



Technical Report

IBM DB2 on NetApp Storage: Deployment and Best Practices

Bobby Oommen and Jawahar Lal, NetApp, Inc.
Updated May 2010 | TR-3272

ABSTRACT

This document describes deployment process and best practices for DB2 database in a NetApp® and UNIX® environment. It elaborates on the configuration steps for the UNIX host and the NetApp storage system. The UNIX host types covered in this document include Linux®, IBM AIX, Sun™ Solaris™, and HP-UX.

TABLE OF CONTENTS

1	PURPOSE AND SCOPE	3
2	ASSUMPTIONS	3
3	ARCHITECTURE	3
3.1	SUPPORTED VERSION OF IBM DB2	4
3.2	DATABASE SERVER WITH DB2 ADMINISTRATOR AND USER ACCOUNTS	4
3.3	NETAPP STORAGE SYSTEM	5
3.4	NETWORK	5
4	CONFIGURATION OVERVIEW	5
5	WHY STORE A DB2 DATABASE ON NETAPP STORAGE?	6
6	CONFIGURING THE NETAPP STORAGE SYSTEM	7
7	CONFIGURING THE DATABASE SERVER IN A NAS ENVIRONMENT	11
7.1	LINUX	12
7.2	IBM AIX	13
7.3	SUN SOLARIS	14
7.4	HP-UX	15
8	CONFIGURING THE DATABASE SERVER IN A SAN ENVIRONMENT	16
8.1	LINUX	16
8.2	IBM AIX	21
8.3	SUN SOLARIS	26
8.4	HP-UX	32
9	CREATING A NEW DATABASE ON NETAPP STORAGE	38
10	MIGRATING A DB2 DATABASE TO NETAPP STORAGE	40
10.1	MIGRATING A DB2 DATABASE USING THE REDIRECTED RESTORE OPERATION	40
10.2	MIGRATING A DB2 DATABASE USING DB2LOOK AND DB2MOVE	41
11	BEST PRACTICES	43
11.1	DESIGNING A DATABASE WITH RECOVERY IN MIND	43
11.2	PERFORMANCE	43
11.3	THE DB2_PARALLEL_IO REGISTRY VARIABLE	44
11.4	DB2 DATABASE STORAGE CONSIDERATIONS	44
11.5	DB2 CONFIGURATION ADVISOR	46
11.6	ADDITIONAL FACTORS THAT AFFECT PERFORMANCE	46
12	ENABLING NVFAIL ON THE STORAGE SYSTEM FOR ADDITIONAL DATA PROTECTION	46
13	CAVEATS	47
14	IMPORTANT LINKS	47
	REVISION HISTORY	48

1 PURPOSE AND SCOPE

This document describes the steps necessary to deploy an IBM DB2 database in a UNIX and NetApp environment:

- Preparing a UNIX server for DB2 installation
- Preparing a storage system for DB2 database storage
- Installing DB2 on the UNIX server and creating a database on the storage system
- Separating database transaction log files from data files and storing on the storage system
- Migrating an existing database from a local disk to a storage system
- Best practices to deploy DB2 in a NetApp environment

2 ASSUMPTIONS

This document was written assuming the reader has the following:

- Familiarity with DB2 commands and features
- A basic understanding of the operation of storage systems
- Basic UNIX administration skills (the examples provided might require modifications before they can be run under your version of UNIX)

The sample scripts in this technical report assume the following:

- The name of the storage system used is `netapp`
- The name of the UNIX host, henceforth known as a database server, is `db2srv1`
- The user name for DB2 instance owner is `db2inst1`
- The DB2 instance on the UNIX server is `db2`
- The home directory for the DB2 instance is `/home/db2inst1`
- Database data files are stored on a FlexVol® volume named `dbdata`, and the storage system is `netapp:/vol/dbdata`
- The location of database transactions log files on the storage system is `netapp:/vol/dblogs`
- The mount point on the UNIX server for the volume is `netapp:/vol/dbdata` is `/mnt/dbdata`
- The mount point on the UNIX server for the volume is `netapp:/vol/dblogs` is `/mnt/dblogs`

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

3 ARCHITECTURE

In order to use DB2 for UNIX in conjunction with a NetApp storage system, the following elements are required (Figure 1):

- A supported version of IBM DB2 Enterprise Server Edition
- One or more UNIX host servers (database servers) with DB2 administrator and user accounts
- One or more NetApp storage systems running Data ONTAP® 7G (or later)
- Network connectivity between database server and storage system
- FCP or iSCSI HBA (not required for NFS or iSCSI software initiator)

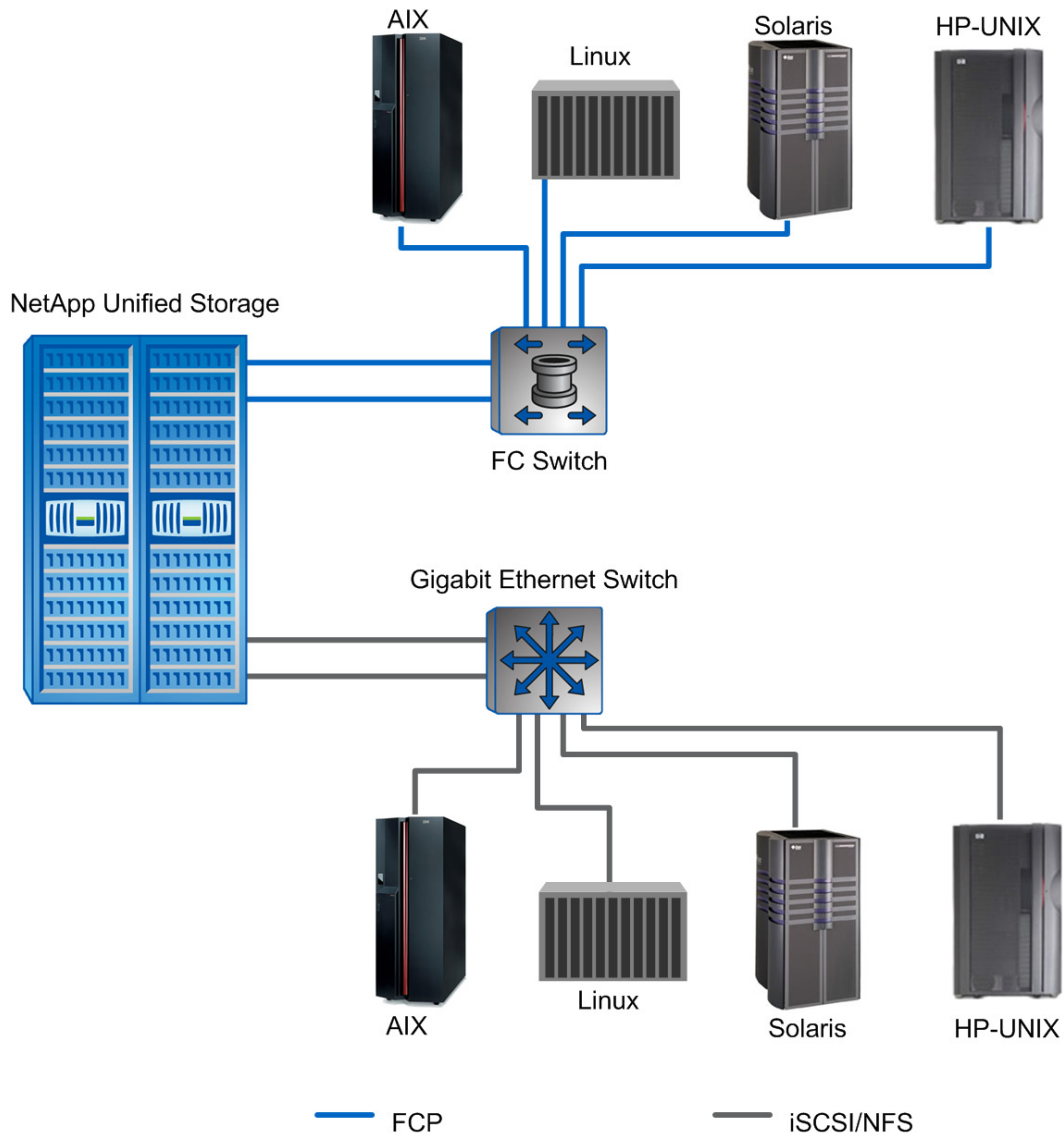


Figure 1) Database setup on NetApp unified storage system using FC, iSCSI, and NFS protocol.

3.1 SUPPORTED VERSION OF IBM DB2

To perform the steps outlined in this document, you must have DB2 V9.x installed and running on a UNIX server or workstation. Before proceeding with installation of DB2, check the system requirements described in the DB2 installation documentation for your host platform and follow its steps to install and configure the DB2 instance and database.

3.2 DATABASE SERVER WITH DB2 ADMINISTRATOR AND USER ACCOUNTS

To perform the steps outlined in this document, 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 installed on your UNIX server, chances are that the appropriate DB2 user accounts already exist. However, if 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.

3.3 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 NetApp storage system. Any NetApp storage system running a supported version of Data ONTAP can be used to store a DB2 database and its transaction log files. To produce this document, we used a NetApp FAS3000 series storage system running Data ONTAP 7G.

3.4 NETWORK

NetApp's unified storage system supports FCP, iSCSI, FCoE, and NFS protocols. The type of connection established depends on the type of environment. 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, an Ethernet connection will work for both IP SAN and NAS. To support FCoE, you will need supported 10GbE HBA.

Note the following regarding network performance:

- Flow control and full-duplex operation should be enabled on the Ethernet interfaces on both the UNIX server and the storage system.
- When using Gigabit Ethernet adapters, consider turning on jumbo frames.
- Consider using local locking for all mount points that communicate with the storage system.
- If possible, dedicate at least one private network to the connection between the DB2 server and the storage system. This can be done using a switch or a crossover cable on most networking topologies. We used Gigabit Ethernet and a crossover cable in the development this document. This is recommended for the following reasons:
 - Any issues of contention or latency are eliminated if the DB2 server and the storage system are the only two nodes in the network.
 - Creating a private network connection provides 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 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.
- To increase the occurrence of parallel I/Os between the database server and the storage system, create separate volumes on the storage system for database data and transaction logs, and use separate mount points to mount them to the database server.
- Each dedicated network connection should use a high-end networking protocol such as Gigabit Ethernet, which provides broad bandwidth between the 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.

4 CONFIGURATION OVERVIEW

Traditional installation of DB2 on a UNIX platform calls for storing database data and corresponding transaction log files on local disks. With a NetApp storage system, a database administrator can take full advantage of technologies such as Snapshot™, SnapMirror®, RAID-DP®, FlexVol, and FlexClone® by placing the database's system files, data, and transaction log files on the storage system.

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 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, then the DB2 database manager will attempt to balance the data load across all containers used.

NetApp Data ONTAP 7G supports a virtual storage layer called flexible volumes, or FlexVol volumes. A FlexVol volume is created within an aggregate and is loosely coupled with its containing aggregate. The FlexVol volume provides greater performance and can grow or shrink as needed by executing a simple command. Data ONTAP makes it easy to control the placement of related database objects on a FlexVol volume. For more information on creating and managing FlexVol volumes, refer to the `vol` command in the appropriate NetApp documentation available online at the NetApp [NOW™](#) Web site.

Note: The terms “volume” and “FlexVol volume” are used interchangeably throughout this document.

5 WHY STORE A DB2 DATABASE ON NETAPP STORAGE?

Running a DB2 database with data and transaction log files stored on a 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 NetApp 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 copy 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.

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. NetApp Snapshot technology, which can create database backups in a matter of seconds without bringing a database down, can be used as complementary technology to provide higher system uptime.

HIGH RELIABILITY

The RAID architecture used for NetApp storage systems is unique and provides greater reliability than many traditional RAID implementations. If a disk in a NetApp RAID group fails, it is reconstructed automatically without any user intervention. Additionally, NetApp supports RAID-DP architecture. RAID-DP is considered approximately 4,000 times more reliable than traditional RAID. For more details on RAID-DP, 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 have 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 NetApp storage system Data ONTAP operating system, only one container needs to be defined for each tablespace used.

FAST READ RESPONSE

NetApp Flash Cache allows customers to improve performance for workloads that are random read intensive without adding more high-performance disk drives. The Flash Cache stores most active data in memory and hence improves throughput by lowering latency.

NO COST DATABASE CLONE

NetApp FlexClone technology creates true clones, instantly replicated data volumes and data sets, without requiring additional storage space. It offers customers great savings in terms of storage and manageability.

6 CONFIGURING THE NETAPP STORAGE SYSTEM

Note: The items described in “Architecture” on page 3 must be in place for this information to be valid.

This section describes the steps involved in installing the DB2 software on a database server and creating a DB2 database on a NetApp storage system. Before you begin the installation process, make sure that the appropriate operating system kernel updates have been applied and that all the kernel parameters have been updated per the installation documentation. Additional information on the DB2 installation procedure can be found at the IBM DB2 Web site in the manual “Quick Beginnings for DB2 Servers.”

Before creating a database on a 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 NetApp storage system, while the other needs to be performed on the database server.

1. Install Data ONTAP 8.0 7-mode on your NetApp storage system if it is not already installed. For Data ONTAP installation and setup instructions, refer to “Data ONTAP 8.0 7-Mode Software Setup Guide” on the NOW Web site.
2. Activate the appropriate license keys.

On the NetApp storage system, 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. Activate license keys by executing the following command from the NetApp storage system:

```
license add [LicenseCode]
```

Where

`LicenseCode` is the product license key you obtained from NetApp.

Note: Parameters shown in angle brackets (< >) are optional; parameters or options shown in square brackets ([]) are required; options shown in curly brackets ({ }) are mutually exclusive, and one value must be selected; a comma followed by ellipses (...) indicates that the preceding parameter can be repeated multiple times.

For example, to activate the license code `123XYZABCD` obtained from NetApp for NFS, execute the following command on the storage system:

```
license add 123XYZABCD
```

3. Update the `/etc/host` file on the storage system used.

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 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® to add entries:

```
[HostIP] [HostName]
```

Where

- `HostIP` is the IP address assigned to the database server.
- `HostName` is the network name of the database server.

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

```
172.17.32.112 db2srv1
```

4. Enable “rsh” access for the database server.
 - a. Execute the following command:

```
options rsh.enable on
```
 - b. Add the database host and user name entry to the `/etc/hosts.equiv` file found on the storage system. The entry to this file looks similar to the following:

```
[HostIP] [UserName]
```

Where

`UserName` is the name of the user who has been granted `rsh` access to the storage system.

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

```
db2srv1 db2inst1
```

5. Change the storage system options for NFS.

The storage system default setting for parameter `options.nfs.tcp.recvwindowsize` might not be appropriate for the database workload. It is recommended that you set this value to 65536 or higher.

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

To create a database on the NetApp storage system, you must create aggregates, flexible volumes, and, in some cases, LUNs.

- a. Create an aggregate.

An aggregate is a physical pool of storage at RAID level and is created using the following command:

```
aggr create [AggrName] -f -t {raid4 | raid_dp} -r [RaidSize]
[nDisk]@[DiskSize] | -d <disk1, disk2,..,diskn>
```

Where

- `AggrName` is the name assigned to the aggregate.
- `RaidSize` is the size of the RAID group that will be created by the command implicitly.
- `nDisk` is the number of disks that are going to be used for the aggregate.
- `DiskSize` is the size of the disks that are being used for the aggregate.

For example, to create an aggregate named `db2aggr1` that has eight 72GB disks in it and that uses RAID-DP, execute the following command:

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

- b. Create FlexVol volumes.

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. A FlexVol volume can be created using the following command:

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

Where

- `VolName` is the name assigned to the FlexVol volume.
- `AggrName` is the name assigned to the aggregate.
- `VolSize` is the size of the volume in terms of megabytes, gigabytes, terabytes, and so on.

For example, to create a flexible volume named `dbdata` that is 100GB in size and that resides in the aggregate named `db2agg1`, execute the following command:

```
vol create dbdata db2agg1 100GB
```

- c. Create LUNs.

LUN is an acronym for logical unit number. One or more LUNs can be created within a volume. LUNs are used to map a storage system in SAN environments. A LUN is created by executing the following command:

```
lun create -s [LunSize] -t [OSType] [LunPath]
```


Where

- LunSize is the size of the LUN.
- OSType is the type of operating system the LUN is created for.
- LunPath is the path for the LUN.

For example, to create a LUN named `/vol/dbdata/lun1` that is 50GB in size and resides in a FlexVol volume named `dbdata`, execute the following command:

```
lun create -s 50G -t linux /vol/dbdata/lun1
```

Note that the LUN created in this example will be accessed by a Linux host.

7. Specify the volume security settings.

You must make sure that the volumes on the storage system used for the DB2 database support either the UNIX or the mixed security style. Specify the security setting of a volume with the following command:

```
qtree security [VolName] { unix | mixed | NTFS }
```

Where

`VolName` is the name assigned to the volume.

For example, to change the security style to UNIX for a FlexVol volume named `dbdata`, execute the following command:

```
qtree security /vol/dbdata unix
```

Repeat this step to change the security style for all the volumes in 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 any volume used for the database and its transaction log files. The automatic Snapshot feature can be turned off using the following command:

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

For example, to turn off the automatic Snapshot feature for a volume named `dbdata`, execute the following command:

```
vol options dbdata nosnap on
```

Repeat this step to turn the auto Snapshot off for all the volumes in the database.

9. Export the volumes for NFS (skip this step for FCP or iSCSI).

To access the volumes on a storage system using the NFS protocol, you must export them. A volume can be exported by adding an entry to the `/etc/exports` file on the storage system and giving the database server appropriate access to it. Use `FilerView` or the `exportfs` command for this purpose. NFS exports are managed using the following command:

```
exportfs -p rw=[HostName],root=[HostName] [PathName]
```

Where

- `HostName` is the name assigned to the database server.
- `PathName` is the path for the volume.

For example, to create an export entry for a volume named `dbdata` and allow `root` access to it from a database server named `db2srv1`, execute the following command:

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

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

10. Set up initiators and LUNs for FCP or iSCSI (skip this step for NFS).

In a SAN or IP SAN environment, UNIX hosts serve as initiators, and the storage systems serve as targets. The logical storage layer created on a 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 known 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, find the WWPN or WWNN for the initiator. In order to determine WWPN/WWNN, refer to “Configuring the Database Server in a SAN Environment” on page 16. Use the WWPN or WWNN to create an igroup on the storage system by executing the following command:

```
igroup create -{ f | I } -t { linux | aix | solaris | hpux } [iGroupName]
[NodeIdentifier]
```

Where

- `iGroupName` is the name assigned to the initiator group being created.
- `NodeIdentifier` is the initiator's WWPN or WWNN.

For example, to create an igroup named `db2srv1_igp` for an initiator on a Linux host that has a WWPN value of `10000000c93a069b`, execute the following command:

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

- b. Create appropriate LUNs and LUN mappings.

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

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

Where

`LunPath` is the path for the LUN.

For example, to create a 75GB LUN within a FlexVol volume named `dbdata`, execute the following command:

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

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

```
lun map [LunPath] [iGroupName] [LunID]
```

Where

- `iGroupName` is the name assigned to the initiator group.
- `LunID` is a numeric ID used to map the LUN to an igroup.

For example, to create a LUN mapping for a LUN named `/vol/dbdata/data1` to an igroup named `db2srv1_fcp_igp`, execute the following command:

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

- c. Obtain a node name for the storage system.

Target storage systems are identified by a unique node name used for creating persistent binding for the storage devices on the host. Find out the node name assigned to a storage system by executing the following command:

```
{ fcp | iscsi } NodeName
```

For example, the following command would find the node name assigned to an iSCSI target:

```
iscsi nodename
```

For iSCSI, the traffic needs to flow on the private storage GbE only, not on the public network interface. One method to achieve that is to disable the iSCSI data traffic on the e0a interface (or whichever interface is the public network interface in the configuration) by using one of the following commands on the storage system:

```
iswt interface disable e0 (for Data ONTAP 7.0 and earlier)
iscsi interface disable e0 (for Data ONTAP 7.1 and later)
```

7 CONFIGURING THE DATABASE SERVER IN A NAS ENVIRONMENT

After creating volumes on the storage system and updating the `/etc/exports/` file, the database server must be configured to make use of these volumes. The steps to configure the database server to use the storage system volumes over NFS depend on the UNIX flavor.

The following steps will 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 storage system to the `/etc/hosts` file found on the database server. The entry in the `/etc/hosts` file should look similar to the following:

```
[StorageSystemIP] [StorageSystemName]
```

Where

- `StorageSystemIP` is the network IP address assigned to the storage system.
- `StorageSystemName` is the name assigned to the storage system.

For example, to add a storage system named `netapp` that has the IP address `10.60.175.57` to the `/etc/hosts` file found on a database server, use the following command:

```
10.60.175.57 netapp
```

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

To produce this document, we used the operating system's authentication mechanism; we created a group named `db2adm` and a user account named `db2inst1` by executing the following commands:

```
groupadd -g 700 db2adm          # group of users to be granted SYSDBA authority
useradd -c "DB2 Instance Owner" -u 701 -g db2adm -G db2adm -d /home/db2inst1
db2inst1 -p db2inst1
```

Additional groups and user accounts can be created using the `groupadd` and `useradd` commands.

4. Create mount points to mount storage system volumes by executing the following command:

```
mkdir -p [MountPoint]
```

Where

`MountPoint` is a directory path on the database server used for mounting a FlexVol volume.

For example, to create a mount point named `/mnt/dbdata`, execute the following command:

```
mkdir -p /mnt/dbdata
```

Repeat this step and create a mount point for each FlexVol volume to be used for the database.

7.1 LINUX

SET UP THE KERNEL PARAMETERS

The default kernel parameter values might not be good enough to run DB2 successfully on your Linux server. Some of the recommended Linux kernel parameter values for DB2 are:

```
kernel.shmmax=268435456    (for 32-bit)
kernel.shmmax=1073741824  (for 64-bit) // 80% of total memory in bytes
kernel.shmall=1887436    // maximum shared memory allocation 4k pages
kernel.msgmni=1024
fs.file-max=8192
kernel.sem=250 32000 32 1024
```

DB2 checks the values of the `kernel.sem`, `kernel.msgmni`, and `kernel.shmmax` parameters when you enter the `db2start` command, and it changes them to the recommended values if the current values are not optimal. However, other parameters might need to be changed manually.

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

```
ipcs -l
```

To make the kernel parameter changes permanent over a system reboot, do the following:

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

```
kernel.shmmax=268435456          // memory in number of bytes
kernel.msgmni=1024
kernel.sem=250 32000 32 1024
```

For a 64-bit kernel and 8GB-memory system, the kernel setting should look similar to the following:

```
kernel.shmmax=1073741824    // memory in number of bytes
kernel.shmall=1887436      // maximum shared memory allocation 4k pages.
kernel.msgmax=65535
kernel.msgmnb=65535
```

Note: The SHMALL parameter value can be calculated using the following formula:

$$\text{Kernel.shmall} = \text{ceil}\left(\frac{\text{shm max}}{4096}\right)$$

2. Tune the Linux kernel parameter.

The kernel configuration parameter `tcp_slot_table_entries` sets the maximum number of (TCP) RPC requests that can be in flight. A high value means high parallelism in the RPC layer. The default value is 16. Increasing this value from 16 to 128 shows a significant performance improvement in our lab tests.

To set this parameter to 128, add the following entry to `/etc/sysctl.conf` file:

```
sunrpc.tcp_slot_table_entries = 128
```

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

The entries in the `sysctl.conf` file are read by the network initialization script during startup. On some distributions, you might be required to add `sysctl -p` in one of the system initialization files (for example, `rc.local`) so that kernel parameters are set after each reboot. For a detailed list of recommended kernel parameters, refer to the Preinstallation Task chapter of “Quick Beginning for DB2 Server.”

MOUNT THE FILE SYSTEM

1. Append an entry for each FlexVol volume to the `/etc/fstab` file found on the database server. The entry should look similar to the following:

```
[StorageSystemName]:[VolName] [MountPoint] nfs
hard,rw,nointr,rsize=32768,wsiz=32768,bg,vers=3,tcp 0 0
```

Where

- `StorageSystemName` is the name assigned to the storage system.
- `VolName` is the name assigned to a FlexVol volume on the storage system.
- `MountPoint` is a directory path on the database server used to mount a FlexVol volume.

For example, to mount a FlexVol volume named `dbdata` on a mount point named `/mnt/dbdata`, add the following line to the `/etc/fstab` file on the database server:

```
netapp:/vol/dbdata /mnt/dbdata nfs hard,rw,nointr,rsize=32768,
wsiz=32768,bg,vers=3,tcp 0 0
```

In this example, the maximum block transfer size across NFS mounts is 32KB, but NFS V3 supports blocks up to 64KB in size. You can tune the mount option to see if changing it to a higher value improves performance.

2. Mount and make the storage system volumes available by executing the following command:

```
mount [MountPoint]
```

For example, to mount a NetApp storage volume named `dbdata` on a mount point named `/mnt/dbdata` and make it available, execute the following command:

```
mount /mnt/dbdata
```

Repeat this step for each FlexVol volume to be used for the DB2 database.

3. To create a database on the storage system volumes, the database instance owner must have ownership of the mounted file systems. Change ownership for each mounted FlexVol volume used for the database by executing the following command on the database server:

```
chown -R db2inst1:db2adm [MountPoint]
```

For example, to grant ownership of a file system mounted on the mount point named `/mnt/dbdata` to the user named `db2inst1`, execute the following command:

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

7.2 IBM AIX

1. Modify the `/etc/filesystems` file found on the database server to add an entry for each storage system volume. Open the `/etc/filesystems` file with a text editor and add entries for each storage system volume. The entry should look similar to the following:

```
[MountPoint]:
dev      = [VolPath]
mount    = true
vfs      = nfs
nodename = netapp
options  = bg,nointr,rw,cio
type     = nfs_mount
account  = false
```

Where

- `MountPoint` is a directory path used to mount a FlexVol volume on a database server.
- `VolPath` is the path assigned to the FlexVol volume.

For example, to add an entry for a volume named `dbdata` mounted on a mount point named `/mnt/dbdata`, append the following lines to the `/etc/filesystem` file on the database server:

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

In this entry, `mount = true` makes a file system persistent across system reboots.

2. After adding the entries to the `/etc/filesystems` file, mount the FlexVol volumes by executing the following command:

```
mount [MountPoint]
```

For example, to mount a FlexVol volume on a mount point named `/mnt/dbdata`, execute the following command:

```
mount /mnt/dbdata
```

Repeat this step for each FlexVol volume used for the database.

3. To create a database on the storage system volumes, the database instance owner must have ownership of the mounted file systems. Change ownership for each mounted FlexVol volume by executing the following command:

```
chown -R db2inst1:db2adm [MountPoint]
```

For example, to change the ownership of a file system mounted on the mount point named `/mnt/dbdata` to the DB2 instance owner named `db2inst1`, execute the following command:

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

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

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

7.3 SUN SOLARIS

1. Append an entry for each FlexVol volume to be used for the database to the `/etc/vfstab` file on the database server. The entry should look similar to the following:

```
[StorageSystemName]:[VolName] [MountPoint] nfs - yes rw,bg,hard,nointr
,rsize=32768,wsiz=32768,{forcedirectio | llock},tcp,vers=3
```

Where

- `StorageSystemName` is the name assigned to the storage system.
- `VolName` is the name assigned to a FlexVol volume on the storage system.
- `MountPoint` is a directory path on the database server used to mount a FlexVol volume.

For example, in order to add an entry for a FlexVol volume named `dbdata` mounted on a mount point named `/mnt/dbdata`, add the following line to the `/etc/vfstab` file:

```
netapp:/vol/dbdata /mnt/dbdata nfs - yes rw,bg,hard,nointr,rsize=32768,
wsiz=32768,forcedirectio,tcp,vers=3
```

2. Mount and make the storage system volumes available by executing the following command:

```
mount [MountPoint]
```

For example, to mount a FlexVol volume named `dbdata` on a mount point named `/mnt/dbdata`, execute the following command:

```
mount /mnt/dbdata
```

Repeat this step for each FlexVol volume used for the DB2 database.

3. To create a database on the FlexVol volumes, the database instance owner should have ownership of the mounted volumes. Change ownership for each mounted file system to be used for the database, by executing the following command:

```
chown -R db2inst1:db2adm [MountPoint]
```

For example, to change the ownership of a FlexVol volume mounted on a mount point named `/mnt/dbdata` to the DB2 instance owner named `db2inst1`, execute the following command:

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

7.4 HP-UX

1. Append an entry for each FlexVol volume to be used for the database to the `/etc/fstab` file on the database server. The entry should look similar to the following one:

```
[StorageSystemName]:[VolName] [MountPoint] nfs rsize=32768,wsiz=32768,  
hard,nointr,rw,bg,vers=3,proto=tcp,noac,forcedirectio,timeo=600 0 0
```

Where

- `StorageSystemName` is the name assigned to the storage system.
- `VolName` is the name assigned to a FlexVol volume on the storage system.
- `MountPoint` is a directory path on the database server that used to mount a FlexVol volume.

For example, to add an entry for a FlexVol volume named `dbdata` mounted on a mount point named `/mnt/dbdata`, append the following line to the `/etc/fstab` file:

```
netapp:/vol/dbdata /mnt/dbdata nfs rsize=32768,wsiz=32768,hard,nointr,  
rw,bg,vers=3,proto=tcp,noac,forcedirectio,timeo=600 0 0
```

2. Mount the FlexVol volumes available by executing the following command:

```
mount [MountPoint]
```

For example, to mount a FlexVol volume named `dbdata` on a mount point named `/mnt/dbdata` and make it available, execute the following command:

```
mount /mnt/dbdata
```

Repeat this step for each FlexVol volume used for the DB2 database.

3. To create a database on the storage system volumes, the database instance owner must have ownership of the mounted volumes. Change ownership for each mounted FlexVol volume used for the database by executing the following command:

```
chown -R db2inst1:db2adm [MountPoint]
```

For example, to change the ownership of the FlexVol volume mounted on the mount point named `/mnt/dbdata` to the DB2 instance owner named `db2inst1`, execute the following command:

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

8 CONFIGURING THE DATABASE SERVER IN A SAN ENVIRONMENT

After creating initiator group (igroup) and LUN mappings, the database server must be configured to utilize the space available on the 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 to configure the most common UNIX hosts.

8.1 LINUX

To configure a Linux server in a SAN environment, follow these steps:

1. Set up the kernel parameters as described on page 12.
2. Configure a Linux host for FCP.
3. Configure a Linux host for iSCSI.
4. Make LUNs accessible on the database server.

CONFIGURE A LINUX HOST FOR FCP

1. Install the NetApp Host Attach Kit.

It is recommended that you download and install the SAN (FCP) Host Attach Kit for Linux provided by NetApp. This product simplifies the configuration of a Linux machine as the host component of a NetApp SAN environment. To install this product, complete the following steps:

- a. Log in to the [NOW](#) Web site using your username and password. Navigate to the SERVICES & SUPPORT page and select **Download Software**.
- b. On the Download Software page, go to the SAN (FCP) Host Attach Kit product row of the table, select **Linux** from the Select Platform drop-down list, and click the **GO** button.
- c. Follow the prompts to reach the Software Download page and download the NetApp software file to a local directory.
- d. Copy the downloaded Attach Kit tar file to a directory (`/tmp/netapp`) on the database server and extract the software by executing the following command:

```
tar -xvzf /tmp/netapp netapp_linux_tools_1_0.tar.gz
```

For example, to extract the tar file named `netapp_linux_tools_1_0.tar.gz`, execute the following command:

```
tar -xvzf /tmp/netapp/netapp_linux_tools_1_0.tar.gz
```

The extracted software is placed in the `/tmp/netapp/netapp_linux_tools_1_0` directory.

- e. Change the directory to `netapp_linux_tools_1_0`, where you have extracted the Attach Kit software, and execute the following command:

```
./install
```

- f. Add the following lines to `<home>/.bash_profile` file found 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, refer to *Installation and Setup Guide 1.0 for Fibre Channel Protocol on Linux* on the [NOW](#) Web site.

2. Install the HBA and driver.

Before you install the HBA on the database server, check product compatibility on the NOW Web site. The compatibility matrix is available on the Compatibility and Configuration Guide for NetApp FCP and iSCSI Products page. After confirming compatibility, install HBA and its driver on the database server. For the HBA and its driver installation steps, refer to the product's installation and configuration guide on the NOW Web site at

now.netapp.com/NOW/knowledge/docs/hba/fcp_linux/fcp_linux10/pdfs/install.pdf.

3. Obtain 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 storage system. The WWPN for the HBA installed on the database server can be obtained by completing the following steps:

- a. After installing the NetApp attach kit, the HBA, and the required driver, execute the following command to obtain the WWPN:

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

Upon execution with `-c` option, the `sanlun` command generates an `igroup create` command. The output of this command looks similar to the following:

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

- b. The WWPN for adapter in this example is `10000000c93a069b`. Execute this command to create igroup on the storage system.
- c. Refresh the FC adapter by executing the following commands:

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

CONFIGURE A LINUX HOST FOR ISCSI

1. Install the NetApp iSCSI Support Kit.

It is recommended that you download and install the iSCSI Host Utilities (Support Kit) for Linux provided by NetApp. This product simplifies the configuration of a Linux host in NetApp IP SAN environment. In order to install this product, complete the following basic steps:

- a. Log in to [NOW](#) Web site using your username and password. Navigate to the “SERVICES & SUPPORT” page and select **Download Software**.
- b. On the Download Software page, go to the iSCSI Host Utilities (Support Kit) product row of the table, select **Linux** from the select platform drop-down list, and click the **GO** button.
- c. Follow the prompts to reach the iSCSI Host Support Kits for Linux page. Select [iSCSI Red Hat Linux Host Utilities 3.1 for Native OS](#).
- d. Read the iSCSI Host utility description page and click **CONTINUE** at the bottom of the page.
- e. Read the cautions and update page and click the **CONTINUE** at the bottom of the page.
- f. Read and accept the license agreement by clicking **ACCEPT** at the bottom of the page. Download the utility tar file from iSCSI Host Utility 3.1 Download Page to a directory on your local machine.
- g. Copy the downloaded support kit file to a directory (`/tmp/netapp`) on the database server. Uncompress and extract it using the following command:

```
tar -xvzf [FileName]
```

Where

`FileName` is the name assigned to the downloaded tar file.

For example, to extract the tar file named `netapp_linux_host_utils_3_1.tar`, execute the following command:

```
tar -xvf /tmp/netapp/netapp_linux_host_utils_3_1.tar
```

- h. Change the directory to `netapp_linux_host_utils_3_1`, where the tar file is extracted and install the product by executing the following command:

```
./install
```

- i. Add the following lines to your `<home>/ .bash_profile` file found 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.

NetApp supports iSCSI Software Initiator for Linux. To install it, 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 and you can skip steps (a) through (e) in this section and start with step (f). Otherwise download the software from <http://sourceforge.net/projects/linux-iscsi/>.

- b. Create a work directory `/tmp/netapp` on the database server and ftp the downloaded initiator tar file to the work directory on the database server.

- c. Uncompress the tar file by executing the following command:

```
tar -xvzf [FileName]
```

Where

`FileName` is the name assigned to the downloaded tar file.

For example, to uncompress and extract the initiator tar file named `linux-iscsi-4.0.2.gz`, execute the following command:

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

- d. 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:

```
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` file.

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` file.

- e. Install the driver by executing the following command:

```
make install
```

- f. To make the initiator discover the LUN devices on the storage system, add the following lines to the iSCSI configuration file named `/etc/iscsi.conf` on the database server:

```
continuous=no
headerDigest=never
dataDigest=never
discoveryAddress=[StorageSystemIP]
```

Where

`StorageSystemIP` is the network IP address assigned to the storage system.

For example, to make LUNs discovered on two storage systems that have IP addresses `10.32.90.100` and `10.32.91.300`, add the following lines to the `/etc/iscsi.conf` file:

```
continuous=no
headerDigest=never
dataDigest=never
discoveryAddress=10.32.90.100
discoveryAddress=10.32.91.300
```

- g. Start the iSCSI service and load the driver by executing the following command:

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

Every time you make any LUN mapping change, the iSCSI driver needs to be refreshed by executing the following command:

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

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

- h. If you want to configure the iSCSI service to start automatically at system boot, enter the following command on the database server:

```
chkconfig iscsi on
```

3. Obtain WWNN for the initiator.

The iSCSI initiator has a unique identification known as Word Wide Node Name or WWNN. To access LUNs on the storage system, associate the initiator on the database server with an igroup on the storage system using WWNN. The Linux software initiator stores its WWNN information in the `/etc/initiatorname.iscsi` file. Obtain the WWNN for the initiator by executing the following:

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

MAKE LUNS ACCESSIBLE ON THE DATABASE SERVER

DB2 supports two types of tablespace containers: file system containers and raw devices containers. The following section describes the steps required to configure both types of containers.

1. Access LUNs as regular file system tablespace containers:

After the FCP or iSCSI driver is refreshed, the database server should be able to discover target LUNs from the storage system as new devices. 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:

```
sanlun lun show
```

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

```
fdisk [DeviceName]
```

Where

`DeviceName` is the name assigned to the LUN device on the database server.

For example, to create partition tables and format a device named `/dev/sdb`, execute the following command:

```
fdisk /dev/sdb
```

The `fdisk` command invokes the format wizard. Answer the series of questions to complete the formatting.

- c. After formatting, create a file system on each device by executing the following command:

```
mkfs [DeviceName]
```

For example, to create file system on the device named `/dev/sdb`, execute the following command:

```
mkfs /dev/sdb
```

- d. If the file system was not created as a Journal file system, add an ext3 journal to the file system by executing the following command:

```
tune2fs -j [DeviceName]
```

For example, to add ext3 journal to a device named `/dev/sdb`, execute the following command:

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

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

```
mkdir -p [MountPoint]
mount [MountPoint] [DeviceName]
```

Where

`MountPoint` is a directory path used to mount a FlexVol volume on a database server.

To mount a device named `/dev/sdb` on a mount point named `/mnt/dbdata`, execute the following command on the database server:

```
mount /mnt/dbdata /dev/sdb
```

- f. To create a database on the storage system volumes, the database instance owner must have appropriate permissions on all file systems to be used for the database. You can do so by changing the ownership of each mount file system to the database instance owner executing the following command:

```
chown -R db2inst1:db2adm [MountPoint]
```

For example, to change the ownership of the FlexVol volume mounted on the mount point named `/mnt/dbdata`, execute the following command on the database server:

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

- g. To make the file system mount persistent over system reboots, add an entry for each file system to the `/etc/fstab` file on the database server. The entry in the `/etc/fstab.iscsi` file should appear similar to the following:

```
[DeviceName] [MountPoint] [FSType] [MountOption] [BkupFreq] [FsckPass]
```

Where

- `FSType` is the file system type for the device.
- `BkupFreq` is the frequency of backup. This is used by `dump` to back up the file system. If the number is zero, then `dump` will ignore the backup for the file system.
- `FsckPass` is the order in which the file system integrity should be checked. If the value is set to 0, `fsck` will not check the file system.

For example, to make mount point persistent for a device named `/dev/sdb` mounted on `/mnt/dbdata`, add the following lines to the `/etc/fstab.iscsi` file on the database server:

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

2. Access 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. To use newly added LUN devices as raw device tablespace containers, complete the following steps:

- a. Convert the newly added LUN devices to character devices by executing the following command:

```
raw /dev/raw/raw[Number] [DeviceName]
```

Where

`DeviceName` is the name assigned to LUN device on the database server.

For example, to convert a block device named `/dev/sdd` to a character device, execute the following command:

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

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

```
[RawDevName] [BlockDeviceName]
```

Where

- `RawDeviceName` is the character device name assigned to LUN device.
- `BlockDeviceName` is the block device name assigned to LUN device.

For example, to add a raw device named `/dev/raw/raw1`, add the following lines to the `/etc/sysconfig/rawdevices` file:

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

- c. After updating the raw device configuration file, start the raw device daemon and reload the raw devices by executing the following command:

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

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

```
chmod 777 [RawDevName]
```

Where

`RawDeviceName` is the character device name assigned to LUN device.

For example, to set permission to 777 for a raw device named `/dev/raw/raw1`, execute the following command:

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

The raw device controller named `/dev/rawctl`, which acts as the central repository for raw block device binding information, might require an adjustment to the permissions setting. You can change the permissions for the controller by executing the following command:

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

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

```
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`, execute the following command:

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

8.2 IBM AIX

To configure an IBM-AIX server in a SAN environment, follow these steps:

1. Configure the AIX host for FCP.
2. Configure an AIX host for iSCSI.

CONFIGURING AIX HOST FOR FCP

1. Install the NetApp Host Attach Kit.

It is recommended that you download and install the SAN (FCP) Host Attach Kit for AIX provided by NetApp on the database server. This product simplifies the configuration and management of the AIX host in a NetApp SAN environment. To install this product, complete the following steps:

- a. Log in to [NOW](#) Web site using your username and password. Navigate to the SERVICES & SUPPORT page and select **Download Software**.
- b. On the Download Software page, go to the SAN (FCP) Host Attach Kit product row of the table, select **AIX** from the Select Platform drop-down list, and click the **GO** button.

- c. Follow the prompts to reach the FCP IBM AIX Attach Kit 1.2 Download page and download the Software tar file to a local directory.
- d. Copy the downloaded software tar file to a work directory (`/tmp/netapp`) on the database server and uncompress the file using the following command:

```
uncompress /tmp/netapp/ntap_aix_fcp_1.2.tar.Z
```

- e. Extract the file by executing the following command:

```
tar -xvf [FileName]
```

Where

`FileName` is the tar file name to be extracted.

For example, to extract a tar file named `ntap_aix_fcp_1.2.tar`, execute the following command:

```
tar -xvf /tmp/netapp/ntap_aix_fcp_1.2.tar
```

- f. Change the directory to `ntap_aix_fcp_1.2`, where you have extracted the attach kit software and execute the following command:

```
./install
```

Answer the prompt and complete the installation.

- g. Add the following lines to the `<home>/.profile` file in the user's home directory on the database server:

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

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

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

```
shutdown -Fr now
```

For detailed installation steps, refer to the *FCP IBM AIX Attach Kit 1.2 Installation and Setup Guide* on NOW Web site.

2. Install the HBA and driver.

Before you install HBA on the database server, check product compatibility on NOW Web site. The compatibility matrix is available on Compatibility and Configuration Guide for NetApp FCP and iSCSI Products page. After confirming compatibility, install the HBA and driver on the database server. For HBA and driver installation, refer to product's installation and configuration guide.

3. Obtain WWPN for the HBA.

Each FC HBA attached to the database server is uniquely identified by a WWPN. To create an igroup on the storage system, the WWPN for the HBA is required. To obtain the WWPN, complete these steps:

- a. After installing the NetApp Host Attach Kit for AIX, HBA, and drivers on the database server, find WWPN by executing the following command:

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

Upon execution with the `-c` option, the `sanlun` command generates an `igroup create` command which can be used to create an igroup on the storage system. The output looks similar to the following:

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

- b. The WWPN for adapter in this example is `10000000c93a069b`. Use the generated command to create an igroup.
- c. Load the FC driver by executing the following command:

```
cfgmgr -l fcs0
```

After the driver is loaded, the database server should be able to see the target LUNs on the storage system as new devices.

CONFIGURING AN AIX HOST FOR ISCSI

1. Install the NetApp iSCSI Host Attach Kit.

It is recommended that you install the iSCSI Host Support Kit for AIX provided by NetApp. This product simplifies the configuration and management of NetApp IP SAN environment and it can be installed by following these steps:

- a. Log in to the [NOW](#) Web site using your username and password. Navigate to the SERVICES & SUPPORT page and select **Download Software**.
- b. On the Download Software page, go to the iSCSI Host Support Kit product row of the table, select **AIX** from the Select Platform drop-down list, and click the **GO** button.
- c. Follow the prompts to reach the iSCSI IBM AIX Initiator Support Kit 1.1 Download Page and download the Support Kit tar file to a local directory.
- d. Copy the downloaded tar file to a work directory (`/tmp/netapp`) on the database server and uncompress it using the following command:

```
uncompress /tmp/netapp/netapp_aix_SAN_kit_1.1.tar.Z
```

- e. Extract the file by executing the following command at the database server:

```
tar -xvf [FileName]
```

Where

`FileName` is the tar file name to be extracted.

For example, to extract a file named `ntap_aix_SAN_kit_1.1.tar`, execute the following command:

```
tar -xvf /tmp/netapp/netapp_aix_SAN_kit_1.1.tar
```

This command will extract the support kit software in the `netapp_aix_SAN_kit_1.1` directory.

- f. Change the directory to `netapp_aix_SAN_kit_1.1`, where you have extracted the support kit software, and install the Support Kit by executing the following command:

```
./install -f -a /tmp/netapp/netapp_aix_SAN_kit_1.1
```

Answer the prompts and complete the installation.

- g. After installing the Support Kit software, add the following lines to the user's `<home>/ .profile` file in 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
```

- h. Reboot the system by executing the following command:

```
shutdown -Fr
```

For detailed installation steps, refer to *iSCSI IBM AIX Support Kit 1.1 Installation and Setup Guide* on NOW Web site.

2. Configure the AIX iSCSI Software Initiator.

NetApp supports iSCSI software initiator for AIX. The iSCSI initiator is included with the IBM AIX 5.2 operating system in the maintenance level 3 (ML3) release. If you are using a less recent version of DB2, then download and install iSCSI Initiator from IBM Web site. The iSCSI software initiator for AIX can be configured using SMIT:

- a. Start SMITTY by executing the following command:

```
smit iscsi
```

- b. Select **Devices > iSCSI Protocol Device**.
- c. Select **Change / Show Characteristics of an iSCSI Protocol Device**.
- d. Verify if the Initiator's WWNN is correct; it must be in a specific format:

```
iqn.yyyy-mm.backward_naming_authority.unique_device_name
i.e. - iqn.1992-08.com.ibm.0xa3cafcd
```

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

- e. Obtain the iSCSI target node name by executing the following command on the storage system:

```
iscsi nodename
```

- f. Define the storage system as an iSCSI target by adding a line to the `/etc/iscsi/targets` file on the database server. The entry in the `/etc/iscsi/targets` file should look like the following:

```
[StorageSystemIP] 3260 [StorageSystemNodename]
```

Where

- `StorageSystemIP` is the network IP address assigned to the storage system.
- `StorageSystemNodename` is the nodename assigned to the target adapter on the storage system.

For example, to add a storage system that has an IP `10.32.90.100`, as an iSCSI target for the initiator, add the following line to the `/etc/iscsi/targets` file:

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

3. Obtain the WWNN for the initiator.

- a. Obtain the WWNN for an initiator by executing the following command:

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

- b. Verify that the initiator's WWNN is correct. The Initiator's WWNN is required for creating an igroup and it must adhere to a naming format similar to the following:

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

i.e. - `iqn.1992-08.ibm650.hostid.0xa3cafcd`

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

```
chdev -l 'iscsi0' -a initiator_name=['NewWWNN']
```

Where

`NewWWNN` is the new WWNN assigned to the initiator.

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

```
chdev -l 'iscsi0' -a initiator_name='iqn.1992-08.com.ibm.0xa384509'
```

- c. Load the iSCSI driver by executing the following command:

```
cfgmgr -l iscsi0
```

4. Make LUNs accessible on the database server.

- a. After the initiator is refreshed, the database server should be able to discover the LUNs as new devices. View these newly added devices by executing the following command:

```
lsdev -Cc disk
```

- b. Get device names by executing the following command:

```
sanlun lun show
```

Upon execution, the `sanlun lun show` command shows the LUN devices along with assigned names. These devices can be used as tablespace containers for the DB2 database.

- c. Verify the attributes of the new disks by executing the following command:

```
lsattr -El [DeviceName]
```

Where

`DeviceName` is the name assigned to the LUN device on the database server.

For example, to verify the attribute of newly added device named `hdisk3`, execute the following command:

```
lsattr -El hdisk3
```

- d. Create a volume group for each LUN device by executing the following command:

```
mkvg -f -y '[VgName]' [DeviceName]
```

For example, to create a volume group named `datavg1` for the device named `hdisk3`, execute the following command:

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

Repeat this process until all necessary volume groups are created.

- e. Verify the volume groups by executing the following command:

```
lspv
```

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

```
varyoffvg [VgName]
```

```
exportvg [ VgName]
```

```
mkvg -q -f -y '[NewVgName]' [DeviceName]
```

Where

- `VgName` is the volume group name to be changed.
- `NewVgName` is the new volume group name.
- `DeviceName` is the name assigned to the LUN device on the database server.

For example, to change a volume group's name from `dblogsvg1` to `logsvg1`, execute the following commands:

```
varyoffvg dblogsvg1
```

```
exportvg dblogsvg1
```

```
mkvg -q -f -y 'logsvg1' hdisk4
```

- f. 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 [UserName]:[UserGroupName] [MountPoint]
```

Where

- `MountPoint` is the mount point name to be used to mount the file system.
- `UserName` is the name assigned to the database instance owner.
- `UserGroupName` is the name assigned to the db instance owner group.

For example, to create a mount point named `/mnt/dbdata` and change ownership to user name `db2inst1`, execute the following command:

```
mkdir -p /mnt/dbdata
```

```
chown =R db2inst1:db2adm /mnt/dbdata
```

- g. Create a new file system on the volume group by executing the following command:

```
crfs -v jfs2 -a bf=true -g '[VgName]' -a size=[FSsize] { M|G } -m  
'[MountPoint]' -p 'rw' -a options=cio -a nbpi=4096 -a ag=64
```

Where

- `VgName` is volume group name to be changed.
- `MountPoint` is the mount point name used to mount the file system.
- `FSSize` is the size of the file system to be created on the volume group.

For example, to create a file system on the volume group named `data1vg` that is mounted on a mount point named `/mnt/dbdata`, execute the following command:

```
crfs -v jfs2 -a bf=true -g 'data1vg' -a size=6000M -m '/mnt/dbdata1/' -p 'rw' -a options=cio -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 64GB in size.

- h. To make the file system mount persistent over system reboots, 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/lv11` persistent on database server reboot, modify the corresponding entry in the `/etc/filesystems` file and after update the entry should look similar to the following:

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

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

- i. 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 `dblogs` shown in the following example should be `/dev/r1v12`. This is the name that would be used to specify a raw device container for a tablespace.

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

- j. Test to make sure that you can write to the raw device by executing the following:

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

8.3 SUN SOLARIS

To configure a Sun Solaris server in a SAN environment:

1. Update the kernel parameter.
2. Configure the Sun Solaris host for FCP.
3. Configure the Sun Solaris host for iSCSI.
4. Make LUNs accessible on the database server.

UPDATE KERNEL PARAMETER

The default values for the kernel configuration parameters might not be good enough to run DB2 efficiently. For example, the default values for `semsys:seminfo_semume` and `shmsys:shminfo_shmseg` kernel parameters are not good enough for most of the multithreaded applications with a fair number of connections. DB2 provides a utility named `db2osconf` to make 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 the `-t` option to get the recommended values:

```
db2osconf -t [NumThreads]
```

Where

`NumThreads` is the count of threads to be run on the database server.

For example, execute the following command to get recommended kernel parameter values for 500 threads:

```
db2osconf -t 500
```

2. To set a kernel parameter value other than default one, add a line to the `/etc/system` file:

```
set [ParameterName]=[NewKernelValue]
```

Where

- `ParameterName` is the parameter name to be updated.
- `NewKernelValue` is the new kernel parameter value.

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

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

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

For detailed list of recommended kernel parameters, refer to the Preinstallation Task chapter of the DB2 manual, *Quick Beginning for DB2 Server*.

CONFIGURE SUN SOLARIS HOST FOR FCP

1. Install the NetApp SAN (FCP) Host Attach kit.

It is recommended that you install the SAN (FCP) Host Attach Kit for Solaris provided by NetApp on the database server. This product simplifies the configuration of the Solaris host in the NetApp SAN environment. To install this product, complete the following steps:

- a. Log in to [NOW](#) Web site using your username and password. Navigate to the SERVICES & SUPPORT page and select **Download Software**.
- b. On the Download Software page, go to the SAN (FCP) Host Attach Kit product row of the table, select **Solaris** from the Select Platform drop-down list, and click the **GO** button.
- c. Follow the prompts to reach the FCP Solaris Attach Kit 2.0 Download Page and download the software tar file to a local directory.
- d. Copy the downloaded software tar file to a work directory (`/tmp/netapp`) on the database server and uncompress the file by executing the following command:

```
uncompress /tmp/netapp/ntap_solaris_fcp_2_0.tar.Z
```

- e. Extract the file by executing the following command:

```
tar -xvf [FileName]
```

Where

`FileName` is the tar file name to be extracted.

For example, execute the following command to extract software tar file named `ntap_solaris_fcp_2_0.tar`:

```
tar -xvf /tmp/netapp/ntap_solaris_fcp_2_0.tar
```

- f. To install the Attach Kit software, change the directory to `ntap_solaris_fcp_2_0`, where you have extracted the software, and execute the following command:

```
./install
```

Answer the prompt and complete the installation.

For detailed installation steps, refer to *FCP Solaris Attach Kit 2.0 Installation and Setup Guide* on NOW Web site.

2. Install the HBA and driver.

Log in to NOW Web site and check the Compatibility and Configuration Guide for NetApp FCP and iSCSI Products for your database server product compatibility. NetApp supports Emulex FC HBA for Sun Solaris. Install the FC HBA and its driver on the database server. For installation steps, refer to *Host Bus Adapter Installation and Setup Guide 1.1 for Fibre Channel Protocol on Solaris* on the NOW Web site (<http://now.netapp.com/NOW/knowledge/docs/hba/sun/relhbas11/pdfs/setup.pdf>).

3. 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. For LUN mappings to take effect, augment the `/kernel/drv/sd.conf` file with additional entries. The entry should look similar to the following:

```
name="sd" parent="lpfc" target=[TargetID] lun=[LunID];
```

Where

- `TargetID` is the target group ID for the target adapter.
- `LunID` is an ID on the target group assigned for the LUN device.

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

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

4. 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:

```
reboot -- -r
```

5. After adding LUN devices, execute the following command to refresh the driver:

```
devfsadm
```

6. Obtain WWPN for the initiator.

Each FC HBA attached to your database server is uniquely identified by a WWPN. To create an igroup on the storage system, you need the WWPN for the HBA. After installing the FC adapter and driver, obtain the WWPN by executing the following `sanlun` command:

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

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

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

The WWPN for initiator in this example is `10000000c93a069b`.

CONFIGURING SUN SOLARIS HOST FOR ISCSI

1. Install the NetApp iSCSI Host Support Kit.

It is recommended that you install the iSCSI Host Support Kit for Solaris provided by NetApp. This product simplifies the configuration and management of the SAN environment. To install this product, complete the following steps.

- a. Log in to the [NOW](#) Web site using your username and password. Navigate to the SERVICES & SUPPORT page and select **Download Software**.
- b. On the Download Software page, go to the iSCSI Host Support Kit product row of the table, select **Solaris** from the Select Platform drop-down list, and click the **GO** button.
- c. Follow the prompts to reach the iSCSI QLogic Initiator Support Kit 1.2 Download page and download the software tar file to a local directory.
- d. 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_basic_iscsi_solaris_tools_1.0.tar.gz
```

- e. After uncompressing, extract the file by executing the following command:

```
tar -xvzf [FileName]
```

Where

`FileName` is the tar file name to be uncompressed and extracted.

For example, execute the following command to extract tar file named `ntap_basic_iscsi_solaris_tools_1.0.tar`:

```
tar -xvf /tmp/netapp/ntap_basic_iscsi_solaris_tools_1.0.tar
```

- f. Change to the `ntap_solaris_openiscsi_1.0N20040713_1731` directory where you extracted the software and execute the following command on the database server:

```
./install
```

2. Install the HBA and driver.

NetApp supports QLogic iSCSI HBA for iSCSI on Solaris. Download the iSCSI driver for the initiator and the SanSurfer iSCSI Command Line utility from the QLogic Web site (www.qlogic.com/products/iscsi_products_hba.asp). We have used QLogic SANsurfer iSCSI Command Line tools for configuring initiator. Log in to your database server as the user `root` and perform these 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/netapp`) on the database server and uncompress the files by executing the following command:

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

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

```
pkgrm QLA4010-0
```

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

- d. Install the driver by executing the following command:

```
pkgadd -d /tmp/netapp/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:

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

This completes the installation for an add-on HBA Driver and SANsurfer CLI tools.

- g. Shut down, power off the system, insert the QLogic HBA into an empty PCI slot on the database server, and connect the adapter to your iSCSI target storage system.
- h. Update the `/kernel/drv/sd.conf` file on your database server by adding the following line for each target LUN:

```
name="sd" class="scsi" class_prop="ataapi" target=[TargetID] lun=[LunID];
```

Where

- TargetID is the target ID.
- LunID is the ID on the target for a LUN device.

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

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

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

- i. Reboot the system by executing the following command:

```
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 igroup on the storage system you will need WWNN for the HBA. Find the WWNN by executing the following Qlogic SANsurfer iSCSI CLI command on the database server:

```
iscli -i
```

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

```
iqn.1992-08.sunv20z.sv112.0xa3cafcd
```

To manage and configure Qlogic HBA on Sun Solaris host in NetApp IP-SAN environment, you can also use 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 of the disk that you want to repartition.

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

For example, specify 7 in order to create a partition and format the 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. Type "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
```

- Modify the predefined partition table, reserve 1 cylinder at the head of the disk for the partition table, and use the all free hog option for slice 6 as follows:

```
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

- Create the new partition table by typing “yes” on the prompt and entering table name:

```
Okay to make this the current partition table[yes]? Yes
Enter table name (remember quotes): dbdata1
```

- Create a label and complete the formatting operation on the database server:

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

- Create the file system on the disk by executing the following command:

```
newfs -b [BlockSize] -i [NBytes] [DeviceName]
```

Where

- **BlockSize** is the block size for the file system, either 4096 or 8192 bytes per block.
- **NBytes** is 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 1GB	2048
Less than 2GB	4096
Less than 3GB	6144
3GB to 1 Tbyte	8192
Greater than 1 Tbyte	1048576

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

`DeviceName` is the name assigned to the LUN device on the database server.

For example, to create a file system on a device identified as `c3t0d0s6`, execute the following command:

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

8.4 HP-UX

To configure an HP-UX server in a SAN environment:

1. Set up the kernel parameters.
2. Configure the HP-UX host system for FCP.
3. Configure the HP-UX host system for iSCSI.
4. Make LUNs accessible on the database server.

SET UP THE KERNEL PARAMETERS

Before installing DB2, it is recommended 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 steps to modify the kernel parameter values:

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

```
db2osconf -t [NumThreads]
```

Where

`NumThreads` is the count of threads to be run on the database server.

For example, execute the following command to get recommended kernel parameter values for 500 threads:

```
db2osconf -t 500
```

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

```
Sam
```

3. Select **Kernel Configuration** from the list.
4. Select **Configurable Parameters** from the list.
5. Select the parameter that you want to change and press Tab.
6. Select **Action > Modify Configuration Parameter** from the main menu and enter the new value in the Formula/Value field.
7. Press Tab and go to **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. On the next panel, confirm your modification by selecting the `yes` option. This will update the kernel parameters and reboot your server automatically.

For detailed list of recommended kernel parameters, refer to the Preinstallation Task chapter in the DB2 manual, *Quick Beginning for DB2 Server*.

CONFIGURE HP-UX HOST SYSTEM FOR FCP

1. Install the NetApp SAN (FCP) Host Attach Kit.

It is recommended that you install the SAN (FCP) Host Attach Kit for HP-UX provided by NetApp. This product simplifies the configuration and management of the HP-UX host component of the NetApp SAN environment. To install this product, complete the following steps.

- a. Log in to [NOW](#) Web site using your username and password and select **Download Software** from SERVICES & SUPPORT page.
- b. On the Download Software page, go to the SAN (FCP) Host Attach Kit product row of the table, select **HP-UX** from the Select Platform drop-down list, and click the **GO** button.
- c. Follow the prompts to reach the FCP HP-UX Attach Kit 1.1 and ASL 3.5 Download page and download the software tar file to a local directory.
- d. 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_1.depot.gz
```

- e. Install the NetApp Host Attach Kit by executing the following command:

```
/usr/sbin/swinstall -s /tmp/netapp/ntap_hpux_fcp_1_1.depot
```

- f. Add the following lines to the `<home>/.profile` file in the user's home directory on the database server:

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

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

For detailed installation steps, refer to the installation guide *FCP HP-UX Attach Kit 1.1 Installation and Setup Guide* available on NOW Web site.

2. Install the HBA and driver.

- a. Log in to the NOW Web site and go to the Compatibility and Configuration Guide for NetApp FCP and iSCSI Products page (http://now.netapp.com/NOW/knowledge/docs/san/fcp_iscsi_config).
- 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 instructions described in HBA Installation and Configuration Guide to 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. To create an igroup on the storage system, you will need the WWPN for the HBA. After installing the HBA and NetApp Host Attach Kit for HP-UX, obtain 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. The output from this command looks similar to the following:

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

The WWPN for the initiator in this example is `50060b00002cf7c8`.

Use this command to create an igroup on the storage system.

- b. To create device nodes for the storage system LUNs, execute the following command:

```
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. The LUN devices are not automatically visible to the database server. To perform LUN discovery, execute the following command:

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

CONFIGURE HP-UX HOST SYSTEM FOR ISCSI

1. Install the NetApp iSCSI Host Support Kit.

It is recommended that you install the iSCSI Host Support Kit for HP-UX provided by NetApp. This product simplifies the configuration and management of HP-UX host component of NetApp IP SAN environment.

- a. Log in to the [NOW](#) Web site using your username and password and select **Download Software** from SERVICES & SUPPORT page.
- b. On the Download Software page, go to the iSCSI Host Support Kit product row of the table, select **HP-UX** from the Select Platform drop-down list, and click the **GO** button.
- c. Follow the prompts to reach the iSCSI HP-UX Initiator Support Kit 1.2 Download page and download the software tar file to a local directory.
- d. Copy the downloaded support kit file to a work directory (`/tmp/netapp`) on the database server and uncompress the file using the following command:

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

- e. Install the Support Kit by executing the following command:

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

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

- f. Add the following lines to the `.profile` file 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.

NetApp supports Software Initiator for HP-UX. To install the HP-UX iSCSI Software Initiator:

- a. Log in to NOW Web site and go to Compatibility and Configuration Guide for NetApp FCP and iSCSI Products page (http://now.netapp.com/NOW/knowledge/docs/san/fcp_iscsi_config).
Check 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>).
- c. Enter "iSCSI Software Initiator" in the search box. When the search results show iSCSI Software Initiator, click **Receive for Free**.
- d. Follow the prompts to reach the Software download page. Download the software initiator file and installation instructions to a local directory.
- e. Copy the downloaded initiator file to a work directory (`/tmp/netapp`) on the database server.
- f. Install the initiator by executing the following command:

```
swinstall -x autoreboot=true -s /tmp/netapp/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 installation is complete.

Add the following line to `.profile` file 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> for detailed installation and configuration steps.

3. Configure the initiator and obtain the WWNN.

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

```
/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 storage system. The initiator's WWNN must be in this specific format:

```
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:

```
iscsiutil /dev/iscsi -i -N [iSCSIInitiatorName]
```

Where

`iSCSIInitiatorName` is the name assigned to the iSCSI Initiator on the database server.

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:hpfc214.2000853943`, execute the following command:

```
iscsiutil /dev/iscsi -i -N iqn.1986-03.com.hp:hpfc214.2000853943
```

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

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

- c. Each iSCSI target used for the database needs to be identified by the database server persistently over system reboots. To make the target's storage system identification persistent, add the name or IP address of the storage system to the kernel configurations of the database server by executing the following command:

```
iscsiutil /dev/iscsi -a -I [StorageSystemIPORName] -P <TCPPort> -M <PortalGrpTag>
```

Where

- `StorageSystemIPORName` is the name or IP address assigned for the storage system.
- `TCPPort` is the TCP port number for the target. The default value is 3260.
- `PortalGrpTag` is the target portal group ID. The default value is 1.

For example, to make static identification for the storage system named `netapp`, execute the following command:

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

- d. To make the database server discover storage system LUNs, execute the following command:

```
ioscan -H 255
```

Check the newly discovered devices by executing the following command:

```
ioscan -funC disk
```

- e. To create device nodes for a storage system LUN device, execute the following command:

```
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 TO THE DATABASE SERVER

To create a database on the storage system LUN devices, create physical volumes, volume groups, and logical volumes on them. Complete these steps to create 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 `sanlun` command:

```
sanlun lun show -p [StorageSystemIPORName]:[LunPath]
```

Where

- StorageSystemIPORName is the name or IP address assigned to the storage system.
- LunPath is the assigned LUN path.

For example, to find a device name for the LUN named /vol/dbdata/data1 on a storage system named netapp, execute the following command:

```
sanlun lun show -p netapp:/vol/dbdata/data1
netapp:/vol/dbdata/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
```

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

2. Create a physical volume on the storage system LUN by specifying its character device name:

```
pvcreate [CharDeviceName]
```

Where

CharDeviceName is the character device name assigned to the LUN.

For example, to create a physical volume on the character device named /dev/rdisk/c4t0d0, execute the following command:

```
pvcreate /dev/rdisk/c4t0d0
```

3. Execute the following command 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
```

4. 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
```

5. 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/[VgName]
```

Where

VgName is volume group name the directory is created for.

For example, to create a device node directory for the volume group named vg01_dbdata, execute the following command:

```
mkdir -p /dev/vg01_dbdata
```

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/[VgName]/group c 64 ox[HexMinorDeviceNumber]0000
```

Where

- `VgName` is volume group name.
- `HexMinorDeviceNumber` is major device number in hexadecimal format.

For example, to create a device node for the volume group `vg01_dbdata`, execute the following command:

```
mknod /dev/vg01_dbdata/group c 64 0x${VGH}0000
```

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

6. Create a volume group by executing the following command:

```
vgcreate [VgName] /dev/dsk/[device], ...
```

Where

`VgName` is volume group name.

For example, to create a volume group named `vg01_dbdata`, execute the following command:

```
vgcreate vg01_dbdata /dev/dsk/c4t0d0, /dev/dsk/c4t0d1
```

A single volume group can have multiple devices (the device name for a LUN is obtained from step 2 of this procedure). Depending on the size of volume group you are creating, you need to specify an extent size to the `vgcreate` command. The total storage in the volume group can be divided into a maximum of 65535 extents. The extent size can be the power of 2 between 1MB and 256MB.

7. Create a logical volume on the volume group by executing the following command:

```
lvcreate -L size -n [VgName] [LVgName]
```

Where

- `LVgName` is the logical volume group name to be created.
- `VgName` is the volume group name that the logical volume group is created on.

For example, to create a 1000MB logical volume named `dbdata` within `vg01_dbdata` volume group, execute the following command:

```
lvcreate -L 1000 -n dbdata vg01_dbdata
```

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

8. To see the volume group detail, execute the following command:

```
vgdisplay -v vg01_dbdata
```

9. If database tablespace containers use raw devices, then skip this step and proceed with step 10.

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

```
newfs -F vxfs -o largefiles /dev/[VgName]/[LVgName]
```

Where

- `LVgName` is the logical volume group name.
- `VgName` is the volume group name the logical volume group is created on.

For example, to create a file system on the logical volume named `dbdata`, execute the following command:

```
newfs -F vxfs -o largefiles /dev/vg01_dbdata/dbdata
```

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

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

```
mkdir -p /mnt/[MountPoint]
```

```
chown -R db2inst1:db2adm [MountPoint]
```

Where

`MountPoint` is the mount point name to be used to mount file system.

For example, to create mount point named `/mnt/dbdata` and change its ownership to the database instance owner named `db2inst1`, execute the following command:

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

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

- c. Mount the file system using newly created mount points by executing the following command:

```
mount -F vxfs /dev/[VgName]/[LVgName] [MountPoint]
```

Where

- `LVgName` is the logical volume group name.
- `VgName` is the volume group name that the logical volume group is created on.

For example, to mount a file system to a mount point named `/mnt/dbdata`, execute the following command:

```
mount -F vxfs /dev/vg01_dbdata/dbdata /mnt/dbdata
```

- d. To make the file system mounting persistent on database server reboot, add an entry for each file system you have created to the `/etc/fstab` file on the database server similar to the following:

```
/dev/[VgName]/[LVgName] [MountPoint] vsfx delaylog 0 2
```

For example, add the following line to the `/etc/fstab` file to make mount persistent for the logical volume named `dbdata` on the mount point named `/mnt/dbdata`:

```
/dev/vg01_dbdata/dbdata1 /mnt/dbdata vsfx delaylog 0 2
```

10. If raw devices are desired as the tablespace containers, then reference the logical volume name prefixed with `r` as the storage container name. For example, the raw device name for the logical volume `/dev/vg01_dbdata/dbdata1` would be `/dev/vg01_dbdata/rdbdata1`. This is the name that would be used to specify the container for a raw DMS tablespace.

9 CREATING A NEW DATABASE ON NETAPP STORAGE

This section describes how to create a DB2 database whose data and transaction log files reside on NetApp storage. The creation of the DB2 database is very straightforward and utilizes standard DB2 commands. Complete the following steps to create a DB2 database:

1. Log in as the instance owner and install the DB2 on the database server. For installation instructions, refer to the DB2 manual, *Quick Beginnings for DB2 Servers*.
2. Create a new database on a FlexVol volume that is mounted to a mount point on the database server by executing the following command on the database server:

```
db2 "CREATE DATABASE [DatabaseName] on [MountPoint]"
```

Where

- `MountPoint` is the name assigned to the mount point used for the database.
- `DatabaseName` is the name of the database.

The mount point name must be explicitly specified in the `CREATE DATABASE` command.

For example, to create a database named `MYDB` on a FlexVol volume named `dbdata` that is mounted on a mount point named `/mnt/dbdata`, execute the following command:

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

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` is the path where the database is created.
- `InstanceName` is the name of the database instance.
- `NodeName` is the number of the database node.
- `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 volume on the storage system by updating the database registry parameter `NEWLOGPATH`. To do this, execute the following command:

```
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 `dblogs` that is mounted on a mount point named `/mnt/dblogs`, execute the following command:

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

Note: In most cases, changes to a database configuration file do not take effect until the database has been restarted after terminating all the database connections.

3. Verify that the database is created correctly and exists on the storage system by executing the following command:

```
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.

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

```
db2 "CONNECT TO [DatabaseName]"
```

Where

`DatabaseName` is the alias assigned to the database that was created. If no alias was specified, then the alias will be the same as the name assigned to the database.

For example, execute the following command to connect to the database named `mydb`:

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

5. List the tablespaces that were created with the database by executing the following command:

```
db2 "LIST TABLESPACES"
```

6. Verify that the tablespace containers associated with each tablespace refer to a file or directory located on a FlexVol volume by executing the following command:

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

Where

`TBSpaceID` is the unique identifier (0, 1, 2, and so on) that has been assigned to the tablespace that container information is to be retrieved for.

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, execute the following command:

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

7. Additional tablespaces can be created on a FlexVol volume using the `CREATE TABLESPACE` command. For details on the command syntax, refer to IBM DB2 SQL Reference Vol. 2.

10 MIGRATING A DB2 DATABASE TO NETAPP STORAGE

The migration of DB2 databases to a FlexVol volume is also straightforward and utilizes standard DB2 commands. This section describes how to convert a 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 NetApp storage.

The items described in “Architecture” on page 3 are required in order for this process to work correctly.

IMPORTANT: Always make sure that you have fully recoverable backup copies of your database and related files before attempting to migrate an existing database to a storage system volume.

10.1 MIGRATING A DB2 DATABASE USING THE REDIRECTED RESTORE OPERATION

The easiest way to 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 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 after a backup image of the database has been obtained.

Complete the following steps to perform this type of DB2 database migration:

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

```
db2 "CONNECT TO [DatabaseName]"
```

Where

`DatabaseName` is the database alias or name.

For example, execute the following command 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. To make the change take effect, stop and restart the DB2 Database Manager instance 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:

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

Where

- `DatabaseName` is the database alias or name.
- `BKUPLocation` is the location to store the database backup image.

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

```
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 [OldDatabaseName] from [BKUPLocation] to [NewLocation]
into [NewDatabaseName] redirect"
db2 "set tablespace containers for [TBSpaceID] using ('[Directory]')"
or
db2 "set tablespace containers for [TBSpaceID] using (FILE '[FileName]'
[Size])"
or
db2 "set tablespace containers for [TBSpaceID] using (DEVICE '[DeviceName]'
[SIZE])"
db2 "restore database [OldDatabaseName] continue"
```

Where

- OldDatabaseName is the alias assigned to the database that was backed up.
- BKUPLocation is the location of the database backup image.
- NewLocation is the location where the overhead files associated with the database will reside.
- NewDatabaseName is the name that will be assigned to the new database to be created.
- TBSpaceID is the unique identifier (0, 1, 2, and so on) that has been assigned to the tablespace that new container information will be assigned for.
- Directory is the new directory SMS tablespace data will be stored in.
- FileName is the name of the file that DMS tablespace data will be stored in.
- DeviceName is the name of the device used to store tablespace data.
- Size is the size, in 4K, 8K, 16K, or 32K pages, of the file or device specified.

For example, to restore a database named `mydb` from a backup location named `/dbbkup/mydb` and move the tablespace containers to a FlexVol volume, your script should look similar to the following:

```
DumpDir=/dbbkup/mydb
NewDBDIR=/mnt/dbdata/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.

6. Execute the script file just created to perform a redirected restore operation.
7. Verify that the database is created correctly and exists on the storage system by executing the following command:

```
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.

10.2 MIGRATING A DB2 DATABASE USING DB2LOOK AND DB2MOVE

Since 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 want to change the tablespace type (for example, to switch from DMS tablespaces to SMS tablespaces).

To do this, you must create a new DB2 database that uses the desired tablespace types, and then copy data from the existing database 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 a storage system.

The `db2move` utility can be run in one of three modes: `EXPORT`, `IMPORT`, or `LOAD`.

- **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 might produce one or more message files that contain warning or error messages that were generated as a result of an export operation.
- **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. In this mode, the file `db2move.lst` establishes a link between the PC/IXF-formatted files needed and the tables into which data will be imported.
- **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` establishes 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, then you must find another way to migrate those objects as well. You can do this with the DB2 utility `db2look`, which analyzes an existing database and produces a set of Data Definition Language (DDL) SQL statements that can then recreate all of the objects found in the database analyzed.

Complete the following steps to perform this type of DB2 database migration:

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 [DatabaseName] -e -o [OutFile]
```

Where

- `DatabaseName` is the database alias or name that will be migrated.
- `OutFile` is the name and location of the file where the DDL information produced will be written.

For example, to create DDL for the object database named `mydb`, execute the following command:

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

This command will place the `mydb.ddl` file in the directory `/export/home/db2ins81`.

3. Edit the file produced in the previous step to delete the first `CONNECT` statement found, and then save the file.
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 [DatabaseName] EXPORT
```

Where

`DatabaseName` is the alias assigned to the database to be migrated.

For example, to migrate a database named `mydb`, execute the following command:

```
db2move mydb export
```

5. Create a new database on the storage system volumes as described in “Creating a New Database on NetApp Storage” on page 38.

Note: If you want to assign the name of the database being migrated to the database being created, then you must first drop or uncatalog the database being migrated. If you uncatalog the database being migrated, then you can always recatalog it.

6. Establish a connection to the database created in the previous step 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 is created correctly and exists on the storage system by executing the following command:

```
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.

11 BEST PRACTICES

11.1 DESIGNING A DATABASE WITH RECOVERY IN MIND

Recovery needs must be considered from the start. To make it simple, you should always store system tables, user data, temporary data, transaction logs, and archive logs on separate volumes on the storage system. This data separation will simplify the use of Snapshot copies and typically make the recovery process much easier.

This allows the database to be backed up on a different schedule from the logs and 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, then the recovery of the volume would overwrite the active and archive logs, 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.

As part of the database design process, you should also identify any relationships that exist between the various database objects. These relationships can be at the 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, group related database objects together on the same logical FlexVol volume or group of volumes. This grouping helps DBAs to fine tune the performance and balance the workload. A documented model of the database design can be a tremendous benefit when making these decisions.

Note: The root volume should be its own volume. No database file should be stored on the root volume.

11.2 PERFORMANCE

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. IBM has done an excellent job in automating many of the registry and configuration parameters. The registry variable `DB2_PARALLEL_IO` has a significant impact on the performance.

11.3 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, sometimes it would be beneficial to have parallel I/O enabled for single container tablespaces. For example, if the container is created on a FlexVol volume or qtree, performance might 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 will 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
```

The first command will turn parallel I/O on for all tablespaces; the second command will turn parallel I/O on only for tablespaces 1, 2, 4, and 8.

11.4 DB2 DATABASE STORAGE CONSIDERATIONS

One of the most important factors to consider is how the database tablespaces are mapped to the I/O subsystem layer or storage layer. Poor database storage design can have significant impact on database performance. Making sure to have the adequate number of spindles as well as configuring the tablespace parameters is a key and is based on the throughput requirement. Follow these considerations when you design your database tablespace:

DECIDE THE NUMBER OF SPINDLES FOR YOUR DATABASE

Even though the storage world is getting more and more virtualized, the reality is that data is still stored in mechanical disk drives or spindles. These spindles are not accessible by the database servers directly and are connected through the storage controllers. The number of spindles used by the database has a significant importance when you have a database with an I/O-intensive workload.

NetApp storage virtualizes the disks using FlexVol volumes and aggregates that offer better performance. All the disk spindles in an aggregate are available to all the FlexVol volumes or LUNs. A study performed by NetApp engineering shows that customers can improve performance by creating aggregates. NetApp offers the Performance Acceleration Module (PAM), which is a new way to optimize the performance of a NetApp storage system by improving the throughput and latency while using fewer spindles. This saves power and eases cooling and rack space requirements.

USE FILE SYSTEMS INSTEAD OF RAW DEVICES

The need for raw devices is fully eliminated with the introduction of direct and concurrent I/O for performance. File systems provide superior manageability to raw devices, as a single file system can be used as a container for multiple tablespaces.

EXTENTSIZE. `EXTENTSIZE` specifies the number of `PAGESIZE` pages that will be written to a container before skipping to the next container and is defined at tablespace creation time. When you create your tablespaces, the `EXTENTSIZE` should be set to a number that leads to efficient big-block I/O performed by prefetchers. On NetApp storage, an aggregate consists of RAID groups and disks are tied to a RAID group. The RAID group and disk segment play important roles in deciding extent size. NetApp's disk segment size is one Tetrus which is 256KB (Tetrus size is 32 4KB blocks). When deciding extent size, a good rule of thumb is having an extent size of 256KB or its multiple.

For example, for a tablespace that has its containers on FlexVol volumes on an aggregate that `EXTENTSIZE` uses RAID-DP, and the RAID group size is 18 disks, the extent size can be multiple of 1024KB (4 x 256KB).

PREFETCHSIZE. The DB2 prefetching technique is similar to NetApp's read ahead feature; it allows DB2 Database Manager to fetch data pages it thinks an application is about to need into the DB2 buffer pools

before requests for the data are actually made. Thus, the prefetch size of a tablespace identifies the number of pages of table data that will be read in advance of the pages currently being referenced by a query in anticipation that they will be needed to resolve the query.

The overall objective of prefetching is to reduce query response time. This can be achieved if page prefetching occurs asynchronously to query execution. The DBA should define a pre-fetch value large enough to allow parallel use of all available containers. EXTENTSIZE of the PREFETCHSIZE can be configured as 1024*4, which is 4096KB. As general rule, the prefetch size should be a multiple of disk strip size. With a RAID group size of 18 disks, the disk strip size is 16*256 = 4096KB.

The prefetch size can be changed after a tablespace has been created, but extent sizes cannot be changed. Therefore, it is important that the appropriate extent size for a tablespace be determined before data is written to a tablespace.

NO FILE SYSTEM CACHING. The FILE SYSTEM CACHING and NO FILE SYSTEM CACHING tablespace parameters specify whether I/O operations should be cached at the file system level. Caching at the file system level and in the buffer pools causes performance degradation because of the extra CPU cycles required for the double caching.

To avoid this double caching, most file systems have a feature that disables caching at the file system level. This is generically referred to as nonbuffered I/O. On UNIX, this feature is commonly known as Direct I/O (or DIO). Some file systems, such as IBM JFS2 or Symantec™ Veritas™ VxFS, support enhanced Direct I/O, called the higher-performing Concurrent I/O (CIO) feature. The database manager supports this feature with the NO FILE SYSTEM CACHING tablespace clause. When this is set, the database manager automatically takes advantage of CIO on file systems where this feature exists.

Since the database manager manages its own data caching using buffer pools, caching at the file system level is not needed if the size of the buffer pool is tuned appropriately. This feature might help to reduce the memory requirements of the file system cache, thus making more memory available for other uses. NetApp recommends using the tablespace option to turn file system caching on or off.

Overhead and TransferRate. These values are used to determine the cost of I/O during query optimization. Both values are measured in milliseconds, and they should be the average for all containers.

The **overhead** provides an estimate of the time that is required by the storage container before any data is read into memory. Overhead can be calculated using the following formula:

$$Overhead = AverageSeekTime + \frac{1}{2} * RotationalLatency$$

$$RotationalLatency = \frac{1}{rpm} * 60 * 1000$$

For example, overhead for a tablespace container on a 15,000 rotations-per-minute Fibre Channel disk with an average seek time of 2 ms would be:

$$Overhead = 2 + \frac{1}{2} * \frac{1}{15000} * 60 * 1000 = 4ms =$$

The **transfer rate** is the amount of time necessary to read one page into memory. The default value for transfer rate is 0.9 ms. You can calculate transfer rate using the following formula:

$$TransferRate = \frac{1}{DiskTransferRate} * Spindles * 1000 * \frac{1}{1024000} * TablespacePageSize$$

You can find the disk the transfer rate from the hardware specification of the disk. The spindle count would be data disks in an aggregate.

11.5 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 that might not be appropriate for your environment. To obtain good performance, you might need to change these parameter values. If your experience with DB2 is limited, 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.

To invoke the Configuration Advisor from the DB2 Control Center:

1. Highlight the database you want to tune from the list of databases.
2. Right-click to display the database action menu.
3. Select **Configuration Advisor**.

11.6 ADDITIONAL FACTORS THAT AFFECT PERFORMANCE

In order to achieve point-of-failure recovery, make sure that the database's active and archive log files are accessible and up-to-date. The first step 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 FlexVol volume named `dbdata` and the database's transaction log files were stored on the FlexVol volume named `dblogs`.)

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.

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

```
db2empfa [DatabaseName]
```

Where

`DatabaseName` is the database alias or name.

For example, to enable multipage file allocation for the database named `mydb`, execute the following command:

```
db2empfa mydb
```

12 ENABLING NVFAIL ON THE STORAGE SYSTEM FOR ADDITIONAL DATA PROTECTION

When storing database data on a storage system, you should enable the `nvfail` feature for the volumes used for the database. This is a Data ONTAP operating system software feature that provides support for special error processing and is enabled by executing the following command on the storage system:

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

Where

`VolName` is the name of the FlexVol volume that the `nvfail` feature will be enabled for.

For example, to enable `nvfail` for the volume named `dbdata`, execute the following command:

```
vol options dbdata nvfail on
```

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

- Put the volumes and LUNs within volumes offline that had the NVRAM failure enabled.
- Stop exporting LUNs over FCP.
- Send 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.)

For additional protection, you can use a second feature that renames certain files that the system administrator or DBA might not want to be made accessible to the network until after they have been carefully examined. 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 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, if the following files exist on the storage system:

```
db2filer:/vol/dbdata/db2inst1/NODE0000/SQL0001/SQLSPCS.1
db2filer:/vol/dbdata/db2inst1/NODE0000/SQL0001/SQLSPCS.2
```

And the file `/etc/nvfail_rename` exists and looks like this:

```
/vol/dbdata/db2inst1/NODE0000/SQL0001/SQLSPCS.1
/vol/dbdata/db2inst1/NODE0000/SQL0001/SQLSPCS.2
```

And if a NVRAM failure has occurred, then the files listed in `/etc/nvfail_rename` will be renamed to:

```
db2filer:/vol/dbdata/db2inst1/NODE0000/SQL0001/SQLSPCS.1.nvfail
db2filer:/vol/dbdata/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. After 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 restarting the DB2 Database Manager.

Note: To make sure that a DBA becomes aware of an NVRAM failure regardless of when any user attempts to access any database stored on a storage system, you might 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.

13 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 might differ from that presented here. If you have any problems with the techniques shown in this document, contact the [author](#).

14 IMPORTANT LINKS

- Sourceforge page for Linux iSCSI driver download
<http://sourceforge.net/projects/linux-iscsi>
- Qlogic HBA driver download page
www.qlogic.com/products/iscsi_products_hba.asp
- Qlogic HBA documentation
www.qlogic.com/support/product_resources.asp?id=341
- Emulex support Web site for driver/firmware download:
www.emulex.com/ts/indexemu.html
- DB2 information center
<http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/index.jsp?topic=/com.ibm.db2.luw.admin.cm.d.doc/doc/r0001933.html>

- HP - Fibre Channel driver software downloaded page
www.hp.com/go/softwaredepot
- HP Home page for Fibre Channel adapter documentation
<http://docs.hp.com/en/netcom.html>
- HP-UX iSCSI Software Initiator Support Guide: HP-UX 11i v1 and 11i v2
www.hp.com/go/softwaredepot
- NetApp Documentation
 - Software download page on NOW
<http://now.netapp.com/NOW/cgi-bin/software>
 - FCP Solaris Attach Kit 2.0 Installation and Setup Guide
<http://now.netapp.com/NOW/knowledge/docs/hba/sun/relhbas20/pdfs/setup.pdf>
 - SAN Host Attach Kit 1.0 for Fibre Channel Protocol on HP-UX Installation and Setup Guide
http://now.netapp.com/NOW/knowledge/docs/hba/fcp_hp-ux/relhp-ux10/html/index.shtml
 - NetApp FCP Quick Reference Commands-Data ONTAP 6.5 and HP-UX 11i
http://now.netapp.com/NOW/knowledge/docs/san/fcp_iscsi_config/QuickRef/FCPHP-UX65QuickRef.pdf
 - NetApp Fibre Channel Configuration Guide
http://now.netapp.com/NOW/knowledge/docs/san/fcp_iscsi_config/QuickRef/FCConfigurationGuide.pdf
 - Block Access Management Guide for FCP
<http://now.netapp.com/NOW/knowledge/docs/ontap/rel70rc/pdfs/ontap/bsagfcp.pdf>
 - The Compatibility and Configuration Guide for NetApp FCP and iSCSI Products
http://now.netapp.com/NOW/knowledge/docs/san/fcp_iscsi_config/index.shtml
 - Configuring NetApp storage systems with iSCSI Initiators
<http://now.netapp.com/knowledge/docs/hba/iscsi/setup.pdf>
 - iSCSI IBM AIX Support Kit 1.1 Installation and Setup Guide
http://now.netapp.com/NOW/knowledge/docs/hba/iscsi/aix/iscsi_aix_1.1/reliscsiaix11/pdfs/setup.pdf

REVISION HISTORY

Date	Author	Comments
March 2006	Jawahar Lal and Roger Sanders	Original draft
Dec 2009	Bobby Oommen	Updated Data ONTAP, DB2, Controller Type, OS Version and Template.
May 2010	Bobby Oommen	Major updates to include architecture diagram and best practices.

NetApp provides no representations or warranties regarding the accuracy, reliability or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information in this document is distributed AS IS, and the use of this information or the implementation of any recommendations or techniques herein is a customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. This document and the information contained herein may be used solely in connection with the NetApp products discussed in this document.

