



Technical Report

# Using Integrated Snapshot Copy Backup Feature of IBM DB2 9.5 with NetApp Storage System

Jimmy Stamkos, Aslam Nomani, Dale McInnis, IBM  
Bobby Oommen, Ashwath Narayan, NetApp

September 2012 | TR-3668

## Executive Summary

This technical report describes how to meet challenges posed by shrinking backup and recovery windows using the integrated storage Snapshot™ features of DB2 9.5. Regardless of the size of the database or the level of activity on the storage system, a Snapshot copy is created within a few seconds. The database restore is based on NetApp SnapRestore®, and it dramatically improves the system availability by providing the ability to recover within minutes, instead of hours or days. Snapshot copy backup and restore are made possible by the integrated DB2 Advanced Copy Service, which allows creating Snapshot copies on the underlying storage. The Tivoli Storage Manager takes this functionality to the next level and offers database clone or instantaneous copy of the database and tape backup with a few simple and easy steps.

## TABLE OF CONTENTS

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Overview</b>	<b>4</b>
2.1	NetApp Snapshot Technology	4
2.2	Key Advantages of NetApp Storage System	5
2.3	Overview of NetApp Storage Layers	6
2.4	Overview of DB2 Advanced Copy Services	10
<b>3</b>	<b>Purpose and Scope</b>	<b>11</b>
<b>4</b>	<b>Requirements and Assumptions</b>	<b>11</b>
4.1	General Assumptions	11
4.2	Provide Remote Shell (RSH) Access for Storage System (Optional)	11
<b>5</b>	<b>DB2 ACS Installation and Configuration</b>	<b>12</b>
5.1	Hardware and Software Requirements	12
5.2	Set Up DB2 ACS Service	13
5.3	Configure ACS	13
5.4	Start and Stop ACS	14
5.5	Change DB2 ACS Configuration	15
<b>6</b>	<b>Create a New Database on NetApp Storage System</b>	<b>15</b>
<b>7</b>	<b>Backup and Restore</b>	<b>16</b>
7.1	DB2 Backup Using Integrated Snapshot Technology	16
7.2	Create a Snapshot Copy Backup for Nonrecoverable Database	17
7.3	Create a Snapshot Copy Backup for Recoverable Database	18
7.4	DB2 Restore Using Snapshot Technology	19
<b>8</b>	<b>Database Clone and Tape Backup Using Tivoli Storage Manager</b>	<b>23</b>
8.1	Database Clone	23
8.2	Mount Snapshot Backup Image on Target Server	25
8.3	Tape Backup from Clone	26
<b>9</b>	<b>Limitations</b>	<b>26</b>
<b>10</b>	<b>Conclusion</b>	<b>27</b>
	<b>Appendixes</b>	<b>27</b>
	Appendix A: ACS Backup Flow	27
	Appendix B: Configure More Than One Instance on a Machine	28
	Appendix C: Delete Backups with db2acsutil	28

Appendix D: DB2 ACS Setup Process .....	29
Appendix E: TSM4ACS Mount Process .....	30
<b>References.....</b>	<b>30</b>
<b>Version History .....</b>	<b>30</b>

**LIST OF TABLES**

Table 1) Backup options for recoverable and nonrecoverable database (PROD). .....	19
Table 2) Restore options for recoverable and nonrecoverable database (PROD). .....	23

**LIST OF FIGURES**

Figure 1) Logical relationship between aggregate, volumes, and disks. ....	7
Figure 2) DB2 9.5 backup process flow with Snapshot copy options. ....	17
Figure 3) DB2 database clone and backup to tape using Snapshot backup images. ....	24

# 1 Introduction

Nowadays, businesses have operations across the globe, and they are required to keep their mission-critical applications running 24/7. They also expect the application performance to be maintained during routine maintenance such as backup and recovery, regardless of the data growth rate, which can sometimes be very high. Since backup windows are shrinking and the amount of data that needs to be backed up is ever increasing, it is a complex task to define the time at which a backup can be created with minimal impact on system performance and availability.

Traditional backup and recovery methods pose several challenges:

- **System performance impact.** Database backup operations typically have a significant impact on the performance of a production system because they place high load on the database server, the storage system, and the underlying network during the backup process.
- **Shrinking backup windows.** Enterprises expect maximum availability of their applications. Defining an appropriate window for creating backup images can pose a challenge when a database needs to be accessible 24/7.
- **Rapid data growth.** Enterprise data is growing exponentially. That means longer windows are needed to back up the data. Organizations that can no longer afford an extended window for backups are forced to perform heavy investment in the backup infrastructure to keep the backup window short. Growing databases also require more tape media or disk storage space for backup images. Incremental backups can address these issues, but longer restore times make them unacceptable.
- **Decreasing mean time to recover (MTTR).** The mean time to recover is the time needed to recover from a database failure.

The MTTR can be divided in two parts:

- Time that is necessary to restore the database from a backup image.
- Time that is necessary to perform roll-forward recovery of the database.

The roll-forward recovery time depends on the number of archives and active logs that need to be reapplied to the database after it has been restored. A Snapshot copy is a local image of data.

Integration of NetApp® backup and restore technology with DB2 using DB2 Advance Copy Services (ACS) offers a unique way to address the challenges posed by traditional backup and recovery methods. DB2 ACS comes as an integral part of the DB2 9.5 server software. IBM Tivoli Storage Manager (TSM) takes advantage of this integration and delivers an instantaneous copy of database (alias database clone) and tape backup with a few easy steps as described in this report.

## 2 Overview

### 2.1 NetApp Snapshot Technology

NetApp Snapshot™ copies are a feature of the WAFL® (Write Anywhere File Layout) storage virtualization technology that is part of NetApp Data ONTAP®, the microkernel that ships with every NetApp storage system. A Snapshot copy is a locally retained point-in-time “frozen” image of a WAFL volume that provides easy access to old versions of files, directory hierarchies, and/or logical unit numbers (LUNs). The high performance of NetApp Snapshot copies makes them highly scalable. A Snapshot copy takes only a few seconds to create, regardless of the size of the volume or the level of activity on the NetApp storage system. After a Snapshot copy has been created, changes to data objects are reflected in updates to the current version of the objects, as if the Snapshot copy did not exist. Meanwhile, the Snapshot versions of the data remain completely frozen. Therefore, Snapshot copies incur no performance overhead; users can comfortably store up to 255 Snapshot copies per volume on the storage system, all of which can be accessible as read-only, online versions of the data.

DB2 ACS uses NetApp Snapshot technology to create an online or offline database backup. The time needed to create a Snapshot copy is independent of the size of the database because a Snapshot copy does not move data blocks. Snapshot copies vastly improve the frequency and reliability of backups, because they incur virtually no performance overhead and can be safely created while a database is up and running. Customers running a DB2 database on a NetApp storage system can take advantage of the integrated Snapshot technology. In addition, they can create backups throughout the day.

DB2 ACS not only enables database backup but also allows quick database restore using integrated SnapRestore technology. The NetApp SnapRestore feature of NetApp Data ONTAP provides a method to restore the entire database or parts of the database to the state it was in at the point when any available Snapshot copy was created. Because no copying of data is involved, an incredible amount of time is saved as the file system is returned to its earlier state. The restore process can be performed within a few minutes, independent of the size of the database. In addition, when low-impact Snapshot copy backups have been created frequently throughout the day, a fewer transaction logs need to be reapplied as part of the recovery process, resulting in a dramatic reduction in recovery time. In other words, the MTTR, which consists of the time required for restore and recovery, is reduced dramatically to several minutes, compared with several hours with conventional backup methods.

The DB2 ACS Snapshot copy backups are stored on the same NetApp storage system as the database. Therefore, NetApp recommends using Snapshot copy backups as a supplement, not a replacement, for backups to a second location, whether backing up to disk or to tape. Although backups to a second location are still necessary, there is only a slight probability that these backups will be needed for restore and recovery. Most restore and recovery actions can be handled by database restore using Snapshot copies. Restores from a second location (disk or tape) are necessary only in a situation where the primary storage system holding the Snapshot copies is damaged or there is a need to restore a database from a backup that is no longer available in the form of a Snapshot copy—for instance, a two-month-old backup.

For more information on the Snapshot and SnapRestore technologies, visit the NetApp [Support](#) site. Also, see "File System Design for an NFS File Server Appliance"<sup>1</sup> by Dave Hitz, James Lau, and Michael Malcolm and the Data Protection Online Backup and Recovery Guide<sup>2</sup>.

## 2.2 Key Advantages of NetApp Storage System

The core of the NetApp approach to database backup and recovery is the use of Snapshot technology that offers various features including stability, high performance, and storage efficiency.

The key advantages of NetApp Snapshot technology considering backup and recovery are:

- **Fast backup.** To meet the challenge posed by shrinking backup windows, the Snapshot copy feature of Data ONTAP is extremely useful. A Snapshot copy of a database can be created in seconds, regardless of the size of the database or the level of activity on the NetApp storage system. This dramatically reduces the database backup window from hours to seconds and helps DBAs to schedule frequent low-impact database backups.
- **Quick recovery.** Using the Data ONTAP SnapRestore command, an entire database can be restored in seconds from a Snapshot copy backup. Because no data copying is involved, a substantial amount of time is saved when a database is returned to the state it was in at the time the Snapshot copy was created. Additionally, because Snapshot copies can be created quickly and a large number of Snapshot copies can be retained, the amount of time required to perform a roll-forward recovery operation against a database can be greatly reduced.

---

<sup>1</sup> [http://media.netapp.com/documents/wp\\_3002.pdf](http://media.netapp.com/documents/wp_3002.pdf).

<sup>2</sup> <http://now.netapp.com/NOW/knowledge/docs/ontap/rel724/html/ontap/onlinebk/index.htm>

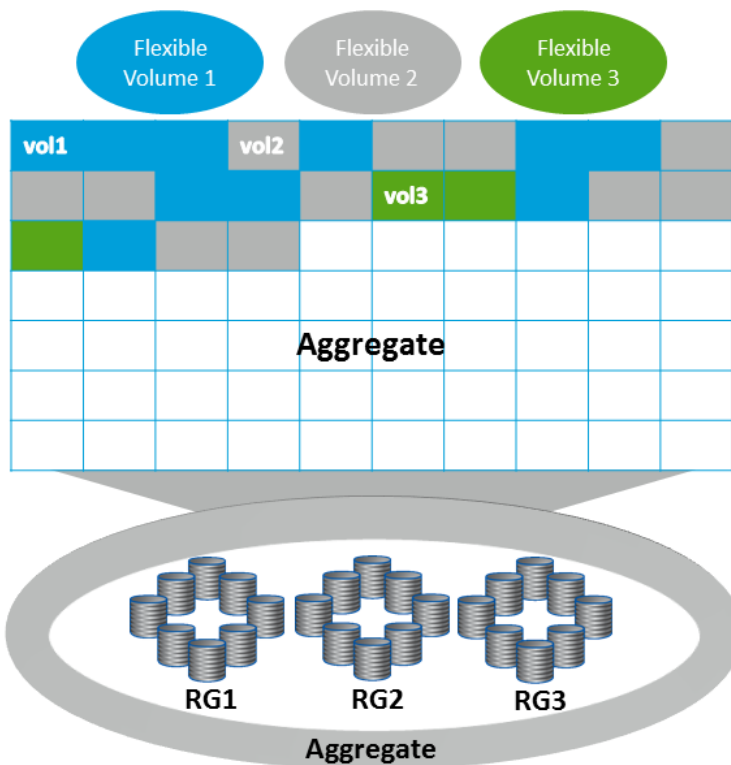
- **High availability.** The need for 24/7 availability is fast becoming a reality for organizations of all sizes. Companies can neither tolerate scheduled downtime nor afford extended periods of slow system response that is often caused by traditional database backup methods. Snapshot copies, in contrast, can be created within seconds without any impact on system response time. This makes sure of high availability and uninterrupted system response.
- **High reliability.** The RAID architecture used for NetApp storage systems is unique and provides greater reliability than direct-attached storage. If a RAID member disk fails, it is automatically reconstructed (using parity disk data) without any user intervention. NetApp supports single parity as well as RAID Double Parity, known as RAID-DP<sup>®</sup>. RAID-DP is considered approximately 10,000 times more reliable than traditional RAID. For details on RAID-DP, refer to [TR-3298: RAID-DP: NetApp Implementation of Double-Parity RAID for Data Protection](#).
- **Uninterrupted system response.** Because a Snapshot copy is only a “frozen” image of the file system at a specific point in time, the process of creating a database backup with a Snapshot copy does not require the actual copying of data; therefore, it has virtually no impact on system response time.
- **Minimum storage requirement.** Two Snapshot copies created in a sequence differ from one another by the blocks added or changed in the time interval between the two. This block-incremental behavior minimizes the amount of storage space consumed. The database clone uses the Snapshot copy backup images and shares unchanged data blocks using Snapshot technology, and hence it does not require any additional storage when created. The cloning feature delivers an ultimate storage savings. The database clone only needs storage for changed or new data blocks. For more information on database clone, refer to [TR-3460: Cloning a Database Using NetApp FlexClone Technology.](#)

## 2.3 Overview of NetApp Storage Layers

Data ONTAP operating in 7-Mode and higher versions supports aggregates, flexible volumes, and LUNs as logical storage layers. An aggregate is a RAID-level physical pool of storage and possesses its own RAID configuration, plex structure, and set of assigned disks. Within each aggregate, you can create one or more flexible volumes, the logical file systems that share the physical storage resources, RAID configuration, and plex structure of that common containing aggregate.

Figure 1 illustrates the logical relationship between aggregate, volume, and disks.

Figure 1) Logical relationship between aggregate, volumes, and disks.



An aggregate can be created by executing the following command on a NetApp storage system:

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

Where:

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

**Note:**

- Parameters shown in angle brackets (< >) are optional.
- Parameters or options shown in square brackets ([ ]) are required mandatorily.
- A comma followed by ellipses (...) indicates that the preceding parameter can be repeated multiple times.

For example, to create an aggregate named `dbaggr1` that has 10 disks, each 72GB in size, you would execute the following command on the NetApp storage system:

```
aggr create dbaggr1 -f -t raid_dp -r 10 10@72
```

A NetApp FlexVol<sup>®</sup> volume, also known as a flexible volume, is a logical storage container. A FlexVol volume resides inside an aggregate and delivers optimum performance by using all of the disk spindles available to the 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 by executing the following command on a NetApp storage system:

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

Where:

- `VolName` identifies the name assigned to the FlexVol volume that is being created.
- `AggrName` identifies the name assigned to the aggregate.
- `VolSize` identifies the size of the volume in terms of MB, GB, TB, and so on.

For example, to create a FlexVol volume named `dbdata` that is 10GB in size and resides in an aggregate named `dbaggr1`, you would execute the following command on the NetApp storage system:

```
vol create dbdata dbaggr1 10G
```

NetApp Snapshot technology works at the volume level; therefore, when planning your database's physical layout, if Snapshot technology will be incorporated into your backup and recovery strategy, make sure that the database's data and transaction log files are stored on separate volumes on the NetApp storage system. If a database recovery operation becomes necessary, maintaining separate volumes for data and logs will enable you to easily restore the database files from the appropriate Snapshot copy backup and roll forward using the database's transaction logs. NetApp strongly recommends having each database and database partition on separate volumes. It is also recommended that when determining the size for volumes; consider that Snapshot copies will also use storage space on that same volume. By default for a FlexVol volume, 20% space is reserved for Snapshot copies.

By default, a FlexVol volume has a Snapshot copy schedule enabled. The Snapshot copies created using the default schedule are not application coordinated. Hence, NetApp recommends turning off the automatic Snapshot copy schedule for all the FlexVol volumes that are used for your database.

The automatic Snapshot copy schedule can be turned off by executing the following command:

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

Where:

- `VolName` identifies the name assigned to the FlexVol volume that is being used for DB2 database.

For example, to turn automatic Snapshot copy off for a FlexVol volume named `dbdata`, you would execute the following command on the NetApp storage system:

```
vol options dbdata nosnap on
```

By default, the security style for a FlexVol volume is set to UNIX<sup>®</sup>. If not, then you can set the security style by executing the following command:

```
qtree security [VolPath] unix
```

Where:

- `VolPath` identifies the path assigned to the FlexVol volume that is being used for the DB2 database.

For example, to set security style UNIX for a FlexVol volume named `dbdata` with full path `/vol/dbdata`, you would execute the following command on the NetApp storage system:

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

After completing preceding steps, you need to export the volume to the production AIX or Linux<sup>®</sup> host by executing the following command:

```
exportfs -p rw=[DbServerName],root=[DbServerName],,sec=sys [VolName]
```



Where:

- `DbServerName` identifies the name of the host that is being used for DB2 database.
- `VolName` identifies the name assigned to the FlexVol volume that is being used for DB2 database.

For example, to export a FlexVol volume named `dbdata` to a host named `db2srv1`, you would execute the following command on the NetApp storage system:

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

After completing the preceding steps, you need to mount the volumes onto the AIX or Linux host. For Linux host, append an entry for each FlexVol volume to the `/etc/fstab` file. The entry should be similar to the following:

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

Where:

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

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

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

For an AIX host you need to add the entry to the `/etc/filesystems` file. The entry should be similar to the following:

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

Where:

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

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

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

You need to add an entry for each FlexVol volume used for the database. After adding the entries to the `/etc/fstab` file for Linux host or `/etc/filesystems` for AIX host, you need to mount and make the storage system volumes available by executing the following command on the database server:

```
mount [MountPoint]
```

Where:

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

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

```
mount /mnt/dbdata
```

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

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

Where:

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

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

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

For more information on configuring NetApp FAS or IBM N-series storage systems for DB2 database, refer to "Integrating IBM DB2 with the IBM System Storage N series."<sup>3</sup>

## 2.4 Overview of DB2 Advanced Copy Services

In a traditional backup or restore operation, the database manager copies data to or from disk or a storage device using operating system calls. DB2 Advanced Copy Services (ACS) enables the use of Snapshot or fast copying technology of a storage device to perform the data copying part of backup and restore operations. Being able to use the storage device to perform the data copying makes the backup and restore operations extremely fast.

In addition to Snapshot copy backup and restore, DB2 ACS offers the following key features:

- Multipartition DB2 UDB database support
- Centralized configuration through the configuration wizard
- Multiple Snapshot copy backups
- Policy-based management of Snapshot copy backups
- To perform Snapshot copy backup and restore operations, a DB2 ACS API driver is required. Integrated into IBM Data Server is a DB2 ACS API driver for the NetApp storage system. For supportability of other storage hardware, refer to IBM Web site page "DB2 Advanced Copy Services (ACS) supported operating systems and hardware."<sup>4</sup>

---

<sup>3</sup> [www.redbooks.ibm.com/abstracts/sg247329.html?Open](http://www.redbooks.ibm.com/abstracts/sg247329.html?Open).

<sup>4</sup> <http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/index.jsp?topic=/com.ibm.db2.luw.admin.ha.doc/doc/r0052876.html>.

## 3 Purpose and Scope

This document describes how to configure and use DB2 9.5 integrated Snapshot and SnapRestore technologies to back up and restore a DB2 database.

Specifically, this report covers the following topics:

- Infrastructure required to integrate DB2 9.5 Enterprise Server Edition with a NetApp storage system
- Creating a backup of a DB2 9.5 database using integrated Snapshot technology

## 4 Requirements and Assumptions

### 4.1 General Assumptions

It is assumed that you are familiar with DB2 9.5 Enterprise Server Edition and its backup and recovery mechanisms. It is also assumed that you are familiar with the operation of a NetApp storage system and its basic Data ONTAP commands. Finally, it is assumed that you have basic knowledge of UNIX used in your environment.

For our test environment, we installed DB2 9.5 on a UNIX host and created a database whose tablespace containers were located on NetApp storage system FlexVol volumes.

The following environment assumptions were made:

- The name of the UNIX host used for the DB2 database is `db2srv1`.
- The name of the DB2 administrator account is `db2inst1`.
- The name of the DB2 9.5 instance is `db2inst1`.
- The name of the database is `prod`.
- The name of the NetApp storage system is `ntap1`.
- The name of the aggregate on the NetApp storage system is `dbaggr1`.
- The database is created on a FlexVol volume named `dbsys`.
- The database's table data is stored on a FlexVol volume named `dbdata`.
- The database's transaction logs are stored on a FlexVol volume named `dblogs`.

The steps and scripts outlined in this document might require significant modifications to run under your database/host UNIX environment.

### 4.2 Provide Remote Shell (RSH) Access for Storage System (Optional)

Remote shell access allows secure access to the NetApp storage system from a remote host. If you plan to execute Data ONTAP commands from the database server, then you must enable remote shell access to the NetApp storage system by completing the following steps:

First, enable the RSH access feature by executing the following command on the NetApp storage system:

```
options rsh.enable on
```

Add an entry in the `/etc/hosts.equiv` file found on the NetApp storage system for the database host and the user. Entries in the `/etc/hosts.equiv` file should look similar to the following:

```
[HostName] [UserName]
```

Where:

- `HostName` identifies the name of the database host from where the `rsh` command is executed.

- `UserName` identifies the name of the user who wants to execute the `rsh` command.

For example, to create an entry for a database server named `db2srv1` and a user named `db2inst1`, you would append the `/etc/hosts.equiv` file with the following on the NetApp storage system:

```
db2srv1 db2inst1
```

After the appropriate entry has been added to the `/etc/hosts.equiv` file, a user can execute Data ONTAP commands directly from the host. The format used to execute Data ONTAP commands from the host is similar to the following:

```
rsh [StorageSystemName] -l [UserName]:[Password] [DOTCommand]
```

Where:

- `UserName` identifies the name of the user who wants to execute the `rsh` command.
- `Password` identifies the password of the user who wants to execute the `rsh` command.
- `StorageSystemName` identifies the name of the storage system.
- `DOTCommand` identifies the Data ONTAP command.

For example, to obtain the RSH option setting on a NetApp storage system that has an entry for the database server in its `/etc/hosts.equiv` file, you would execute the following command on the database server:

```
rsh ntap1 -l root:prodnal options rsh
```

**Note:** If no console password has been set, RSH access is allowed, even if there is no entry in the `hosts.equiv` file.

## 5 DB2 ACS Installation and Configuration

To use the integrated Snapshot feature of DB2 9.5 for database backup and restore, your environment needs to meet the following requirements.

### 5.1 Hardware and Software Requirements

The DB2 9.5 Snapshot backup feature is supported with certain storage hardware. Before you begin configuring your environment, make sure your environment meets the requirements listed on the Web page "DB2 Advanced Copy Services (ACS) supported operating systems and hardware."<sup>5</sup> In addition to this, the minimum version of Data ONTAP required for NetApp storage systems is 7.1.

To make DB2 ACS work successfully, you need to make sure that all the libraries and other components are installed. Also, make sure that `libgcc` and `openssl` are installed on your host. After installing `libgcc` on AIX, make sure you create a link `libgcc` with ACS by the following command:

```
ln -s [libgccPath] /usr/lib/libgcc_s.a
```

Where:

- `libgccPath` identifies the path where `libgcc` is installed.

---

<sup>5</sup><http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/index.jsp?topic=/com.ibm.db2.luw.admin.ha.doc/doc/r0052876.html>.

For example, to create a link for libgcc version 4 on AIX 5.3 that is installed on /opt/freeware/lib/gcc/powerpc-ibm-aix5.3.0/4.0.0/ path, you would execute the following command on the AIX server:

```
/opt/freeware/lib/gcc/powerpc-ibm-aix5.3.0/4.0.0/libgcc_s.a /usr/lib/libgcc_s.a
```

Similarly, install openssl library, if it does not exist. Create link by executing the following command:

```
ln -s /opt/ibm/db2/V9.5/lib64/icc/osslib/libcrypto.so.0.9.7a /usr/lib/libcrypto.so.0.9.7
ln -s /opt/ibm/db2/V9.5/lib64/icc/osslib/libssl.so.0.9.7a /usr/lib/libssl.so.0.9.7
```

## 5.2 Set Up DB2 ACS Service

To set up DB2 ACS, perform the following steps:

1. Define a TCP port: DB2 ACS is a service and communicates through a TCP port. A port number can be defined by adding an entry to /etc/services file.

For example, to add a port number 57328 for db2acs service, you need to add the following line to /etc/services file on the host:

```
db2acs      57328/tcp
```

2. After defining the port for ACS service, log in as user root and install the ACS service for the DB2 instance by running setup.sh:

```
./<DB2 Install directory>/acs/setup.sh -a <action> -d <DB2 ACS Directory> -u <DB2 Instance id> -g <DB2 Instance Group>
```

For example, to enable DB2 ACS service for the DB2 instance named db2inst1, execute the following command on the database server:

```
/home/db2inst1/sqllib/acs/setup.sh -a enable -d /home/db2inst1/sqllib -u db2inst1 -g db2admin
```

If the installation fails, investigate the output for errors and rerun the installation after fixing the errors.

## 5.3 Configure ACS

As the DB2 instance user, run the setup.sh wizard, located in /home/db2inst1/sqllib/acs/, to configure ACS.

The following example is a step-by-step walkthrough of setup.sh:

1. Enter whether the TSM for ACS license exists:

Do you have a full TSM license to enable all features of TSM for ACS?[y/n]

```
n
```

2. Specify the location for the ACS directory:

```
ACS_DIR [/home/db2inst1/sqllib/acs ] /home/db2inst1/sqllib/acs
ACS_DIR is the path to which ACS is installed. By default, this should be the
[/home/db2inst1/sqllib/acs directory.
```

3. Specify the host name and port number on this host:

```
ACSD [localhost 57328 ] db2srv1 57328
```

ACSD is the master server's host name and port. This port will be reserved for ACS and should match the number that was configured in the /etc/services file in step 3 of section 5, "DB2 ACS Installation and Configuration." If only a single database partition exists, localhost may be used as the host name. To configure multiple instances on hosts, refer to Appendix B.

4. Specify whether the ACS trace should be turned on:

```
TRACE [NO]
```

5. Specify the location of the ACS repository:

```
ACS_REPOSITORY *mandatory parameter* /home/db2inst1/acsrepository
```

ACS\_REPOSITORY is the directory on the db2 server that will keep a record of all backups created. The ACS repository should be placed on a shared path between all machines running the db2 instance. This path, however, cannot be a part of any of the volumes that will be part of the backup, whether it is a log or data volume.

6. Specify the TSM backup option:

```
TSM_BACKUP [NO]
```

7. Specify the number of versions of the database to keep:

```
MAX_VERSIONS [ADAPTIVE] 2
```

8. MAX\_VERSIONS is the maximum number of Snapshot copy backups that can be created. If no license key was applied, it can be set to a maximum of two. If the license key was applied and MAX\_VERSIONS is set to ADAPTIVE, then an unlimited number of backups can be created. Specify that the OS should use freeze and thaw functionality:

```
LVM_FREEZE_THAW [YES]
```

9. Specify the device class as STANDARD for NetApp devices:

```
DEVICE_CLASS [STANDARD]
```

10. Specify the hardware type as NAS\_NSERIES to use NetApp NFS:

```
COPYSERVICES_HARDWARE_TYPE *mandatory parameter* NAS_NSERIES
```

11. Specify the network host name of the NetApp storage system:

```
COPYSERVICES_PRIMARY_SERVERNAME [localhost ] ntap1
```

12. Specify a user on the NetApp storage system that has root access:

```
COPYSERVICES_USERNAME [superuser ] root
```

The user specified here must have the privileges on the NetApp storage system to perform Snapshot copy backup and restore operations.

13. Specify the appropriate passwords after the profile is created:

The profile has been successfully created.

Do you want to continue by specifying passwords for the defined devices? [y/n]

```
y
STANDARD
Master
```

The STANDARD password is the password to the account on the NetApp storage system for COPYSERVICES\_USERNAME. The master password is the password used for communication for the ACS services and may be set to any value.

After successful execution of setup.sh, entries will be added to /etc/inittab for the ACS process agents.

## 5.4 Start and Stop ACS

DB2 ACS does not start and stop together with DB2. To start and stop the DB2 ACS agents, the start and stop ACS commands must be explicitly issued from the ACS directory that is located under the subdirectory sqllib of DB2 instance home.

To start or stop the ACS service, issue the following command on the server:

```
setup.sh -a [start|stop] -d [DB2ACSDirectory] -u [InstanceUserIDName] -g [InstanceGroupName]
```

Where:

- `DB2ACSDirectory` identifies the directory name where DB2 ACS is installed.
- `InstanceUserIDName` identifies the db2 instance owner name.
- `InstanceGroupName` identifies the name of primary group for db2instance owner user.

For example, to start the ACS under db2inst1 instance, execute:

```
/home/db2inst1/sqllib/acs/setup.sh -a start -d /home/db2inst1/sqllib -u db2inst1 -g db2admin
```

Similarly, to stop the DB2 ACS service for a DB2 instance named db2inst1, execute:

```
/home/db2inst1/sqllib/acs/setup.sh -a stop -d /home/db2inst1/sqllib -u db2inst1 -g db2admin
```

**Note:** Make sure that ACS agents are started on every physical node prior to performing a backup.

## 5.5 Change DB2 ACS Configuration

To change an ACS configuration parameter that was originally set, stop the ACS service and run the setup.sh script again.

## 6 Create a New Database on NetApp Storage System

Creation and configuration of a database are simple with DB2. You can create a database by executing the following command on the database server:

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

Where:

- `MountPoint` identifies the name assigned to the mount point to be used for the database.
- `DatabaseName` identifies the name of the database to be created.

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

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

```
db2 create db prod on /dbdata
```

Note that in a DPF configuration, the actual data is stored in `/dbdata/db2inst1/NODE0000`, and `/dbdata/db2inst1/NODE0001`. These paths are the mount points for the NetApp volumes. In a nonpartitioned environment, `/dbdata/db2inst1` is the mount point.

By default, when a database is first created, circular logging is used, and primary log files are created in the subdirectory `[DbPath]/[InstanceName]/[NodeName]/SQLnnnn/SQLLOGDIR/`.

Where:

- `DbPath` identifies the path where the database is created.
- `InstanceName` identifies the name of the database instance.
- `NodeName` identifies the number of the database node.
- `SQLnnnn` identifies 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. You can do so by updating the database registry parameter named NEWLOGPATH. To update this parameter, you would execute the following command on the database server:

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

Where:

- `DatabaseName` identifies the name of the database to be created.
- `NewLogLocation` identifies the name assigned to the mount point to be used for the database transaction logs.

For example, to move transaction logs for a database named `prod` to another volume named `dblogs` that is mounted on a mount point named `/mnt/dblogs`, you would execute the following command on the database server:

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

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

Finally, if the database is required to be recoverable, enable archive logging. You can enable archive logging by executing the following command:

```
db2 "UPDATE DB CFG FOR [DatabaseName] USING logarchmeth1 disk:[ArchLogLocation]"
```

For example, to enable archive logging for a database named `prod` and store archive logs on another volume named `dbarchive` that is mounted on a mount point named `/mnt/dbarchive`, you would execute the following command on the database server:

```
db2 update db cfg for prod using logarchmeth1 disk:/dbarchive
```

## 7 Backup and Restore

### 7.1 DB2 Backup Using Integrated Snapshot Technology

DB2 ACS integration with DB2 9.5 enabled a very powerful feature of DB2 that allows DB2 to take advantage of Snapshot copy capabilities of the underlying storage for its backup and restore operations. DB2 9.5 allows the invocation of NetApp Snapshot copy-based backup and restore operations from the native DB2 backup and restore commands. The integrated NetApp Data ONTAP SDK libraries enable DB2 to discover underlying storage volumes on the NetApp storage used by the database and establish volume-level relationships. During the backup process, the following operations are performed:

1. User invokes native DB2 backup command.
2. DB2 uses Advance Copy Service (ACS) for HW to identify volumes relationship.
3. TSM provides database container list.
4. DB2 suspends writes and requests Snapshot copy.
5. TSM for HW creates Snapshot copy.
6. DB2 resumes writes.
7. Removes old backups, if necessary.

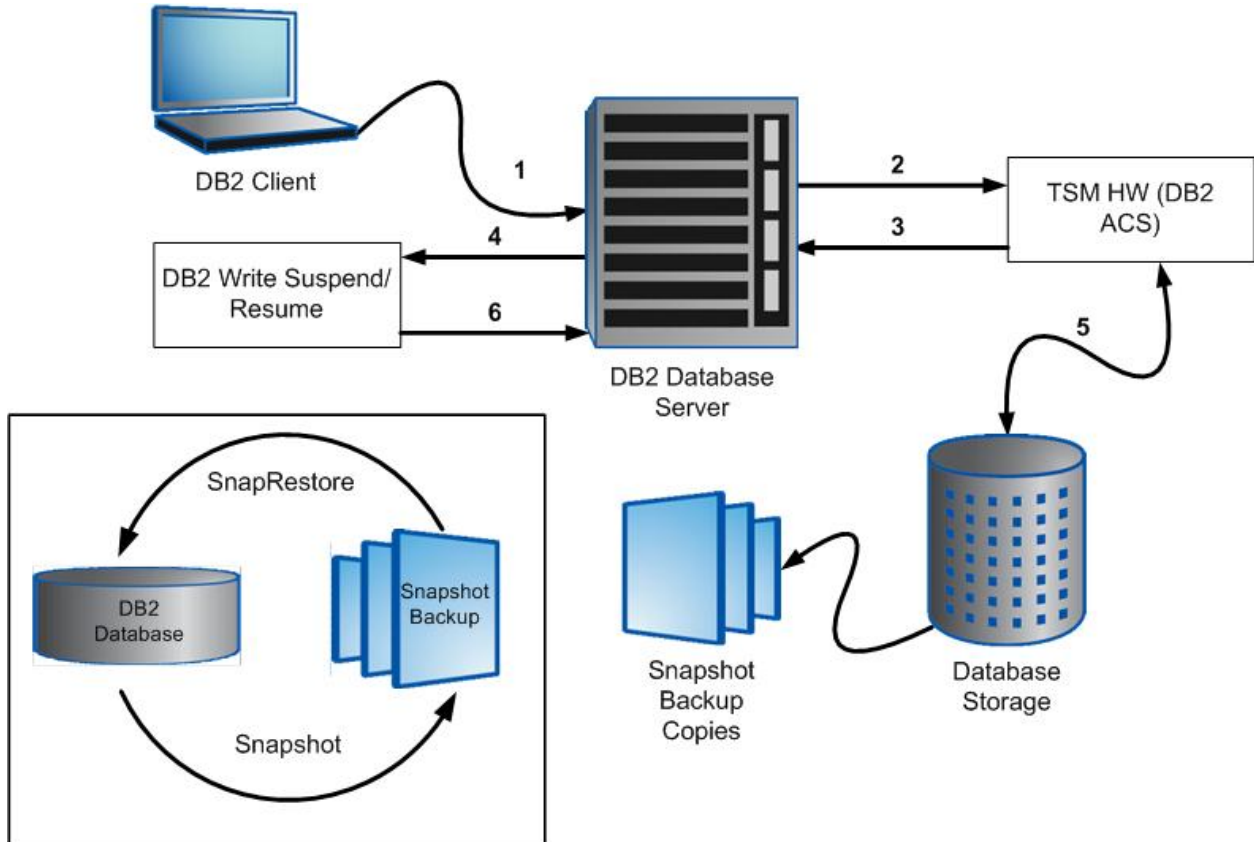
The Snapshot copy backups are allowed for single-node and multinode DB2 databases. For a multinode or DPF database it is recommended to use the `ON ALL DBPARTITIONNUMS` option when invoking the backup command. The `ON ALL DBPARTITIONNUMS` option allows DB2 to suspend and resume I/O on the entire database, and the database backup will be created across all of the database partitions at the same point in time.



Database backup steps vary based on the recoverability type of the created database. If archive logging is enabled, then the database is recoverable; otherwise, it is nonrecoverable. A nonrecoverable database backup can be created only after bringing the database offline. Roll-forward recovery is not permitted for a nonrecoverable database. For recoverable databases, backups may be done as online or offline. After database restore, roll-forward recovery is required to recover a database to a given point in time.

Figure 2 illustrates Snapshot copy backup process flow for DB2 9.5 with Snapshot copy options.

Figure 2) DB2 9.5 backup process flow with Snapshot copy options.



The following subsection describes the steps required for creating a backup for both nonrecoverable and recoverable databases.

## 7.2 Create a Snapshot Copy Backup for Nonrecoverable Database

To back up a single-partition nonrecoverable database, execute the following commands on the database server with appropriate privilege:

```
db2 deactivate database [DatabaseName]
db2 backup db [DatabaseName] use snapshot
```

Where:

- `DatabaseName` identifies the name of the database to be created.

For example, to create a Snapshot copy backup of a nonrecoverable database named `prod`, you would execute the following command on the database server:

```
db2 deactivate db prod
DB20000I The DEACTIVATE DATABASE command completed successfully.
```

```
db2 backup db prod use snapshot
Backup successful. The timestamp for this backup image is : 20080526175525
```

To back up a DPF database, you need to execute the following command:

```
db2_all "db2 deactivate database [DatabaseName]"
db2 backup db [DatabaseName] on all nodes use snapshot
```

Where:

- DatabaseName identifies the name of the database to be created.

For example, to create a Snapshot copy backup for a DPF database named prod and second host db2srv2, you need to execute the following commands from the server:

```
db2_all "db2 deactivate db prod"
DB20000I The DEACTIVATE DATABASE command completed successfully.
db2srv1: db2 deactivate db ... completed ok
SQL1496W Deactivate database is successful, but the database was not
activated.
db2srv2: db2 deactivate db ... completed rc=2

db2 backup db prod on all nodes use snapshot
Part Result
-----
0000 DB20000I The BACKUP DATABASE command completed successfully.
0001 DB20000I The BACKUP DATABASE command completed successfully.
Backup successful. The timestamp for this backup image is :
```

### 7.3 Create a Snapshot Copy Backup for Recoverable Database

Snapshot copy backups can be created for a recoverable database when the database is online or offline. Offline backups are performed in the same way for recoverable and nonrecoverable databases. To create an online Snapshot copy backup, execute the following command from the database server:

```
db2 backup db [DatabaseName] <on all nodes> online use snapshot
```

Where:

- DatabaseName identifies the name of the database to be created.

For example, to create an online Snapshot copy backup of a DPF database named prod, you would execute the following backup command on the database server:

```
db2 backup db prod on all nodes online use snapshot
```

Whether recoverable or not, once the Snapshot copy backup is complete, db2 will drive cleanup of old backups to make sure that the previously configured maximum number of backups (MAX\_VERSIONS) still holds true. To see example ACS log files for a backup, see Appendixes

Appendix A.

DB2 9.5 provides a utility named db2acsutil to manage Snapshot copy backups. This utility can be used to list, view the status, and delete the backup copies. To view Snapshot copy backups for all databases and their status, execute the following command with the query option:

```
db2acsutil query status
```

For example, execute the following db2acsutil command:

```
db2acsutil query status
Instance Database Part Image Time Status
=====
db2inst1 PROD 0 20080521092749 Repetitively restorable + Destructively restorable
```

To delete unwanted backups through db2acsutil, refer to Appendix C.

The online and offline backup processes for recoverable and nonrecoverable databases can be summarized in a few easy steps, as illustrated in Table 1.

**Table 1) Backup options for recoverable and nonrecoverable database (PROD).**

Database Type	Offline	Online
Recoverable database (archive logging enabled)	db2 backup db prod use snapshot	db2 backup db prod online use snapshot
Nonrecoverable database (archive logging disabled)	db2 backup db prod use snapshot	N/A

## 7.4 DB2 Restore Using Snapshot Technology

For larger databases, a conventional restore might take a significant amount of time to complete, therefore increasing the downtime of the database. DB2 9.5, however, allows database restores from a Snapshot copy backup that can complete in seconds, regardless of the size of the database.

In DB2 9.5, Snapshot copy backup restores are invoked through DB2 ACS and retrieved by the Snapshot copy backup image from their respective volumes. Similar to a traditional backup, these volumes then get restored. However, the process used to perform this restore is slightly different.

The DB2 restore process invoked with USE SNAPSHOT option performs the following internal operations:

1. Determine the volumes to restore.
2. Unmount the database file systems.
3. Perform volume-level restores on the NetApp storage system.
4. Remount the database file systems.
5. Place the database in roll-forward pending state (for recoverable database only).

The default behavior when restoring data from a Snapshot copy backup image will be a FULL DATABASE OFFLINE restore of all paths that make up the database. These include containers in the database path (DBPATH) and the storage paths of the most recent Snapshot copy backup. If a timestamp is provided, then that Snapshot copy backup image will be restored. EXCLUDE LOGS is the default for all Snapshot copy backup commands unless INCLUDE LOGS is explicitly stated. Therefore, if include logs option was not specified in the backup, then primary and mirror logs cannot be restored using that backup.

The Snapshot copy backup restore is a volume-level restore; hence, all data currently in the volumes will be essentially destroyed and replaced with a Snapshot copy. For this reason, it is strongly recommended that the volumes used by the database should not be shared by other applications or users.

Prior to issuing a restore, make sure that all log paths are mounted and accessible. If log path is not found, restore will use the default log directory. This is the SQLOGDIR directory inside the database path.

The steps required to recover the database depend on its recoverability type. The following subsections describe the steps required to recover both recoverable and nonrecoverable databases.

1. Snapshot copy-based restore for a nonrecoverable database.

To perform a single partition database restore from a Snapshot copy backup, execute the following command:

```
db2 restore db [DatabaseName] use snapshot
```

Where:

- `DatabaseName` identifies the name of the database to be created.

By default, db2 takes the latest backup image from the storage system when the Snapshot copy option is used. To restore the database from a specific Snapshot copy, specify the timestamp of the backup copy:

```
db2 restore db [DatabaseName] use snapshot taken at [BackupTimeStamp]
```

Where:

- `DatabaseName` identifies the name of the database to be created.
- `BackupTimeStamp` identifies the time when the database backup was created.

The timestamp for the intended Snapshot copy backup can be obtained by executing the following command:

```
db2acsutil query db [DatabaseName]
```

Where:

- `DatabaseName` identifies the name of the database to be created.

For example, to query the backup timestamp for a database named prod, execute the following db2acsutil command from the database server:

```
db2acsutil query db prod
Instance Database Part Image Time Host First Log
=====
db2inst1 PROD 0 20080526121157 db2srv1 33
db2inst1 PROD 1 20080526121157 db2srv2 32
db2inst1 PROD 0 20080526121104 db2srv1 34
db2inst1 PROD 1 20080526121104 db2srv2 35
```

For example, to restore a database named prod using a Snapshot copy-based backup created at 20080526121157, you need to execute the following restore command:

```
export DB2NODE=0
db2 terminate
db2 DB20000I The TERMINATE command completed successfully.
db2 restore db prod use snapshot taken at 20080526121157
SQL2539W Warning! Restoring to an existing database that is the same as the backup image
database. The database files will be deleted.
Do you want to continue ? (y/n) y
DB20000I The RESTORE DATABASE command completed successfully.
export DB2NODE=1
db2 terminate
DB20000I The TERMINATE command completed successfully.
db2 restore db prod use snapshot taken at 20080526121157
SQL2539W Warning! Restoring to an existing database that is the same as the backup image
database. The database files will be deleted.
Do you want to continue ? (y/n) y
DB20000I The RESTORE DATABASE command completed successfully.
```

For a nonrecoverable database, the database recovery is complete after the Snapshot copy is restored.

## 2. Snapshot copy-based restore for a recoverable nonpartitioned database.

To perform a restore for a recoverable database from a Snapshot copy backup, execute the following command:

```
db2 restore db [DatabaseName] use snapshot
```

Where:

- `DatabaseName` identifies the name of the database to be created.

For example, to restore a database named `prod` using the latest Snapshot copy-based backup, execute the following command:

```
db2 restore db prod use snapshot
SQL2539W Warning! Restoring to an existing database that is the same as the backup image
database. The database files will be deleted.
Do you want to continue ? (y/n) y
DB20000I The RESTORE DATABASE command completed successfully.
```

After the restore has completed, the database will be in roll-forward pending state. To fully recover the database and bring the database out of roll-forward pending state, a roll-forward command must be issued. Roll-forward recovery may be performed to end of logs, to a point in time, or to end of backup.

Prior to issuing a roll-forward, make sure that all log files that are required to roll forward through are available. To check which log files the backup image includes, execute a `db2acsutil query db` command:

```
db2acsutil query db [DatabaseName]
```

Where:

- `DatabaseName` identifies the name of the database to be created.

For example, to check the log file required for the restore of a database named `prod`, you need to execute the following command:

```
db2acsutil query db prod
Instance Database Part Image Time Host First Log
=====
svtdbm7 PROD 0 20080526180512 db2srv1.ibm.com 18
svtdbm7 PROD 0 20080526180410 db2srv1.ibm.com 10
```

To perform a roll-forward recovery to end of logs, execute the following command:

```
db2 rollforward db [DatabaseName] to end of logs and complete
```

Where:

- `DatabaseName` identifies the name of the database to be created.

For example, to perform roll-forward recovery for a database named `prod`, execute the following command:

```
db2 rollforward db prod to end of logs and complete
```

You can also perform point-in-time recovery by executing the following command:

```
db2 rollforward db [DatabaseName] to <timestamp> and complete
```

Where:

- `DatabaseName` identifies the name of the database to be created.

For example, to perform a point-in-time recovery for a database name `prod`, you need to execute the following command:

```
db2 rollforward db prod to 2008-05-26-22.14.12 and complete
```

As an alternative, roll-forward may be performed to the end of backup. In this case, the database will only be rolled forward to the minimum recovery time:

```
db2 rollforward db [DatabaseName] to end of backup and complete
```

Where:

- `DatabaseName` identifies the name of the database to be created.

For example, to perform an end-of-backup roll-forward, execute the following command:

```
db2 rollforward db prod to end of backup and complete
```

### 3. Snapshot copy-based restore for a recoverable partitioned database.

A DPF database must be restored by partitions individually. In this case, first restore the catalog partition, followed by all other partitions. To restore a partition, execute the following commands on each partition:

```
export DB2NODE=[DBPartitionNumber]
```

Where:

- `DBPartitionNumber` identifies the database partition node.

```
db2 restore db [DatabaseName] use snapshot
```

Where:

- `DatabaseName` identifies the name of the database to be created.

For example, to restore a two-partition database named `prod`, execute the following commands from the database server:

```
export DB2NODE=0
db2 terminate
DB20000I The TERMINATE command completed successfully.
db2 restore db prod use snapshot
SQL2539W Warning! Restoring to an existing database that is the same as the backup image
database. The database files will be deleted.
Do you want to continue ? (y/n) y
DB20000I The RESTORE DATABASE command completed successfully.
export DB2NODE=1
db2 terminate
DB20000I The TERMINATE command completed successfully.
db2 restore db prod use snapshot
SQL2539W Warning! Restoring to an existing database that is the same as the backup image
database. The database files will be deleted.
Do you want to continue ? (y/n) y
DB20000I The RESTORE DATABASE command completed successfully.
```

Following the restore of all partitions, the database must be rolled forward to the end of logs, to a point in time, or to the end of backup, only as in the preceding roll-forward case with a single partition database.

Prior to issuing a roll-forward, make sure that all log files that are required to roll forward through are available. To check which log files the backup image includes, execute a `db2acsutil query db` command:

```
db2acsutil query db [DatabaseName]
```

Where:

- `DatabaseName` identifies the name of the database to be created.

For example, to check the log file required for the restore of a database named `prod`, execute the following command:

```
db2acsutil query db prod
Instance Database Part Image Time Host First Log
=====
db2inst1 PROD 0 20080526162627 db2srv1.ibm.com 9
db2inst1 PROD 1 20080526162627 db2srv2.ibm.com 8
```

Finally, you need to complete recovery by using one of the roll-forward recovery options to end of logs, to a point in time, or to the end of backup. For example, to roll forward to the end of logs, execute the following command:

```
db2 rollforward db prod to end of logs and complete
```

**Note:** By default, similar to conventional restore, DB2 Snapshot copy-based restore does not include the log directories from the backup image. The database manager will return an error when the restore is issued with the LOGTARGET INCLUDE option because the preexisting log directory on disk conflicts with the log directory in the backup image. To restore the database in this case, execute the following restore command:

```
db2 restore db [DatabaseName] use snapshot logtarget include force
```

Where:

- `DatabaseName` identifies the name of the database to be created.

For example, to restore a database named prod and replace the log directory on disk with the log directory from the Snapshot copy backup image; execute the following command from the database server:

```
db2 restore db prod use snapshot logtarget include force
SQL2539W Warning! Restoring to an existing database that is the same as the
backup image database. The database files will be deleted.
Do you want to continue ? (y/n) y
DB20000I The RESTORE DATABASE command completed successfully.
```

The restore process for recoverable and nonrecoverable databases can be summarized in a few easy steps, as illustrated in Table 2.

**Table 2) Restore options for recoverable and nonrecoverable database (PROD).**

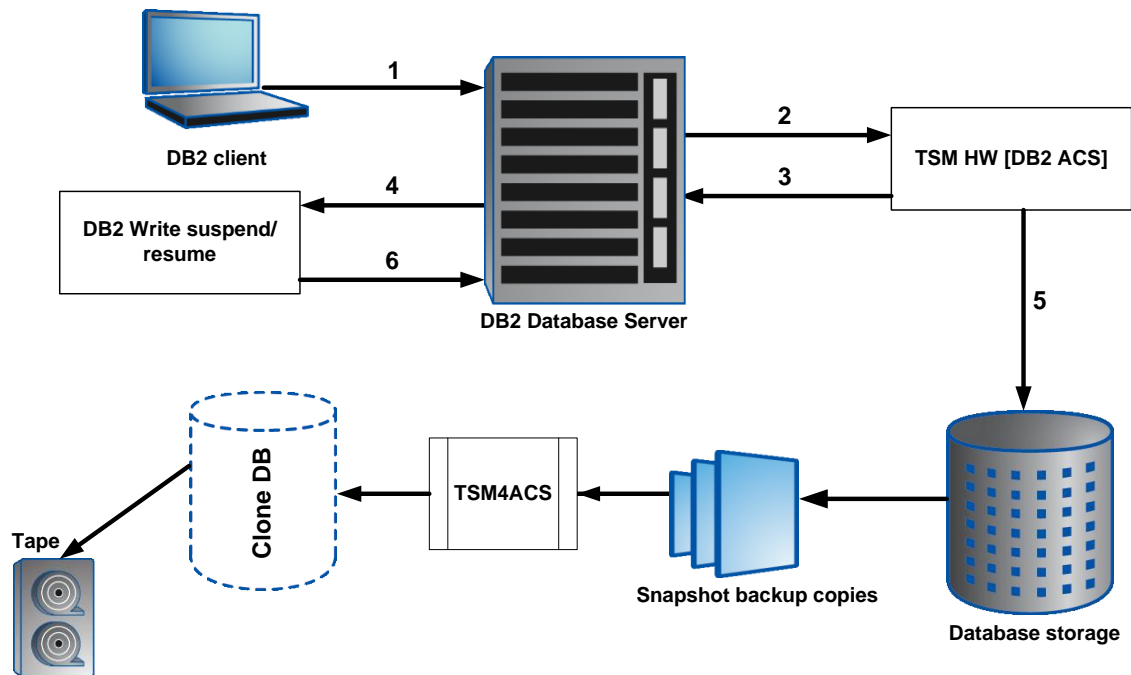
Database Type	Offline	Online
Recoverable database (archive logging enabled)	db2 restore db prod use snapshot without rolling forward	db2 restore db prod online use snapshot db2 rollforward db prod to end of logs and complete
Nonrecoverable database (archive logging disabled)	db2 backup db prod use snapshot without rolling forward	N/A

## 8 Database Clone and Tape Backup Using Tivoli Storage Manager

### 8.1 Database Clone

The database clone using a Snapshot copy-based backup image is the fastest and most storage-efficient way to provide a copy of a DB2 database. Each cloned database is a transparent, virtual copy of your production database that can be used for essential enterprise operations such as tape backup, testing, bug fixing, and multiple simulations against large datasets and platform and upgrade checks. Similar to DB2 ACS, which allows Snapshot backups and restore, TSM has an offload agent named TSM4ACS that allows creating database clones from Snapshot backup images. Figure 3 illustrates DB2 database clone and backup to tape using Snapshot backup images.

Figure 3) DB2 database clone and backup to tape using Snapshot backup images.



The offload agent provides a single user interface for all added functionality associated with a TSM for ACS package.

To create the database clone using TSM4ACS agent, you need to complete the following prerequisites on both source and target servers:

1. Install the TSM4ACS agent.
2. Update the DB2 ACS profile.
3. Add the TSMACS license.
4. Restart the DB2 ACS demon.

The TSM4ACS agent does not come with the DB2 9.5 install files and needs to be installed separately. To update the DB2 ACS profile, log in as the DB2 instance user and run the setup.sh wizard, located in the `/home/db2inst1/sqllib/acs/` directory. Appendix D describes the step-by-step process to update the profile using setup.sh.

After successfully following the preceding steps to install and configure the TSM4ACS agent, the database clone can be created by completing the following easy steps:

1. Identify the backup image timestamp.
2. Mount the Snapshot backup image.
3. Initialize the database.
4. Identify the database image timestamp.

To create the clone of a DB2 database using a Snapshot copy-based backup image, you need to identify the timestamp of the backup image. The timestamp can be obtained by executing the following DB2 ACS utility command:

```
db2acsutil query db [DatabaseName]
```

Where:



- `DatabaseName` identifies the name of the source database.

For example, to find the Snapshot copy backup timestamp of the source database named `prod`, you need to execute the following command on the database server:

```
db2acsutil query db prod
Instance Database Part Image Time Host First Log
=====
db2inst1 PROD 0 20080526162627 db2srv1.ibm.com 9
db2inst1 PROD 1 20080526162627 db2srv2.ibm.com 8
```

The timestamp is an optional parameter for the TSM mount command. If a timestamp is not provided during the mount operation, the TSM for ACS offload agent will pick the most recent Snapshot backup image for clone.

## 8.2 Mount Snapshot Backup Image on Target Server

If executed with the mount option, the TSM4ACS agent allows you to mount a Snapshot backup image on a target system.

The agent internally performs the following steps:

1. Create FlexClone<sup>®</sup> volumes on underlying NetApp storage system based on the provided timestamp for the Snapshot backup image.
2. Export the new FlexClone volumes on the underlying NetApp storage system.
3. If database is SAN environment, it creates igroup mapping for LUNs.
4. Mount the new FlexClone volumes/LUNs on the target system.

To perform the mount function of TSM4ACS agent, execute the following command on the target system under the location `/home/db2inst1/sqlib/acs`:

```
./tsm4acs -f mount -c [source Server] -p [Source ACS profile] -i [source instance] -d [source database] -T [timestamp]
```

Where:

- `[-c]` `Source Server` identifies the name of the source database server.
- `[-p]` `Source ACS profile` identifies the absolute path of the source ACS profile.
- `[-i]` `Source instance` identifies the DB2 instance name of the source database.
- `[-d]` `Source database` identifies the source database name.
- `[-T]` `Timestamp` identifies the timestamp of the Snapshot backup image on the source server.

For example, to create a database clone from a Snapshot backup image that was created at 20080526162627 for the database named `prod` that resides on server named `db2serv1.ibm.co` under the instance `db2inst1`, you need to execute the following command:

```
./tsm4acs -f mount -c db2srv1.ibm.com -p /home/db2inst1/sqlib/acs/profile -i db2inst1 -d prod -T 20080526162627
```

The output snippet from the preceding command can be found in Appendix E.

1. Initializing the database clone.

After successfully mounting the Snapshot backup image on the target server, the newly created cloned database must be initialized before it is opened for use. To initialize the database, run the following commands on the target database server:

```
db2 catalog db [DatabaseName] on [Database path]
db2inidb [DatabaseName] as acs standby
```

Where:

- `DatabaseName` identifies the name of the target database.
- `Database path` identifies the path of the target database.

For example, to initialize the database named `prodcl`, execute the following command on the target database server:

```
db2 catalog db prodcl on /mnt/dbdata
DB20000I The CATALOG DATABASE command completed successfully.
DB21056W Directory changes may not be effective until the directory cache is refreshed.
db2inidb prodcl as acs standby
DBT1000I The tool completed successfully.
```

## Roll Forward with ACS Clone

Before you roll forward the database created through ACS clone, you need to either clean up the existing transaction logs, which come with the clone, or move the logs to an alternate location. To roll forward the database, you will copy the archival logs from the source database to the active logs directory of the target or clone database. If you do not perform this step, the database will stop the roll-forward operation at the time of the Snapshot copy creation.

**Note:** The TSM4ACS function unmount will release all resources on the target server that were used by the TSM4ACS mount function by an appropriate device agent started by TSM4ACS. The following steps will be performed internally during an unmount operation:

1. Unmount the associated file systems with the cloned database copy.
2. Destroy the FlexClone volumes on the underlying NetApp storage system.

## 8.3 Tape Backup from Clone

We assume that you are using TSM to back up the database to tape. After the database clone is successfully created, you can easily back up your database to tape by executing the following command:

```
db2 backup db [DatabaseName] use tsm
```

Where:

- `DatabaseName` identifies the name of the target database.

To back up the newly created clone database `prodcl`, execute the following command on the target database server:

```
db2 backup db prodcl use tsm
```

## 9 Limitations

Following are the two limitations of DB2 9.5 ACS:

- Needs separate ACS license.
- Supports only one storage head.

**DB2 ACS license.** DB2 9.5 comes with an unlicensed version of DB2 ACS that allows the retention of only two Snapshot backup copies. In a real production environment, you need to keep more than two backup copies. To keep more than two copies, you need to obtain a valid license key and apply the license key for ACS to remove this restriction.

**Single storage head support.** The DB2 9.5 ACS supports one storage head per database partition; that is, a database partition cannot span across multiple storage heads.

## 10 Conclusion

The DB2 ACS together with NetApp storage system offers a compelling advantage for database administrators in terms of database backup, recovery, and creating writable database clone copies using the integrated TSM commands. By creating Snapshot copies frequently and keeping an appropriate number of Snapshot copies online, database administrators have the ability to restore databases without incurring the overhead normally associated with recovery from tape. Additionally, backup and recovery performance is dramatically improved—the process of creating or restoring from a Snapshot copy can be done in a fraction of the time needed when conventional methods are used. Finally, with the ability to clone the database from the Snapshot copy backups, database administrators can create virtual copies of the production databases and back them up to TSM without affecting the application's performance.

## Appendixes

### Appendix A: ACS Backup Flow

#### ACS Log Files

Located under the <db2 install path>/sqlib/acs/logs, the ACS log files contain a record of all Snapshot copy backup, restore, and query operations. Following is an example of a backup operation:

#### ACSD Log Output for Backup Operation

```
=====
New backup operation started for database instance db2inst1, database PROD.

=====
DB 0007 05/26/08 12:11:59 BKI1510I: New connection received.
DB 0008 05/26/08 12:11:59 BKI1510I: New connection received.
DB 0007 05/26/08 12:11:59 BKI1513I: *****> Database client connected: Instance db2inst1,
database PROD, partition NODE0001
DB 0007 05/26/08 12:11:59 BKI1515I: Client is logging to
db2srv1.ibm.com:/home/db2inst1/sqlib/acs/logs/client.db2inst1.PROD.1.20080526121158.log.
DB 0008 05/26/08 12:11:59 BKI1513I: *****> Database client connected: Instance db2inst1,
database PROD, partition NODE0000
DB 0008 05/26/08 12:11:59 BKI1515I: Client is logging to
db2srv1.ibm.com:/home/db2inst1/sqlib/acs/logs/client.db2inst1.PROD.0.20080526121158.log.
DB 0007
DB 0008
DEV 0004 05/26/08 12:12:00 BKI1514I: *****> Device client connected.
DEV 0004 05/26/08 12:12:00 BKI1557I: Device client is logging to
db2srv1.ibm.com:acsnnas.db2srv1.ibm.com.STANDARD.1.20080526121158.log.
DEV 0005 05/26/08 12:12:00 BKI1510I: New connection received.
DEV 0005 05/26/08 12:12:00 BKI1514I: *****> Device client connected.
DEV 0005 05/26/08 12:12:00 BKI1557I: Device client is logging to
db2srv1.ibm.com:acsnnas.db2srv1.ibm.com.STANDARD.0.20080526121158.log.
DB 0007 05/26/08 12:12:08 BKI1521I: Retaining 2 backups
DB 0007 05/26/08 12:12:08 BKI1511I:
=====

New delete operation started for database instance db2inst1, database PROD.

=====
DB 0007 DB 0009 05/26/08 12:12:08 BKI1510I: New connection received.
DB 0007 DB 0009 05/26/08 12:12:08 BKI1513I: *****> Database client connected: Instance
db2inst1, database PROD, partition NODE0001
DB 0007 DB 0009 05/26/08 12:12:08 BKI1515I: Client is logging to
db2srv1.ibm.com:/home/db2inst1/sqlib/acs/logs/client.db2inst1.PROD.1.20080526121158.log.
DB 0007 DB 0009 05/26/08 12:45:48 BKI1511I:
=====
```

## ACS Client Log Output for Backup Operation

```
BKI8806I: IBM Tivoli Storage Manager for Advanced Copy Services 5.5.0.0 (323)
Jan 11 2008
BKI8816I: DB2 version 'SQL09051' detected.

BKI8800I: The command is: backup
BKI8801I: Time: 2008-05-26-15.00.54.0966 --- PID: 499850
BKI8300I: db2ACSBEGINoperation() returned with code 0.
BKI1553I: acsd is logging to db2srv1:/home/db2inst1/sqllib/acs/logs/acsd.20080525.log.
BKI8300I: db2ACSPartition() returned with code 0.
BKI8300I: db2ACSPrepare() returned with code 0.
BKI8300I: db2ACSSnapshot() returned with code 0.
BKI8300I: db2ACSVerify() returned with code 0.
BKI8300I: db2ACSStoreMetaData() returned with code 0.
BKI1553I: acsd is logging to db2srv1:/home/db2inst1/sqllib/acs/logs/acsd.20080525.log.
BKI8300I: db2ACSEndOperation() returned with code 0.
BKI8300I: db2ACSTerminate() returned with code 0.
BKI8810I: Cleaning up resources of process 499850.
```

## Appendix B: Configure More Than One Instance on a Machine

To use ACS with multiple instances of db2 on a machine, additional configuration is required. Follow the same process as was performed on the first instance, except:

1. Add another entry into /etc/services with a new port number for ACS for the new instance.

For example, if instance db2inst1 (previously configured) had:

```
db2acs      57328/tcp
```

2. Add a new port such as the following:

```
db2acs2    57329/tcp
```

3. Specify this different port number during setup in setup.sh.

For example, if instance db2inst1 (previously configured) has:

```
ACSD      [localhost 57328 ] db2srv1 57328
```

4. Configure instance db2inst2 to have:

```
ACSD      [localhost 57328 ] db2srv1 57329
```

After successful execution of setup.sh, new entries will be added to /etc/inittab to start new ACS processes.

## Appendix C: Delete Backups with db2acsutil

DB2 9.5 provides a utility named db2acsutil to manage Snapshot backup copies. This utility can be used to list, view the status of, and delete the backup copies.

To view Snapshot backup copies for all databases to find images to delete, execute the following command with the query option:

```
db2acsutil query status
```

For example:

```
db2acsutil query status
Instance  Database  Part Image Time      Status
=====  =====  ===  =====  =====
db2inst1  PROD      0  20080521091638  Repetitively restorable + Destructively restorable
db2inst1  PROD      1  20080521091638  Repetitively restorable + Destructively restorable
```

To delete the queried Snapshot copy backup, execute the `db2acsutil` command with the delete option:

```
db2acsutil delete snapshot taken at <timestamp> db [DatabaseName]
```

For example, to delete a Snapshot backup copy created at 20080521091638 for the database named `prod`, execute the following command:

```
db2acsutil delete snapshot taken at 20080521091638 db prod
Instance Database Part Image Time Host First Log
=====
db2inst1 PROD 0 20080521091638 db2srv1 712
Are you sure (y/[n])? y
Deleted.
db2inst1 PROD 1 20080521091638 db2srv2 703
Are you sure (y/[n])? y
Deleted
```

## Appendix D: DB2 ACS Setup Process

### setup.sh Wizard Output for tsm4acs

```
$ ./setup.sh

Do you want to install TSM for ACS on the production system? [Y/N]
Answer "no" if you want to install a backup server.
Y
Please enter the directory where the profile will be stored: [/home/db2inst1/acs]
/home/db2inst1/sqllib/acs
selected profile path: /home/db2inst1/sqllib/acs

Do you have a full TSM license to enable all features of TSM for ACS ?[y/n]
y

***** Profile parameters for section GLOBAL: *****
ACS_DIR [/home/db2inst1/sqllib/acs ]
ACSD [localhost 57328 ]
TRACE [NO ]
***** Profile parameters for section CLIENT: *****
APPLICATION_TYPE *mandatory parameter* DB2
TSM_BACKUP [NO ] YES
MAX_VERSIONS [ADAPTIVE ]
LVM_FREEZE_THAW [YES ]
DEVICE_CLASS [STANDARD ]

***** Profile parameters for section OFFLOAD: *****
OPTIONS [ ]
PARALLELISM [AUTO ]
NUM_SESSIONS [1 ]
NUM_BUFFERS [AUTO ]
BUFFER_SIZE [AUTO ]
***** Profile parameters for section ACSD: *****
ACS_REPOSITORY *mandatory parameter* /home/db2inst1/acsrep
ADMIN_ASSISTANT [NO ]
***** Profile parameters for section STANDARD: *****
COPYSERVICES_HARDWARE_TYPE *mandatory parameter* NAS_NSERIES
COPYSERVICES_PRIMARY_SERVERNAME *mandatory parameter* db2srv1
COPYSERVICES_USERNAME [superuser ] db2inst1
=====
The profile has been successfully created.
Do you want to continue by specifying passwords for the defined devices? [y/n]
y

Please specify the passwords for the following profile sections:
OFFLOAD
STANDARD
master

Creating password file at /home/db2inst1/sqllib/acs/shared/pwd.acsd.
```

## Appendix E: TSM4ACS Mount Process

### Mount Function Output of tsm4acs

```
TSM for ACS offload agent
Version 5, Release 5, Level 0.0 for AIX LF -
Build: 329 generated on Feb 22 2008
© Copyright IBM Corporation, 2000, 2008, All Rights Reserved.

BKI8800I:      The command is: mount
BKI6501I:      Initializing 'mount' request.
BKI6502I:      Executing 'mount' request.
BKI1553I:      acsd is logging to db2serv1:/home/db2inst1/sqlllib/acs/logs/acsd.20080530.log
BKI6503I:      Terminating 'mount' request.
BKI6512I:      The 'mount' request for database 'PRODCL' with partitions (0) processed
successfully.
BKI0024I:      Return code is:0.
```

## References

The following references were used in this technical report:

- Parity RAID for Enhanced Data Protection with RAID-DP:  
<http://www.netapp.com/library/tr/3298.pdf>
- File System Design for an NFS File Server Appliance:  
[http://media.netapp.com/documents/wp\\_3002.pdf](http://media.netapp.com/documents/wp_3002.pdf)
- Data Protection Online Backup and Recovery Guide:  
<http://now.netapp.com/NOW/knowledge/docs/ontap/rel701r1/html/ontap/onlinebk/index.htm>
- DB2 9 for UNIX: Integrating with a NetApp Storage System:  
<http://www.netapp.com/library/tr/3531.pdf>
- Data Recovery and High Availability Guide and Reference:  
[ftp://ftp.software.ibm.com/ps/products/db2/info/vr82/pdf/en\\_US/db2hae81.pdf](ftp://ftp.software.ibm.com/ps/products/db2/info/vr82/pdf/en_US/db2hae81.pdf)
- Quick Beginnings for DB2 Servers:  
[ftp://ftp.software.ibm.com/ps/products/db2/info/vr82/pdf/en\\_US/db2ise81.pdf](ftp://ftp.software.ibm.com/ps/products/db2/info/vr82/pdf/en_US/db2ise81.pdf)
- Data ONTAP 7.1 System Administration Guide:  
<http://now.netapp.com/NOW/knowledge/docs/ontap/rel71/pdfs/ontap/sysadmin.pdf>

## Version History

Version	Date	Document Version History
Version 1.0	July 2008	Original draft- Jimmy Stamkos, Aslam Nomani, Dale McInnis - IBM Bobby Oommen, Jawahar Lal - NetApp
Version 1.0.1	September 2012	Updated new rollforward step for ACS cloning - Bobby Oommen, Ashwath Narayan

Refer to the [Interoperability Matrix Tool](#) (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

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.

[Go further, faster®](#)



[www.netapp.com](http://www.netapp.com)

© 2012 NetApp, Inc. All rights reserved. No portions of this document may be reproduced without prior written consent of NetApp, Inc. Specifications are subject to change without notice. NetApp, the NetApp logo, Go further, faster, Data ONTAP, FlexClone, FlexVol, RAID-DP, SnapRestore, Snapshot, and WAFL are trademarks or registered trademarks of NetApp, Inc. in the United States and/or other countries. Linux is a registered trademark of Linus Torvalds. UNIX is a registered trademark of The Open Group. All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such. TR-3668-0912