recommends configuration parameter values that will improve overall database performance. The Configuration Advisor is invoked from the DB2 Control Center by highlighting the database you want to tune, from the list of databases presented, pressing the right mouse button to display the database action menu, and selecting **Configuration Advisor**.

8.3 Additional factors that affect performance

Finally, when running a DB2 database whose data and transaction files reside on an IBM N series storage system, you should take the following into consideration.

In order to achieve point-of-failure recovery, you must ensure that the database's active and archive log files are always accessible and up-to-date. The first step towards doing this is to store a database's transaction log files on a volume that is separate from the volume where the database's object files reside. (In our test environment, the database object files were stored on the IBM N series storage system volume named db2data and the database's transaction log files were stored on the IBM N series storage system volume named db2logs, as shown in Figure 8-2).

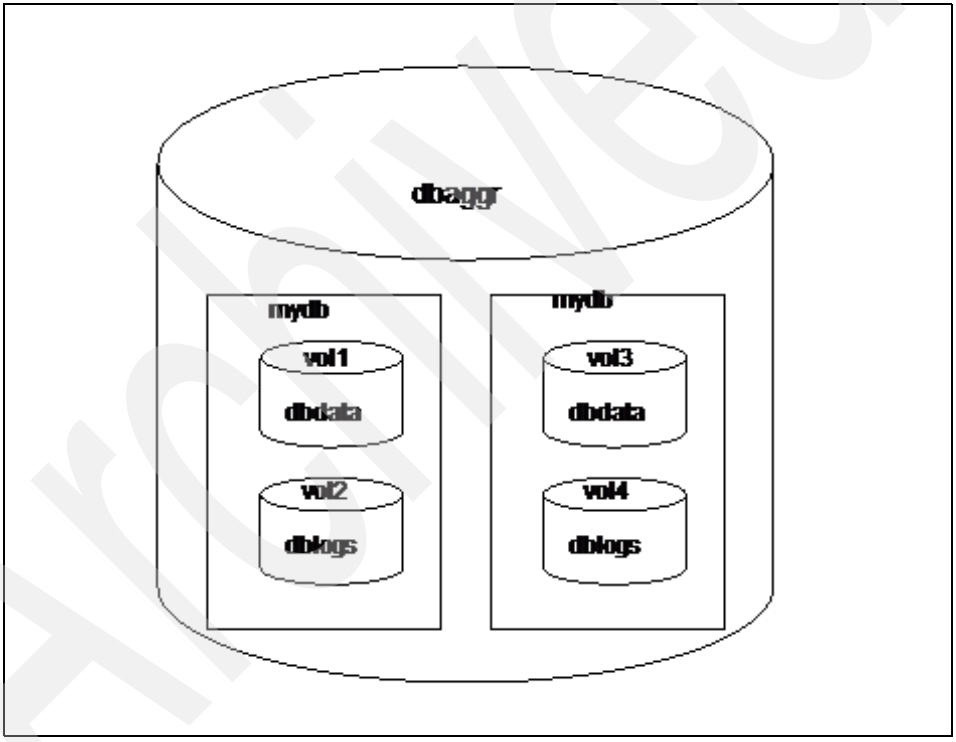


Figure 8-2 Volume placement

Each tablespace should be assigned a prefetch size that is a multiple of the extent size used multiplied by the number of data disks used in the Storage System RAID group the tablespace resides on:

prefetch size = (extent size) x tablespace data disks

If you are using SMS tablespaces, tell DB2 to allocate new SMS space by the extent rather than the page. This is done by enabling multipage file allocation for a database by executing the following DB2 UDB command on the database server:

```
db2empfa [databaseName]
```

For example, to enable multipage file allocation for the database named mydb, you would execute the following command on the database server:

```
db2empfa mydb
```

9

Enabling NVFAIL on the IBM N series

This chapter will discuss enabling NVFAIL on the IBM N series storage system for additional data protection.

When storing database data on an IBM N series storage system, it is a good idea to enable the NVFAIL feature for the volumes used for the database. NVFAIL is a Data ONTAP operating system software feature and provides support for special error processing, and it is enabled by entering the following command on the IBM N series storage system:

```
vol options [VOL_NAME] nvfail on
```

Where VOL_NAME is the name of the IBM N series storage system volume the NVFAIL feature is to be enabled for.

For example, to enable the NVFAIL feature for the volume named db2data, you would execute the following command on the IBM N series storage system:

```
vol options db2data nvfail on
```

The resultant status change would look like Figure 9-1.

```
itsotuc2> vol status db2data
         Volume State      Status            Options
        db2data online     raid_dp, flex     nosnap=on, nvfail=on
                Volume has clones: db2datacl
                Containing aggregate: 'aggr0'
itsotuc2>
```

Figure 9-1 vol status

If an NVRAM failure occurs on a volume, Data ONTAP detects the failure at boot up time. If you enabled the vol options nvfail option for the volume and it contains the LUNs, Data ONTAP performs the following actions:

- Takes the LUNs off line in the volumes that had the NVRAM failure.
- Stops exporting LUNs over FCP.
- Sends error messages to the console stating that Data ONTAP took the LUNs offline or that NFS file handles are stale. (This is also useful if the LUN is accessed over NAS protocols.)

If desired, a second feature that renames certain files that the system administrator or DBA may not want to be made accessible to the network until after they have been carefully examined can be used to provide additional protection. This feature is controlled by the presence or absence of the file /etc/nvfail_rename (stored on the root volume of the IBM N series storage system). The format of the /etc/nvfail_rename file is simply the name of a file found on the IBM N series storage system, one file name per line; if this file exists, each file listed in it is renamed by having the string ".nvfail" appended to it.

For example, if the following files exist on the IBM N series storage system:

db2filer:/vol/db2data/db2inst1/NODE0000/SQL0001/SQLSPCS.1

db2filer:/vol/db2data/db2inst1/NODE0000/SQL0001/SQLSPCS.2

If the file /etc/nvfail_rename exists, it looks something like this:

/vol/db2data/db2inst1/NODE0000/SQL0001/SQLSPCS.1

/vol/db2data/db2inst1/NODE0000/SQL0001/SQLSPCS.2

If status checking determines that an NVRAM failure has occurred, the files listed in the file /etc/nvfail_rename will be renamed to:

db2filer:/vol/db2data/db2inst1/NODE0000/SQL0001/SQLSPCS.1.nvfail

db2filer:/vol/db2data/db2inst1/NODE0000/SQL0001/SQLSPCS.2.nvfail

Since this occurs before the IBM N series storage system begins providing network service, these files will no longer have the same file names they had before the error occurred. As a result, the DB2 Database Manager will not be able to open the files needed (because they no longer exist) and an error will occur. Once the NVRAM problem has been corrected, you can restore use of the Storage System by stopping the DB2 Database Manager, renaming the affected files, and then restarting the DB2 Database Manager.

Note: To ensure that a DBA becomes aware of an NVRAM failure regardless of when any user attempts to access any database stored on an IBM N series storage system, you may want to create a /etc/nvfail_rename file that contains the names of every file on the IBM N series storage system that is being used by a particular DB2 UDB database.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information about ordering these publications, see “How to get IBM Redbooks” on page 92. Note that some of the documents referenced here may be available in softcopy only.

- *IBM N Series Storage Systems in a Microsoft Windows Environment*, REDP-4083
- *IBM Storage System N series Antivirus Scanning Best Practices Guide,* REDP-4084
- *Multiprotocol Data Access with IBM System Storage N series*, REDP-4176
- *N Series SnapManager with Microsoft Exchange*, REDP-4160

Other publications

These publications are also relevant as further information sources:

- *IIBM System Storage N series MultiStore Management Guide,* GA32-0524-01
- *BM System Storage N series Network Management Guide,* GA32-0525-01
- *IBM System Storage N series Software Setup Guide*, GA32-0530-01
- *Quick Beginnings for DB2 Servers*, found at:

  ftp://ftp.software.ibm.com/ps/products/db2/info/vr82/pdf/en_US/db2ise81.pdf

Online resources

These Web sites are also relevant as further information sources:

- Support for System Storage N5200

  http://www-03.ibm.com/servers/storage/support/nas/n5200/installing.html

- Support for System Storage N5500

  http://www-03.ibm.com/servers/storage/support/nas/n5500/

- Support for System Storage N7000

  http://www-03.ibm.com/servers/storage/support/nas/n7000/

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

(0.2”spine)
0.17”<->0.473”
90<->249 pages

Integrating IBM DB2 with the IBM System Storage N series

Preparing IBM N series for IBM UDB DB2 installation

Creating a database on the IBM N series

Migrating a database to the IBM N series

This IBM Redbook describes the steps carried out to configure and install an IBM DB2 database in a N series storage system UNIX environment. It particularly elaborates the configuration steps for the UNIX host as well as the IBM N series storage system. The UNIX host types covered in this IBM Redbook are Linux, IBM AIX, Sun Solaris, and HP-UX.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-7329-00 ISBN 0738496855