



DB2 9 for Windows: Integrating with a NetApp or IBM N Series Storage System

Jawahar Lal and Roger Sanders, NetApp, Inc.

August 01, 2006 | TR 3489

Executive Summary

DB2 9 is validated and certified by IBM to run on a NetApp or IBM N Series storage system. This document describes the steps needed to configure and install a DB2 9 database in a Windows® environment on a NetApp or IBM N Series

storage system. Specifically, it describes the configuration steps needed for both the Windows host and the NetApp or IBM N Series storage system.

Table of Contents

1. Purpose and Scope	4
2. Overview of SnapDrive for Windows	4
2.1. Introduction to SnapDrive for Windows	4
2.2. Benefits of using SnapDrive.....	4
2.3. SnapDrive common terms and technologies.....	4
2.4. SnapDrive recommendations.....	5
3. Assumptions	6
4. Infrastructure	6
4.1. DB2 9	7
4.2. Database Server with DB2 administrator and user accounts	7
4.3. NetApp storage system	7
4.4. SnapDrive for Windows	7
4.5. Microsoft iSCSI software initiator	7
4.6. Network	7
5. Database Considerations	8
5.1. Database storage considerations.....	8
5.2. Designing a database with recovery in mind.....	8
5.3. Aggregate and Volume Configuration	9
6. Advantages of storing DB2 databases on a NetApp FAS or IBM N Series storage system	9
7. Configuring the database server and the storage system	10
7.1. Configuring the storage system	11
7.2. Configure authentication.....	15
7.3. Configuring a Database Server.....	16
8. Creating a New Database on a NetApp storage system	23
9. Migrating an existing DB2 database to a NetApp storage system	25
9.1. Migrating a DB2 Database Using a Redirected Restore Operation	26
9.2. Migrating a DB2 Database Using db2look and db2move	27
10. Performance Considerations	29
10.1. The DB2_PARALLEL_IO Registry Variable	29
10.2. The DB2 Configuration Advisor.....	30
10.3. Additional factors that affect performance.....	30
11. Enabling NVFAIL on the storage system for additional data protection	30
12. Caveats	31
13. Important links	31

1. Purpose and Scope

This document describes the steps necessary to integrate DB2 9 for Windows with a NetApp™ or IBM N Series storage system. Specifically, the following topics are covered:

- Preparing the Windows server for DB2 installation
- Preparing the storage system for DB2 database storage
- Installing DB2 on a Windows server and creating a database on a storage system
- Storing database transaction log files on a storage system
- Migrating an existing database from local disks to a storage system

2. Overview of SnapDrive for Windows

2.1. Introduction to SnapDrive for Windows

SnapDrive® for Windows is an enterprise-class storage and data management solution for the Microsoft® Windows 2000 and 2003 servers. It simplifies storage management and increases availability and reliability of the application data. It also provides storage virtualization of volumes on the storage system using FC or iSCSI protocols and presents LUNs on volumes as locally attached disks to the Windows server.

With SnapDrive for Windows, server and storage administrators can use a wizard-based approach to map, manage, and migrate data between new and existing storage resources from a host-consistent perspective. SnapDrive for Windows eliminates the need to maintain manual scripts normally used to back up and restore data to specific drives being used by various downstream applications and databases without extensive downtime. It can also be used to add additional storage as needed, eliminating the need to pre-allocate large amounts of storage resources based only on forecasted demand.

2.2. Benefits of using SnapDrive

SnapDrive for Windows provides a layer of abstraction between a Windows server and a storage system. Applications running on a Windows server with SnapDrive can access virtual disks on the storage system as if they were locally attached disks. Some of the key benefits of using SnapDrive on a Windows server are:

- Simplified storage management
- Error-free application-storage provisioning
- Consistent data Snapshot copies
- Easy backup based on Snapshot
- Rapid application recovery
- Complements the native NTFS file system as it runs on the Windows host
- Integrates seamlessly with Microsoft Cluster Server (MSCS)

2.3. SnapDrive common terms and technologies

This section describes the most common terms used for a SnapDrive for Windows environment.

- **Host/Database server:**

A host or database server is a computer system that accesses the storage on a storage system.

For this document's purpose, the host should run at least the following software:

- Windows 2000/2003 server with necessary service packs.
- SnapDrive 4.1 for Windows.
- Microsoft iSCSI software initiator.

- **Host Bus Adapter (HBA)**

A host bus adapter is the adapter used to connect hosts and storage systems in a storage area network (SAN) so that hosts can access logical unit numbers (LUNs) on the storage systems using FCP or iSCSI protocol.

- **Network Interface Card (NIC)**

A network interface card is a Gigabit Ethernet (GbE) or a Fast Ethernet card that connects hosts and storage systems to a local area network (LAN) or to data-center switching fabrics, specifically, to enable hosts to connect to LUNs on storage systems using iSCSI.

- **Initiator**

An initiator is used to send SCSI I/O commands to a target. An FCP initiator is a port on an HBA on a host and an iSCSI initiator is a port on a NIC on a host. NetApp supports both software and hardware initiators. For details, refer to “The Compatibility and Configuration Guide for NetApp FCP and iSCSI Products” on the [NOW™](#) Web site.

- **Target**

In the SAN or IP SAN environment, targets receive the SCSI I/O command sent by initiators. In a NetApp FAS or IBM N Series SAN environment, the host works as the initiator and the storage system as the target.

- **LUN**

A logical unit number (LUN) is a SCSI identifier of a logical unit of storage on a target. This document often refers to LUNs as “virtual disks,” and vice versa.

- **Virtual disk**

A virtual disk is a functional unit of storage system storage that, for all practical purposes, behaves like a locally attached disk on a Windows host. This document often refers to virtual disks as “logical unit numbers” (LUNs), and vice versa.

- **iSCSI**

The iSCSI is a standards-based transport protocol that is used to connect storage devices over a TCP/IP network. To the operating system or application that uses the storage, an iSCSI device appears to be a local disk drive. The iSCSI protocol defines the way that standard SCSI commands and responses are encapsulated into TCP/IP packets and transmitted over an Ethernet network.

The user of iSCSI storage is an initiator. The provider of iSCSI storage is a target. For NetApp FAS or IBM N Series storage systems, each host has one or more initiators, and the storage system is the target. The set of initiators, targets, and the connecting network makes up an iSCSI storage area network (SAN).

- **File system**

File system refers to NTFS, the native Windows 2000 Server and Windows Server 2003 file system supported by SnapDrive. (NetApp FAS or IBM N Series storage systems use the Write Anywhere File Layout [WAFL®] file system internally, but SnapDrive makes WAFL transparent to virtual disk users who interact with data stored on the storage system using Windows procedures only.)

2.4. SnapDrive recommendations

While using SnapDrive, you should pay attention to the following recommendations:

- Place the host and storage system in the same broadcast domain, so that virtual disk I/O commands do not need to make router hops.
- For Windows cluster configurations, segregate internal cluster traffic from both host-storage system traffic and data-center traffic whenever possible.
- Do not permit internal cluster traffic on a Gigabit Ethernet (GbE) network used for transferring data from host to storage system. Instead, use a Fast Ethernet connection for all cluster traffic. This

ensures that a single network error cannot affect both the connection for internal cluster traffic and the connection to the quorum disk.

- Don't create or delete LUNs using FilerView® or the filer command line commands. Use SnapDrive to create and manage all the virtual disks that are used for the database.
- The root volume (/vol/vol0) is used by Data ONTAP® to administer the storage system. Therefore, do not create any LUNs or qtrees in the root volume.
- Never disable the space reservation setting for any virtual disk managed by SnapDrive.
- Do set the snap reserve setting on the storage system to 0%.
- Place all virtual disks connected to the same host on a dedicated volume accessible by just that host.
- Unless you can be sure that name resolution publishes only the storage system interface you intend, configure each network interface by IP address, rather than by name.
- If you use Snapshot copies, you cannot use the entire space on a storage system volume to store your virtual disk. The storage system volume hosting the virtual disk should be at least twice the combined size of all the virtual disks on the volume.

3. Assumptions

This document describes the deployment of IBM DB2 9 into an already functioning MS Windows and NetApp FAS or IBM N Series storage system environment. Therefore, it is assumed that you are familiar with DB2 9 and the operation of NetApp or IBM N Series storage systems. It is also assumed that you possess basic Windows server administration skills. The examples provided may require some modifications before they can be run under your version of Windows.

Throughout this document, the phrase “storage system” is used as a synonym for a NetApp FAS or IBM N Series storage system. Both NetApp FAS and IBM N Series run the same operating system named Data ONTAP. The steps described in this document are applicable to both NetApp FAS and IBM N Series storage systems.

It is recommended that you read DB2 installation documentation and follow its steps to install and configure the DB2 instance and database. If there are contradictions between the DB2 documentation and this technical report you should assume that the DB2 documentation is correct. Please inform NetApp of any such contradictions found so this document can be corrected.

The sample scripts in this technical report assume the following:

- The name of the storage system is **netapp**
- The name of the Windows server, henceforth known as the database server, is **db2srv1**
- The user name for the DB2 instance owner is **db2inst1**
- The DB2 instance on the Windows server is **db2inst1**
- The home directory for the DB2 instance is **C:\IBM\DB2V9\SQLLIB**
- The names of the FlexVol™ volumes on the storage system are **db2data** and **db2logs**
- Data files for the database are stored on a virtual disk **F:** which is a LUN on the FlexVol volume **db2data**
- Transaction log files for the database are stored on a virtual disk **G:** which is a LUN on the FlexVol volume **db2logs**

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

4. Infrastructure

In order to use DB2 9 for Windows in conjunction with a NetApp or IBM N Series storage system, the following elements are required:

- DB2 9 software
- One or more Windows servers (database servers) with DB2 Administrator and user accounts
- One or more NetApp FAS or IBM N Series running Data ONTAP 7G (or later)

- Network connectivity between the database server and the storage system
- FC HBA or iSCSI software initiator
- SnapDrive 4.1 for Windows installed on the database server

4.1. DB2 9

In order to perform the steps outlined in this document, you need to have DB2 9 installed and running on a Windows server or workstation. Before proceeding with installation of DB2, check the system requirements described in the DB2 installation documentation.

4.2. Database Server with DB2 administrator and user accounts

You will need to have both a DB2 administrator and one or more DB2 user accounts on the Windows server to install and perform database operations. If you already have DB2 installed on your Windows server, chances are that the appropriate DB2 user accounts already exist. If, however, you are installing DB2 for the first time, a DB2 administrator and one or more user accounts can be created as part of the DB2 installation process. Refer to the DB2 documentation for detailed instructions on setting up the appropriate accounts during the installation process. Database server configuration is also covered in a later section of this document.

4.3. NetApp storage system

DB2 stores data in tablespace containers. A tablespace container, which can be a file or a raw device, can physically reside on a storage system. Any NetApp FAS or IBM N Series storage system running a supported version of Data ONTAP can be used to store a DB2 database and its transaction log files. In order to produce this document we used a NetApp FAS3000 series storage system running Data ONTAP 7.1.

4.4. SnapDrive for Windows

To simplify the storage management for DB2 database and manage database data we used SnapDrive for Windows version 4.1. SnapDrive for Windows is a licensed product and can be downloaded from the NetApp Web site (<http://now.netapp.com>) or IBM Web site (<http://www-03.ibm.com/servers/storage/network/software/>).

4.5. Microsoft iSCSI software initiator

For MS Windows 2000 or 2003 servers, NetApp supports both software and hardware initiators when using the iSCSI protocol. For details on the supported initiators, refer to the “*NetApp iSCSI Solutions Support Matrixes*” page on the [NOW](#) Web site.

4.6. Network

In order to store a DB2 database on a NetApp FAS or IBM N Series storage system you must first establish a network connection between the DB2 database server and the storage system. The type of connection you establish is dependent upon the type of environment you are using. If your environment is SAN with Fibre Channel (FC) then you will need to establish an FC connection between the database server and the storage system; otherwise, a Gigabit Ethernet (GbE) connection will work for the IP SAN. For simplicity, we used a single Windows 2003 host attached to a single storage system through a single Gigabit Ethernet (GbE) switch to develop this document. Figure 1 illustrates the high level network architecture that is used to integrate a Windows host with a NetApp storage system.

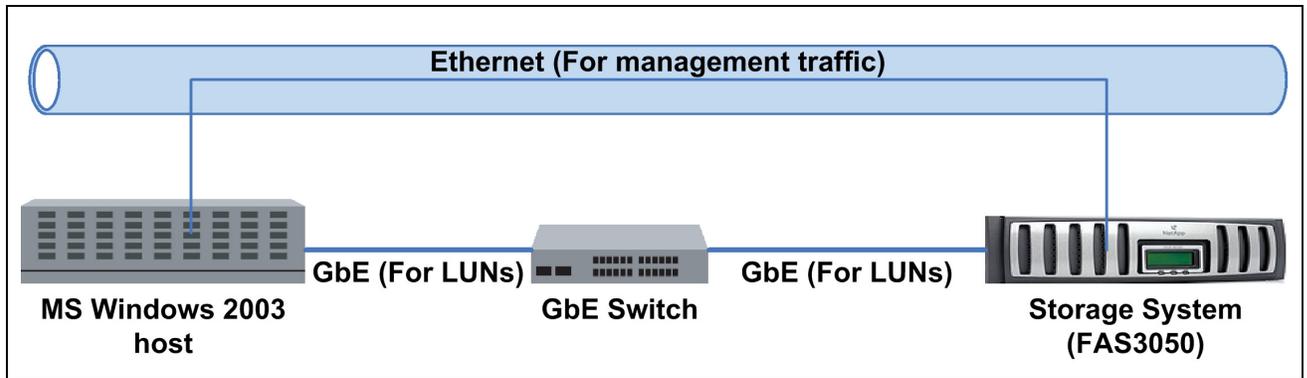


Figure 1) Network architecture for a Windows and NetApp storage system environment.

For details on other iSCSI or FC configurations, refer to “*SnapDrive 4.1 for Windows Installation and Administration Guide.*”

For optimal performance, it is **strongly** recommended that you use Gigabit Ethernet interfaces to connect your database server to the storage system. You should also heed the following performance recommendations:

- Flow control and full-duplex operation should be enabled on the Ethernet interfaces on *both* the Windows server and the storage system.
- When using Gigabit Ethernet adapters, consider turning on jumbo frames.
- If possible, dedicate at least one private network for the connection between the database server and the storage system. This can be done using a switch or a crossover cable on most networking topologies (we used a Gigabit Ethernet switch to develop this document).
- To increase the occurrence of parallel I/O between the database server and the storage system, create separate virtual disks, using Snapdrive, on the storage system for database data and transaction logs.
- Dedicated or private network connections between the database server and the storage system are recommended for the following reasons:
 - Any issues of contention or latency are eliminated if the database server and the 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 NetApp/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 bandwidth between the storage system and the database server.

5. Database Considerations

5.1. Database storage considerations

Traditional installation of DB2 on a Windows platform calls for the storage of database data and corresponding transaction log files on local disk(s). SnapDrive provides storage virtualization and presents LUNs on a storage system as a locally attached disk to a Windows server. By storing a database’s overhead files, data, and transaction log files on virtual disks that are created using SnapDrive, a database administrator can take full advantage of NetApp technologies such as Snapshot, SnapMirror®, RAID-DP™, and FlexClone™.

5.2. 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 the 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 and put on the same virtual disk that resides on a volume on the storage system used. 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. In addition, it goes without saying that the archive logs should be stored on a separate virtual disk 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.

5.3. Aggregate and Volume Configuration

The basic data container in DB2 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 containers used.

Data ONTAP 7G supports a new virtual storage layer called flexible volumes, hereafter known as a FlexVol 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. For more information on creating and managing FlexVol volumes, refer to the vol create command in the manual “*Data ONTAP 7.1 Commands: Manual Page Reference, Volume 1*” on the [NOW](#) Web site.

A key to ease of use and manageability of a DB2 9 database stored on a NetApp FAS or 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 storage system to manage the physical data storage and ensure high performance and availability. However, there are a few 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.
- It is strongly recommended that all the database object files be stored on a volume on the storage system that is not the root volume.
- Create volumes for the exclusive use of the LUNs; no other files or directories should exist in the volumes that contain LUNs.
- Disable scheduled auto Snapshot copies for the volumes that are used for the database.
- It is strongly recommended 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 also 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 the active and archive logs to be overwritten, resulting in a loss of all 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.
- It is strongly recommended 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.

NOTE: Throughout this document, the terms volume and FlexVol volume are interchangeable unless otherwise mentioned.

6. Advantages of storing DB2 databases on a NetApp FAS or IBM N Series storage system

Running a DB2 database with data and transaction log files stored on a NetApp FAS or IBM N Series storage system has several advantages:

- **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 storage system used. This reduces the database backup window from hours to seconds and allows DBAs to take frequent full database backups without having to take the database offline.
- **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 needing to restore from tape.
- **High Availability:** The need for 24x7 availability is fast becoming a reality for organizations of all sizes. Organizations cannot tolerate scheduled downtime nor can they afford extended periods of slow system response caused by traditional database backup methods. Database backups using NetApp Snapshot technology can be created in a matter of seconds without bringing a database down.
- **High Reliability:** The RAID architecture used for NetApp FAS or IBM N Series storage systems is unique and provides greater reliability than many traditional RAID implementations. If a disk in a RAID group fails, it is reconstructed automatically without any user intervention. Additionally, the support for new architecture known as RAID-DP makes the NetApp FAS or IBM N Series storage systems highly reliable. RAID-DP is considered approximately 4,000 times more reliable than traditional RAID. For more details on RAID-DP please read [NetApp Data Protection: Double Parity RAID for Enhanced Data Protection with RAID-DP](#).
- **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 doesn't involve actual data I/O, the process of backing up a database has virtually no performance impact on system response time.
- **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.
- **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 Data ONTAP operating system, only one container needs to be defined for each tablespace used.

7. Configuring the database server and the storage system

This section describes the steps involved in installing the DB2 software on a database server and creating a DB2 database on a storage system. (*Note:* The items described in section [4. Infrastructure](#) must exist in order for this information to be valid.) Before you begin the installation process, you must ensure that the appropriate service packs have been applied and all environment parameters have been updated as per the installation documentation. Additional information on the DB2 installation procedure can be found at the DB2 Web site in the manual "*Quick Beginnings for DB2 Servers.*"

Before you create a database on a NetApp FAS or IBM N Series storage system, you need to complete a series of steps to configure your environment. These steps can be broken down into three sets: one set of steps must be performed on the storage system while the other needs to be performed on the database server. The third set, "configuring authentication," needs to be performed on both database server and storage system.

7.1. Configuring the storage system

(1). Set up and install Data ONTAP 7.0

Install Data ONTAP 7G on your storage system if it is not already installed. For Data ONTAP installation and setup instructions please refer to the “Software Setup Guide” on the NOW Web site.

(2). Activate license and start the appropriate services

On the NetApp FAS or IBM N Series storage system, file access protocols FCP, iSCSI, NFS, and CIFS are licensed services. You need to enable the following services by activating license keys for the protocol you intend to use.

- iSCSI, if you plan to use iSCSI-accessed virtual disks
- FCP, if you plan to use FCP-accessed virtual disks
- SnapRestore, which is required only for restoring virtual disks from Snapshot copies
- SnapMirror, if you plan to use the SnapMirror option

The license keys can be activated by executing the following command on the NetApp storage system:

```
license add [LicenseCode]
```

where

- *LicenseCode* identifies the necessary license key for the service/product.

NOTE: Parameters or options shown in square brackets ([]) are required and must be provided; parameters shown in angle brackets (< >) are optional; a comma followed by ellipses (...) indicates that the preceding parameter can be repeated multiple times.

For example, to activate the license code for iSCSI, you would execute the following command on the storage system:

```
license add 123XYZABCD
```

After activating the license key you need to start the service by executing the following command:

```
[fcp | iscsi ] start
```

(3). Update the /etc/host file on the storage system

The storage system must be able to communicate with the database server and vice versa. The storage system can communicate with 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 like 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. Entries made in the `/etc/hosts` file should have the following format:

```
[DBServerIP] [DBServerName]
```

where

- *DBServerIP* identifies the IP address assigned to the database server.
- *DBServerName* identifies the name assigned to the database server.

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

```
172.17.32.112      db2srvr1
```

(4). Enable rsh access

In order to use `rsh` (remote shell) from a database server, you must enable `rsh` by turning the `rsh` option on and editing the `/etc/hosts.equiv` files. You can turn the `rsh` option on by executing the following command on the storage system:

```
options rsh.enable on
```

Edit the `/etc/hosts.equiv` file and add an entry for each user on the database server that can issue the `rsh` command. Each entry should have the following format:

```
[DBServerName] [UserName]
```

where

- **DBServerName** identifies the name assigned to the database server.
- **UserName** identifies the name assigned to the user on the database server.

For example, to enable `rsh` for a user named `db2inst1` on a database server named `db2srv1`, you would add the following entry to the `/etc/hosts.equiv` file on the storage system:

```
db2srv1 db2inst1
```

If a user is a domain user then the entry should follow a format that looks similar to the following:

```
[DBServerName] [DomainName]\[UserName]
```

where

- **DBServerName** identifies the name assigned to the database server.
- **DomainName** identifies the domain name the user belongs to.
- **UserName** identifies the name assigned to the user on the database server.

For example, to enable `rsh` from a database server named `db2srv1` for a user named `db2inst1`, who belongs to a domain named `corp`, you would add the following entry to the `/etc/hosts.equiv` file:

```
db2srv1 corp\db2inst1
```

(5). Create an aggregate and volumes to be used for the virtual disks

In order to create a database on the storage system, you need to create aggregates, flexible volumes, and, in some cases, LUNs. An aggregate is a physical pool of storage, comprised of one or more RAID groups, and is created using the following command:

```
aggr create [AggrName] -r [RaidSize] [NDisks]@[DiskSize] -t [raid-dp/raid4]
```

where

- **AggrName** identifies the name assigned to the aggregate.
- **RaidSize** identifies the raid group size for the aggregate.
- **NDisks** identifies the number of disks used for the aggregate.
- **DiskSize** identifies the disk size for the disks that are used for the aggregate.

For example, to create an aggregate named `db2aggr1` that has sixteen 72 Gig disks in it and that uses RAID-DP, you would execute the following command on the storage system:

```
aggr create db2aggr1 -r 8 -t raid_dp 16@72g
```

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

```
vol create [VolName] [AggrName] [VolSize]
```

where

- *volname* identifies the name assigned to the new volume.
- *AggrName* identifies the name assigned to the aggregate.
- *VolSize* identifies the size of the new volume in kilobyte (KB), megabyte (MB), gigabyte (GB), or terabyte (TB).

In order to create a virtual disk using SnapDrive, you need to create dedicated FlexVol volumes.

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

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

You should create at least two dedicated volumes, one for the database's data and one for the database's transaction logs.

Note: The volume must be more than twice the combined size of all the virtual disks on the volume if a Snapshot copy of the volume will be created. This enables the volume to hold the virtual disks and a special reserved space, so that no matter how much the contents of the virtual disks change between Snapshot copies, the entire contents of the disks can be written to the volume.

(6). Specify the volume security settings to use

You must ensure that the volume(s) on the storage system that are to be used to house a DB2 database support either the **NTFS** or the **Mixed** security style. The security setting of a volume can be set using the following command:

```
qtree security [VolName] [ Unix | NTFS | mixed ]
```

where

- *VolName* identifies the name assigned to the new volume.

For example, to change the security style of a volume named `db2data` to **NTFS**, you would execute the following command on the storage system:

```
qtree security /vol/db2data NTFS
```

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

(7). Create CIFS shares for the volumes.

Next, you need to create CIFS shares for each volume that you created in the previous step. A CIFS share can be created by executing the following command on the storage system:

```
cifs shares -add [ShareName] [VolPath] <-comment [ShareDescription]>
```

where

- *ShareName* identifies the name assigned to the CIFS share created for the dedicated volume.
- *VolPath* identifies the path assigned to the volume.
- *ShareDescription* identifies the description for the CIFS share.

For example, to create a share named `db2data` for a FlexVol volume named `db2data`, you would execute the following command on the storage system:

```
cifs shares -add db2data /vol/db2data -comment "DB2 data storage"
```

Repeat this step and create CIFS shares for all the volumes to be used by the database.

(8). Disable the automatic Snapshot feature

Normally, a database is backed up based on a user-defined schedule. Therefore, it is recommended that you turn off the automatic Snapshot feature for all volumes that are used for the database and its transaction log files. The automatic Snapshot feature can be turned off by executing the following command on the storage system:

```
vol options [VolName] nosnap on
```

where

- **VolName** identifies the name assigned to the new volume.

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

```
vol options db2data nosnap on
```

Repeat this step and turn the auto Snapshot off for all the volumes that are used by the database.

(9). **Setting the snap reserve option for the volume**

By default, Data ONTAP sets the snap reserve options for new volumes to 20%. You should reset the snap reserve option to 0% on all volumes holding SnapDrive virtual disks. To reset the snap reserve option, you would execute the following command on the storage system:

```
snap reserve -V [VolName] [Percent]
```

where

- **VolName** identifies the name assigned to the new volume.
- **Percent** identifies the volume space reserved for the Snapshot copies. A valid value is between 0 and 100.

For example, to set snap reserve to 0 for a flexible volume name `db2data`, you would execute the following command on the storage system:

```
snap reserve -V db2data 0
```

(10). **Obtain a node name for the storage system**

Target NetApp/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 find out the node name assigned to a NetApp/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 NetApp storage system:

```
iscsi nodename
```

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

```
# iscsi nodename
iSCSI target nodename: iqn.1992-08.com.netapp:sn.101171898
```

Make a note of the node name returned by this command; you will need it later when configuring the HBA on the database server.

NOTE: Most of the tasks covered in this document use the command line that is provided through a telnet session to the storage system. These tasks can also be completed using FilerView, which is a Web-based interface that can be used to manage the storage system.

7.2. Configure authentication

Your windows environment for DB2 can use one of the following authentication methods:

- 1) Domain Name Service (DNS) – a domain controller is required
- 2) Network Information Server (NIS) – an NIS server is required
- 3) Pass-through authentication – no domain controller or NIS is required

A pass-through authentication between a Windows host and a storage system can be used for any of the following reasons:

- i. A domain controller is not defined or available.
- ii. You want to install your Windows host as a stand-alone server in a workgroup environment without any dependency on another system for authentication, even if there is a domain controller available.
- iii. Windows host and the storage system are in two different domains.
- iv. Windows host is in a domain and you want to keep the storage system in a workgroup with no direct access by domain users or a domain controller.

You can use any of the authentication methods for your database environment. For the purpose of producing this document, we used pass-through authentication. In order to configure pass-through authentication, you must complete the following steps:

1) Steps performed on the storage system

- a). Create a user that belongs to the local administrator group by entering the following command on the storage system:

```
useradmin useradd [UserName] -g [AdminGroup]
```

where

- **UserName** identifies the name assigned to the user.
- **AdminGroup** identifies the name assigned to the local administrator group.

For example, to create a user named `db2inst1` that belongs to the local administrator group named `administrator`, you would execute the following command on the storage system:

```
useradmin useradd db2inst1 -g administrator
```

- b). Enter a password when prompted to do so for the user account you are creating. (You will be prompted twice to enter a password.)

2) Steps performed on the Windows server

- a). Create a local user account that has administrator privileges. Make sure that the user name and password for the local user account are the same as those you created on the storage system in step 1 of section 7.2.
- b). Open a Windows command prompt window and enter the following command:

```
rsh [StorageSystemName] version
```

where

- **StorageSystemName** identifies the name assigned to the storage system.

For example, to check if `rsh` is enabled for a storage system named `netapp`, you would execute the following command in the command window on the database server:

```
rsh netapp version
```

7.3. Configuring a Database Server

After configuring the storage system, the database server must be configured so that SnapDrive can make use of the volumes that are created by following the steps described in section 7.1. Following are the basic steps that need to be completed to configure the database server:

- 1) Log in to the database server as a user with `Administrator` privileges and install all service packs and hot-fixes required for SnapDrive. For details on the necessary service packs, refer to the “*SnapDrive 4.1 for Windows Installation and Administration Guide*,” which can be obtained from the [NOW](#) Web site.
- 2) If you are installing on Windows NT®, Windows 2000, Windows XP, or Windows Server 2003, you require the following DB2 server user accounts:
 - An installation user account and
 - One or more setup user accounts
 - A DB2 Administration Server (DAS) user account
 - A DB2 instance owner account

The installation user account must be defined prior to running the DB2 Setup wizard and must have domain administrator or local administrator privileges. The setup user accounts can be defined prior to installation or you can have the DB2 Setup program create them for you. All user account names must adhere to your system naming rules and to DB2 naming rules.

- 3) Next, you need to download and install the Microsoft iSCSI software initiator by completing the following steps:
 - a). Download the Microsoft iSCSI software initiator from the Microsoft Web site (<http://www.microsoft.com/downloads/>) and install the initiator using the downloaded file.
 - b). After successful installation, reboot the system.
 - c). Next, configure the initiator. To do this, open the initiator property panel by navigating to **Start -> Programs -> Microsoft iSCSI initiator -> Microsoft iSCSI Initiator** or by double-clicking on the icon named **Microsoft iSCSI initiator** on the desktop.
 - d). Select the **General** tab and check the initiator name. The initiator’s name must adhere to the following naming format:

`iqn. [yyyy-mm] . [BackwardNamingAuthority] : UniqueDeviceName`

where

- `yyyy-mm` is the month and year in which the naming authority acquired the domain name.
- `BackwardNamingAuthority` identifies the reserved domain name (backward) of the authority responsible for naming this device.
- `UniqueDeviceName` identifies the unique name assigned to the host system.

For example, the initiator name should look similar to the one shown in Figure 2.

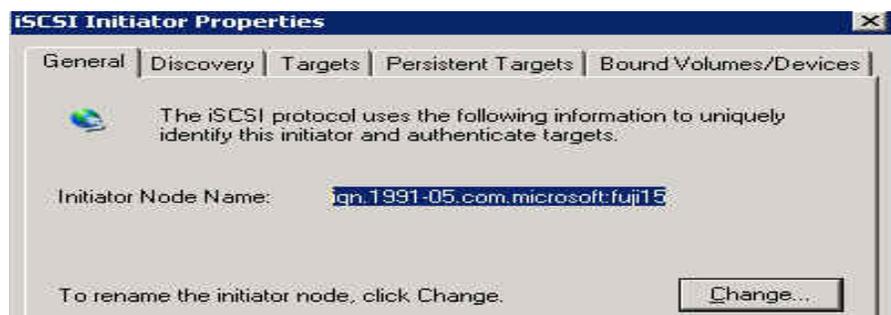


Figure 2) iSCSI software initiator name (`iqn.1991-05.com.microsoft:fuji15`).

If the name is not in the required format, change it by clicking on the **Change** button and providing an appropriate name.

NOTE: If SnapDrive is not used to create virtual disks, then the initiator name is required to create an appropriate igroup on the storage system.

- e). Next, select the **Discovery** tab and add a target by clicking on the **Add** button. Enter the storage system's IP address or DNS name as illustrated in figure 3. The default port value for the storage system is 3260 – if the storage system port value in your environment is not 3260 then provide the appropriate port value.

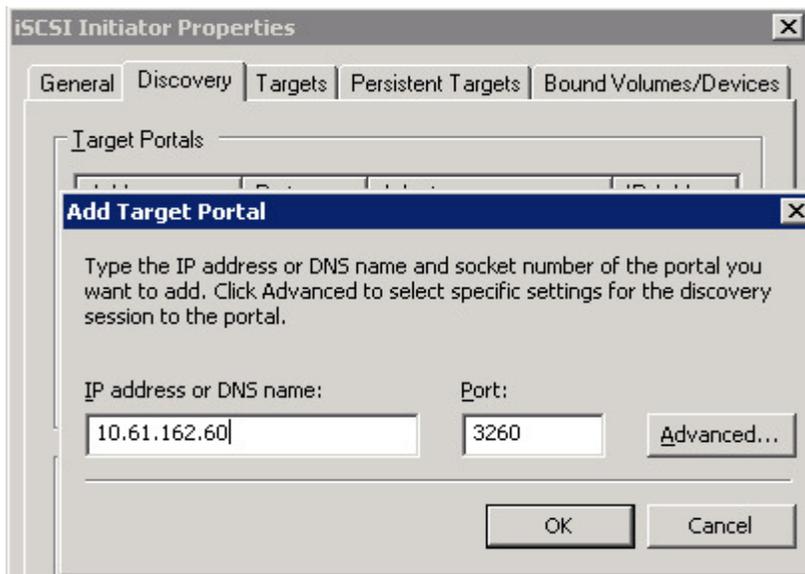


Figure 3) Add a target storage system

After completing above steps, click on the targets tab and check the target name. This name should be the same as you obtained in step 10 of section 7.1. Figure 4 illustrates two targets available for the the initiator.

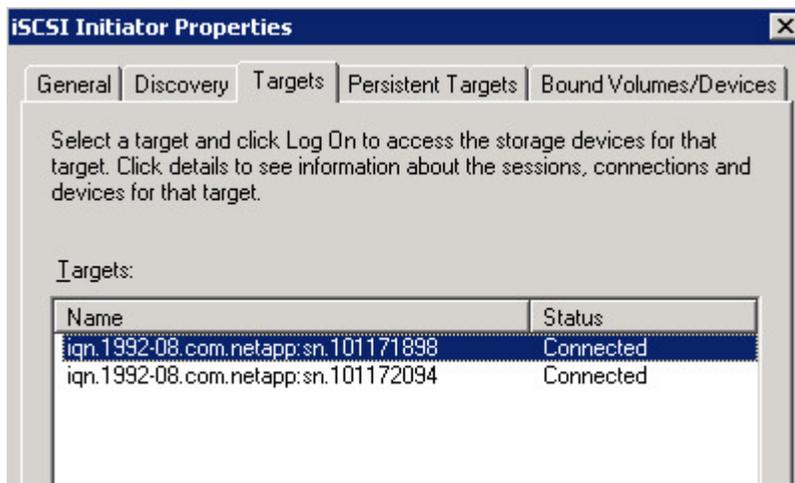


Figure 4) List of the targets the initiator is used for.

After completing the above mentioned steps the initiator is ready to be used.

NOTE: In order to develop this document we used the Microsoft iSCSI software initiator in our environment, though the process steps described here are also applicable for the Fibre Channel protocol.

4) Log in to the [NOW](#) Web site and download SnapDrive 4.1 for Windows. Download the “*SnapDrive 4.1 for Windows Installation and Administration Guide*” along with the software. Before you start the installation process, you should read the installation guide and you must obtain the following license keys:

- LUN Provisioning and Snapshot Management SnapDrive module
- NetApp MPIO with Path Management SnapDrive module

If you are not using MPIO, LUN provisioning and the Snapshot management module should be sufficient for most of the SnapDrive functionality needed.

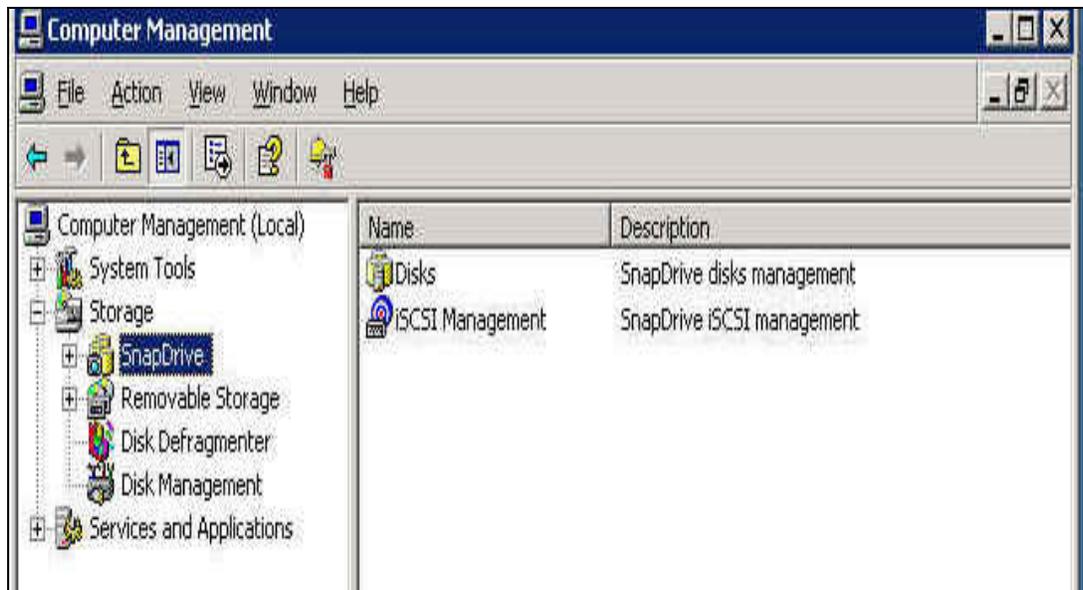
5) Log in as the user who will own the DB2 instance and install the SnapDrive product. For further information, refer to “*SnapDrive 4.1 for Windows Installation and Administration Guide*.”

NOTE: The iSCSI and FCP licenses supplied with SnapDrive enable all the CIFS functionality necessary for using SnapDrive. If you also want full-featured, direct CIFS access to a particular storage system, you must activate a separate CIFS license on that storage system.

6) Create virtual disks

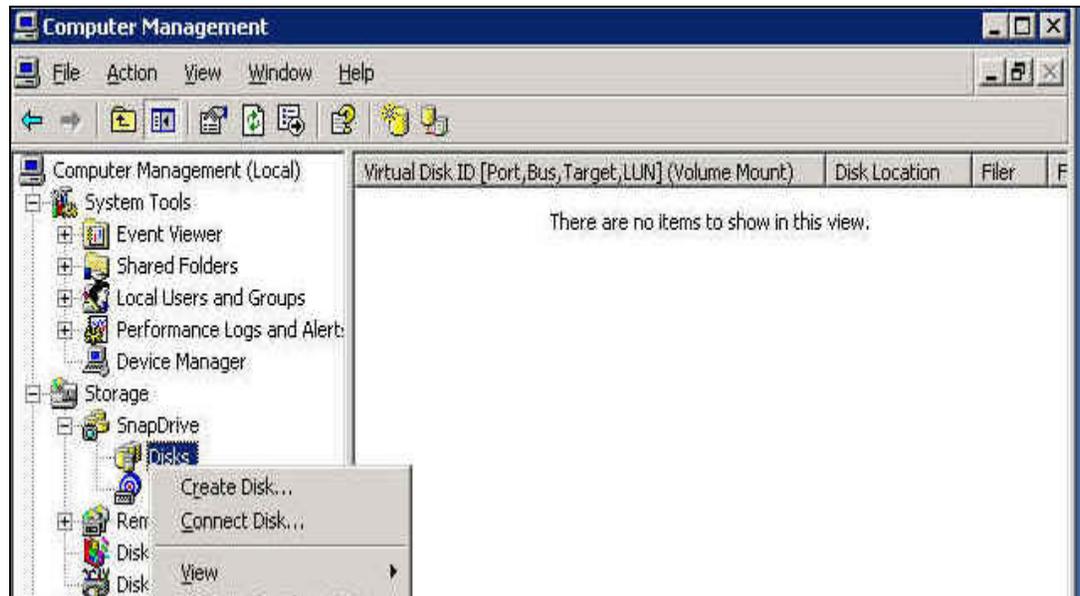
As mentioned earlier you should not create LUNs using FilerView or the command line; use SnapDrive to create virtual disks on the Windows server. In order to create virtual disks, complete the following steps:

a). On the **Start Menu**, click **Run**, type `compmgmt.msc` and then click **OK**. Microsoft Computer Management wizard opens.



Screen 1: Computer management wizard.

- b). Navigate the **Storage -> SnapDrive -> Disks** tree and right-click on the **Disks** node.



Screen 2: Creating virtual disk.

- c). Select **Create Disk** from the menu. This will activate the Create Disk Wizard. When it is displayed, click the **Next** button.



Screen 3: Create disk wizard.

- d). When the next screen is displayed, enter the path and CIFS share name that you created for the volumes on the storage system and click the **Next** button.

Create Disk

Provide a Path and Name
Enter a UNC path to the filer network share and a name for the new virtual disk.

Provide a UNC path to the filer network share for the virtual disk you want to create. The share must be at the root of the filer volume or qtree.

Enter a virtual disk UNC path to filer volume or qtree:

\\10.61.160.29\db2data

Browse...

Enter a name for the new virtual disk:

db2 database data storage container

< Back Next > Cancel

Screen 4: Create disk wizard.

- e). NetApp recommends using dedicated virtual disks for a database. Therefore, when the next screen is displayed, specify that the virtual disk being created is to be dedicated by selecting the **Dedicated** radio button. When this is done, click on the **Next** button.

Create Disk

Select a Virtual Disk Type
Select whether to create a dedicated disk or a shared disk.

The virtual disk can be created as a dedicated disk or a shared disk. A dedicated disk is accessible from only one system. A shared disk is used only with Microsoft® Cluster Services.

Select "Shared" if you are using this disk as a Microsoft Cluster Services physical disk resource. Otherwise, select "Dedicated".

Specify Virtual Disk Type

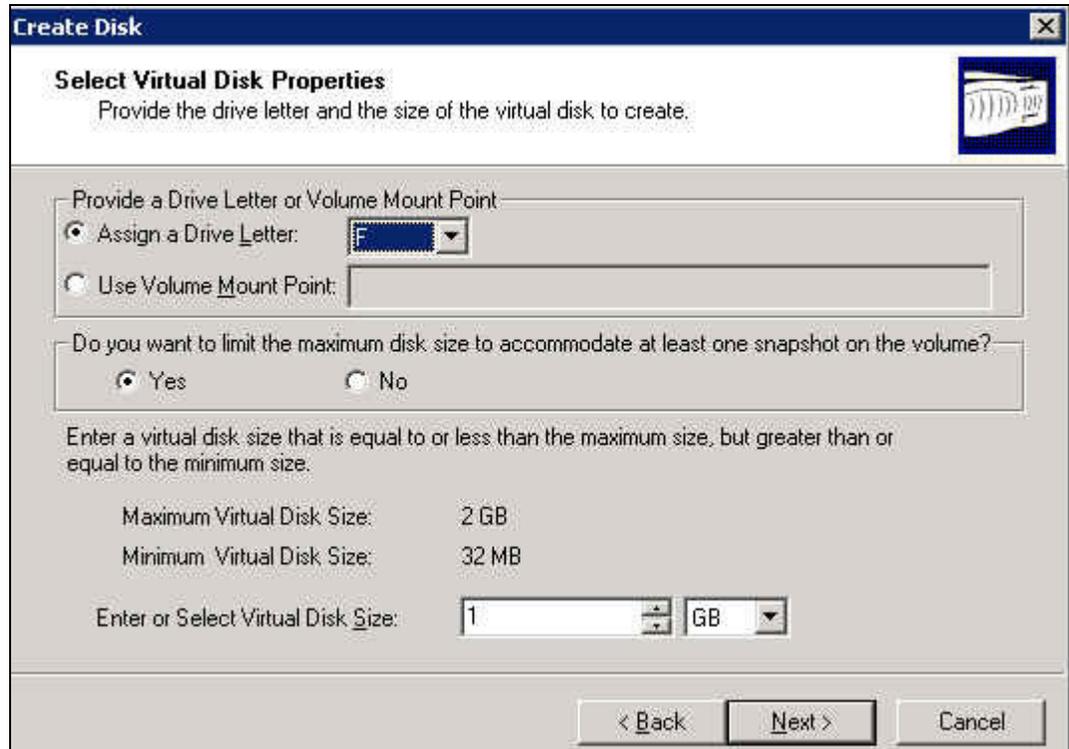
Dedicated

Shared (Microsoft Cluster Services only)

< Back Next > Cancel

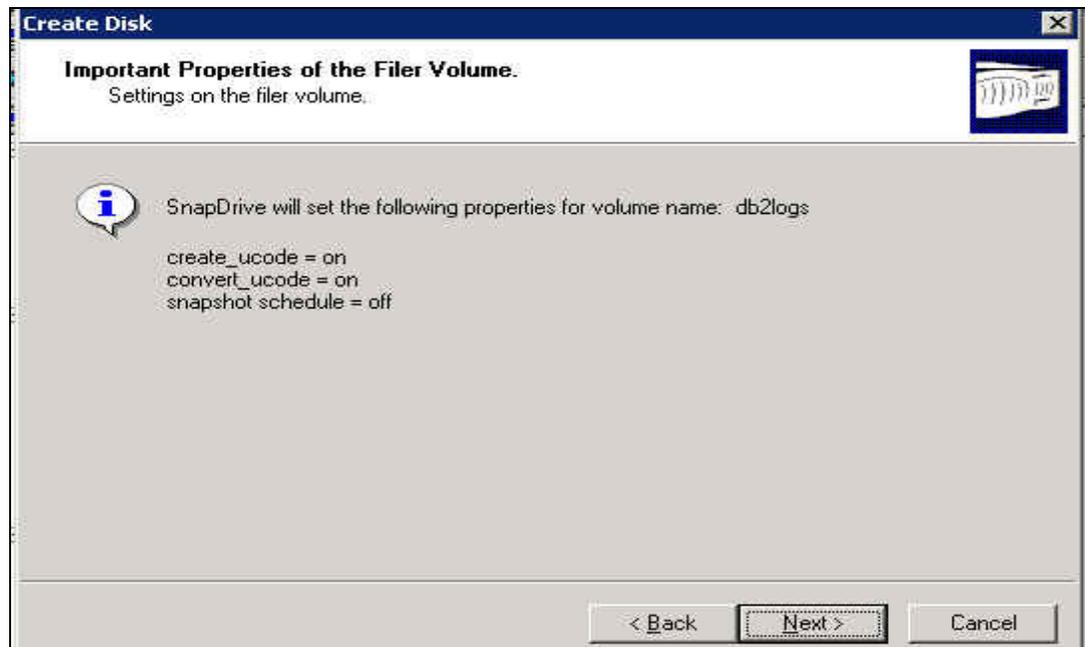
Screen 5: Create disk wizard.

- f). By default, SnapDrive will select a Drive Letter and Size for the disk. You can change the disk size by specifying appropriate space available on your volume. After verifying the drive letter and entering the disk size, click the **Next** button.



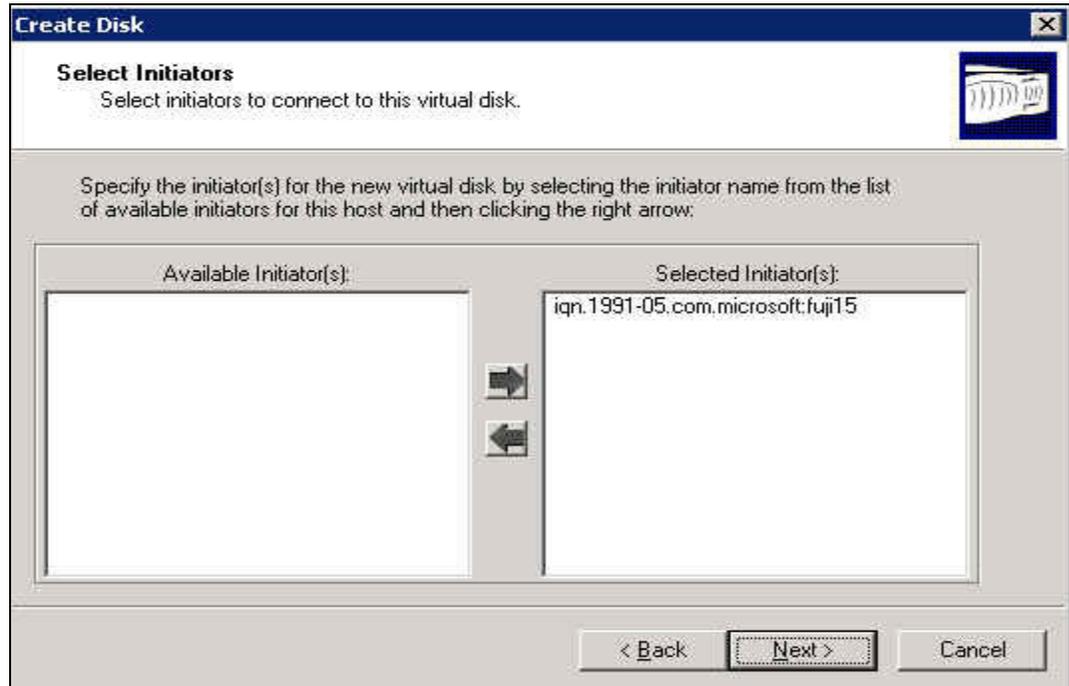
Screen 6: Create disk wizard.

- g). The next page of the wizard will inform you of some changes SnapDrive will make for the volume. After reviewing these changes, click the **Next** button.



Screen 7: Create disk wizard.

- h). Next, you need to specify the initiator that will be used to connect to the virtual disk. After selecting the appropriate initiator from the list of available initiators shown, click the **Next** button.



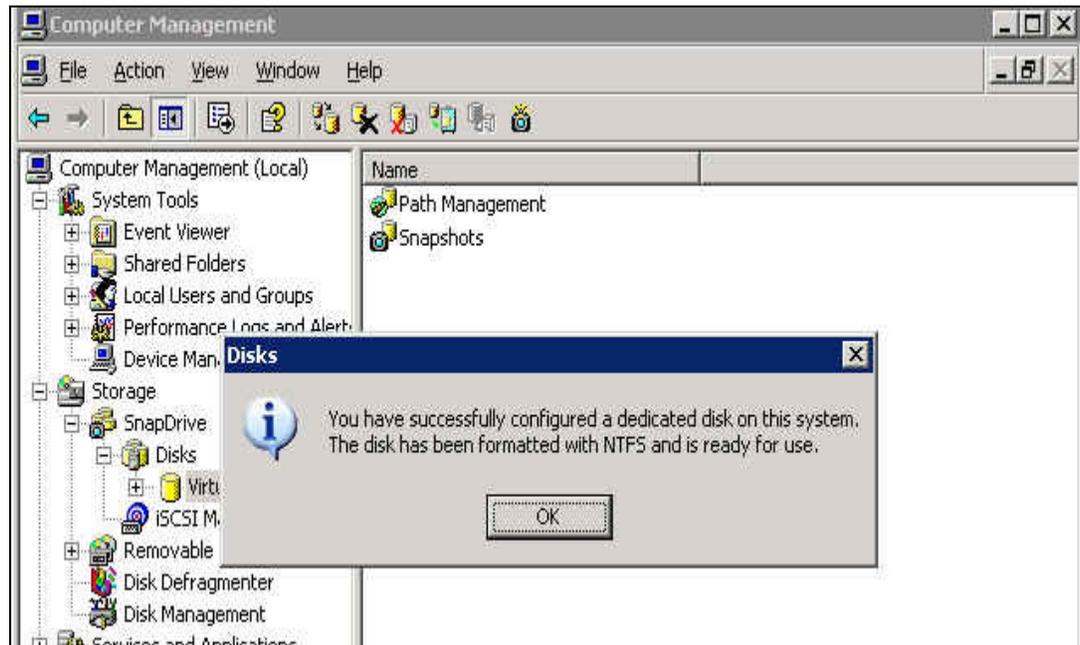
Screen 8: Create disk wizard.

- i). The confirmation screen will provide you with summary information about the disk that is about to be created. When ready to proceed, click the **Finish** button and the system will create a virtual disk. It may take some time as the virtual disk is formatted in the NTFS format.



Screen 9: Create disk wizard.

- j). After completing the above steps you should receive confirmation that the virtual disk has been created. (Your screen should look similar to the following.)



Screen 10: Create disk wizard.

Repeat these steps and create virtual disks for all the volumes that are to be used by the database.

NOTE: When you create a virtual disk using SnapDrive, SnapDrive creates an igroup for the initiator and LUNs created for the initiator are mapped to this igroup.

8. Creating a New Database on a NetApp storage system

This section describes the steps needed to create a new DB2 database that has its data and transaction log files stored on the virtual disks created using SnapDrive. For the purpose of this document, only the creation of a single partition database is covered. For further details on installation, refer to the IBM document “*Quick Beginnings for DB2 Servers.*” The creation of a DB2 database is a very straightforward process that utilizes standard DB2 commands – to create a DB2 database, complete the following steps:

- 1) Log in as the local administrator and install the DB2 for Windows software on the database server. For installation instructions please refer to DB2 manual “*Quick Beginnings for DB2 Servers.*”
- 2) Once DB2 is installed, start DB2 CLI from the menu by selecting **Start -> Programs -> IBM DB2 -> Command Line Tools -> Command Line Processor**, or **Start -> Run ->** and typing **db2cmd**. Create a DB2 instance by executing the following command in the command window on the database server:

```
db2icrt [InstanceName]
```

where

- **InstanceName** identifies the name assigned to the DB2 instance.

For example, to create an instance named `db2inst1`, you would execute the following command:

```
db2icrt db2inst1
```

Note: The DB2 Control Center can be used to create and manage database instances and databases. It is a tool that provides a robust database and instance administration through a GUI-based interface. It also has very strong integration with other DB2 tools such as the Health Center, Task Center, Replication Center, and the Journal.

- 3) Verify whether the database instance was created successfully by executing the following command in the command window on the database server:

```
db2ilist
```

- 4) Next, create a new database on a virtual disk seen by the database server. You can do so by executing the following command:

```
db2 "CREATE DATABASE [DBName] on [VirtualDisk]"
```

where

- **DBName** identifies the name assigned to the database.
- **DBServerName** identifies the drive letter assigned to the virtual disk seen by the database server.

The virtual disk drive letter has to be explicitly specified in the CREATE DATABASE command.

For example, to create a database named `MYDB` on a virtual disk that is assigned a drive letter `F`, you would execute the following command on the database server:

```
db2 "CREATE DATABASE MYDB ON F:"
```

- 5) By default, when a database is first created, circular logging is used and primary log files are created in the subdirectory

```
"[DBPath]\[InstanceName]\[NodeName]\SQLnnnn\SQLLOGDIR\."
```

Where

- **DBPath** identifies the name of directory the database is created on.
- **InstanceName** identifies the name of the DB2 instance used for the database.
- **NodeName** identifies the database node name.
- **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`.

`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`.

It is recommended that you move transaction log files to another virtual disk that has been created on the storage system. The new transaction log location can be specified by updating the database configuration parameter `NEWLOGPATH`. You would execute the following command on the database server in order to update this configuration parameter:

```
db2 "UPDATE DB CFG FOR MYDB USING NEWLOGPATH [NewLogLocation]"
```

where

- **NewLogLocation** identifies the new location for the transaction log files of the database.

For example, to move the transaction log files for a database named `mydb` to virtual disk `G` that was created on a volume on the storage system, you would execute the following command on the database server:

```
db2 "UPDATE DB CFG FOR MYDB USING NEWLOGPATH G:"
```

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) You can verify if the database was created correctly and exists on the virtual disk specified 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.

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

```
db2 "CONNECT TO [DBName]"
```

where

- *DBName* identifies the alias assigned to the database (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"
```

- 8) List the tablespaces that were created along with the database by executing the following command on the database server:

```
db2 "LIST TABLESPACES"
```

- 9) Verify that the tablespace container(s) associated with each tablespace refer to a file or directory located on the virtual disk(s) that were created on the volumes on the storage system. Tablespace container information can be obtained by executing the following command:

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

where

- *TBSpaceID* identifies the id assigned to the tablespace whose container information is to be retrieved.

If the database was created correctly, the container information returned by the above command should identify the file or directory that was assigned to the tablespace specified when the **CREATE DATABASE** command was executed. The tablespace containers should reside on the virtual disks.

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"
```

- 10) Additional tablespaces can be created on NetApp storage system volumes using the "CREATE TABLESPACE" command. For detailed syntax for this command, refer to the IBM DB2 SQL Reference, Vol. 2.

9. Migrating an existing DB2 database to a NetApp storage system

The migration of existing DB2 databases to virtual disks that have been created on a NetApp FAS or IBM N Series storage system is also a fairly straightforward process that utilizes standard DB2 commands. This section describes the steps needed to convert an existing DB2 database whose data and transaction log files reside on a local disk to a new DB2 database whose data and transaction log files reside on virtual disks that reside on NetApp/N Series storage system volumes.

Note: The items described in section 4, Infrastructure, 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 a NetApp/N Series storage system volume.

9.1. 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 [DBName]"
```

where

- *DBName* identifies the alias assigned to the database (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 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) Create a full backup image of the database to be migrated by issuing the following command on the database server:

```
db2 "BACKUP DATABASE [DBName] TO [BKUPLocation]"
```

where

- *DBName* identifies the alias assigned to the database (if no alias was specified, the alias will be the same as the name assigned to the database).
- *BKUPLocation* identifies the location where the database backup image is to reside.

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

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

- 4) 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 [DBNameOld] FROM [BKUPLocation] TO [NewLocation]  
INTO [DBNameNew] REDIRECT"
```

```
db2 "SET TABLESPACE CONTAINERS FOR [TSpaceID] USING ('[Directory]')"
```

or

```
db2 "SET TABLESPACE CONTAINERS FOR [TSpaceID] USING (FILE '[FileName]'  
[Size])"
```

or

```
db2 "SET TABLESPACE CONTAINERS FOR [TSpaceID] USING (DEVICE  
' [DeviceName] ' [Size])"
```

```
db2 "RESTORE DATABASE [DBNameOld] CONTINUE"
```

where

- *DBNameOld* identifies the alias assigned to the database that was backed up.
- *BKUPLocation* identifies the location where the database backup image is stored.
- *NewLocation* is the location where the overhead files associated with the database are to reside.
- *DBNameNew* is the name that is to be assigned to the new database that will be created from the backup copy.
- *TSpaceID* is the unique identifier (0, 1, 2, etc.) that has been assigned to the tablespace that the new container information is to be assigned for.
- *Directory* identifies the new directory SMS tablespace the data is to be stored in.
- *FileName* identifies the name of the file that the DMS tablespace data is to be stored in.
- *DeviceName* is the name of the device that is to be used to store tablespace data.
- *Size* is the size, in 4, 8, 16, or 32 k 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 virtual disks on NetApp FAS or IBM N Series storage system volumes, you should use a script that looks similar to the following:

```
BKUPLocation = /dbbkup/mydb  
NewLocation = /mnt/db2data/mydb  
DBNameOld = mydb  
DBNameNew = mydb  
db2 "restore database $ DBNameOld from $ BKUPLocation TO $ NewLocation  
INTO $ DBNameNew redirect without prompting" > restore.out  
db2 "SET TABLESPACE CONTAINERS FOR 0 USING (PATH $ NewLocation  
/catalog)"  
db2 "SET TABLESPACE CONTAINERS FOR 1 USING (PATH $ NewLocation /temp)"  
db2 "restore db $ DBNameOld 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.

- 5) Verify that the database was successfully migrated by performing a few simple tests (as outlined in step 9 of *Section 8*).

9.2. 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 DB2 database that uses tablespaces that are associated with files stored on virtual disks that are created on a NetApp 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), etc., 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's 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 [DBName] -e -o [OutFile]
```

where

- **DBName** identifies the alias assigned to the database that is to be migrated.
- **OutFile** is the name and location of the file that the DDL information produced by `db2look` command is to be written to.

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 c:\db2migrate\db2inst1\mydb.ddl
```

When executed, the command will place the `mudb.ddl` file in the directory `c:\db2migrate\db2inst1`.

- 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 formatted file:

```
db2move [DBName] EXPORT
```

where

- **DBName** identifies 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 virtual disks that are created on a NetApp FAS or IBM N Series storage system as outlined in section 8.

NOTE: If you want to assign the same name to the database being created as the database being migrated, you must first drop or uncatalog the database being migrated.

- 6) Establish a connection to the database created in step 5 of this section and create the database objects using the script created in the `db2look` utility in step 2:

```
db2 "connect to mydb1"  
db2 -t -f c:\db2migrate\db2inst1\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 simple tests as outlined in step 9 of section 8.

10. Performance Considerations

DB2 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. One of these registry variables, **DB2_PARALLEL_IO**, should always be set when DB2 is used in conjunction with a NetApp/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.

10.1. 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 a NetApp FAS or IBM N Series storage system volume, performance may be improved if read and write calls are issued in parallel.

To force DB2 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, whereas the second command will turn parallel I/O on only for tablespaces 1, 2, 4, and 8.

10.2. The DB2 Configuration Advisor

When a database is created using the CREATE DATABASE command it is created with a set of 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 is limited, you can use 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**.

10.3. Additional factors that affect performance

Finally, when running a DB2 database whose data and transaction files reside on a NetApp FAS or 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 volume named **db2data** and the database's transaction log files were stored on a volume named **db2logs**. Both of these volumes were created on a NetApp FAS or IBM N Series storage system.)

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.

11. Enabling NVFAIL on the storage system for additional data protection

When storing database data on a NetApp FAS or IBM N Series storage system, it is a good idea to enable the `nvfail` feature for the volumes used for the database. The `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 NetApp storage system:

```
vol options [VolName] nvfail on
```

where

- `VolName` identifies the name of a NetApp 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 storage system:

```
vol options db2data nvfail on
```

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:

- ff-lines the LUNs 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 have 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 storage system). The format of the `/etc/nvfail_rename` file is simply the name of a file found on the 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, suppose the `/etc/nvfail_rename` file contents looked similar to the following:

```
F:\db2data\db2inst1\NODE0000\SQL0001\SQLSPCS.1
F:\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:

```
F:\db2data\db2inst1\NODE0000\SQL0001\SQLSPCS.1.nvfail
F:\db2data\db2inst1\NODE0000\SQL0001\SQLSPCS.2.nvfail
```

Since this occurs before the 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 a NetApp FAS or IBM N Series storage system, you may want to create a `/etc/nvfail_rename` file that contains the names of every file on the storage system that is being used by a particular DB2 database.

12. Caveats

The information presented in this document has been tested by NetApp using only a limited set of hardware and software options. Therefore, your experience may differ from that presented here. If you have any problems with the techniques shown in this document, please contact the [author](#).

13. Important links

1. DB2 Product Manual "Quick Beginning for DB2 Server"
(<http://www-306.ibm.com/software/data/db2/udb/support/manualsv9.html>)
2. Data ONTAP 7.1 Commands: Manual Page Reference, Volume 1
(http://now.netapp.com/NOW/knowledge/docs/ontap/rel71_vs/pdfs/ontap/cmdref1.pdf)
3. SnapDrive 4.1 for Windows Installation and Administration Guide
(<http://now.netapp.com/NOW/knowledge/docs/san/>)
4. The Compatibility and Configuration Guide for NetApp FCP and iSCSI Products
(http://now.netapp.com/NOW/knowledge/docs/san/fcp_iscsi_config/)
5. Data ONTAP 7G: The Ideal Platform for Database Applications FlexVol and FlexClone
(<http://www.netapp.com/library/tr/3373.pdf>)
6. Introduction to Data ONTAP 7G
(<http://www.netapp.com/library/tr/3356.pdf>)
7. Database Layout with Data ONTAP 7G
(<http://www.netapp.com/library/tr/3411.pdf>)



© 2006 NetApp, Inc. All rights reserved. Specifications subject to change without notice. NetApp, the NetApp logo, Data ONTAP, FilerView, SnapDrive, SnapMirror, SnapRestore, and WAFL are registered trademarks and NetApp, FlexClone, FlexVol, NOW, RAID-DP, and Snapshot are trademarks of NetApp, Inc. in the U.S. and other countries. Windows, Windows NT, and Microsoft are registered trademarks of Microsoft Corporation. All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such.