

Use Tivoli Storage Manager to back up and recover a DB2 database

Bharat Vyas
Deepa Vyas

October 17, 2012

This article describes the basics of IBM® Tivoli® Storage Manager and IBM DB2® architecture, and shows you how to use the Tivoli Storage Manager backup and restore features. This article also provides step-by-step instructions that show you how to back up and restore data on a Tivoli Storage Manager server for the DB2 database. This document can be used as a guide for DB2 database administrators and Tivoli Storage Manager administrators.

Introduction

IBM Tivoli Storage Manager is a software product that addresses the challenges of complex storage management across distributed environments. This product protects and manages a broad range of data, from the workstation to the corporate server environment. IBM DB2 is a family of relational database management system (RDBMS) products. The DB2 database includes a data dictionary or a set of system tables that describe the logical and physical structure of the data. DB2 provides self-tuning capabilities, and dynamic adjustment and tuning.

This article first reviews concepts and considerations. It explains how to install and configure the Tivoli Storage Manager backup-archive client, and provides step-by-step instructions and techniques that show you how to back up and restore data on a Tivoli Storage Manager server for the DB2 database.

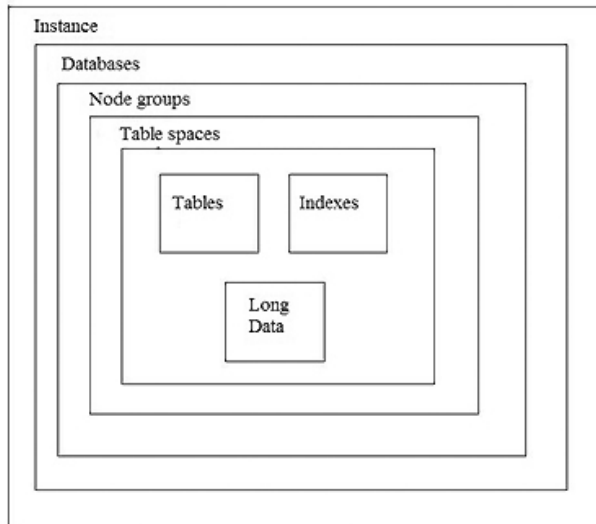
DB2

IBM DB2 is an RDBMS in which the data can be referenced in terms of its content without regard to the way the data is stored. A DB2 database has a physical and logical storage model to handle data. A DB2 database contains various objects such as tables, views, indexes, schemas, locks, triggers, stored procedures, packages, buffer pools, log files, and table spaces. Some of these objects, such as tables and views, help to identify how the data is organized. Other objects, such as table spaces, refer to the physical implementation of the database. Some other memory-related objects, such as buffer pools, deal with database performance.

An instance (database manager) is a logical environment for managing data and system resources assigned to it. A system can have more than one instance. Each instance has its own set of

database manager configuration parameters and security. An instance can have one or more databases. The hierarchy is shown in [Figure 1](#).

Figure 1. DB2 object hierarchy in an instance



The user's data is in the tables. While tables contain rows and columns, the users do not know about the actual physical representation and storage of the data. This fact is sometimes referred to as the physical independence of the data. The tables are placed into table spaces. A *table space* is used as a layer between the database and the container objects that hold the actual table data. A table space can contain more than one table. A *container* is a physical storage device. It can be identified by a directory name, a device name, or a file name. A container is assigned to a table space. A table space can span various containers, which lets you get around operating system limitations that might restrict the amount of data that one container can accommodate.

Tivoli Storage Manager

IBM Tivoli Storage Manager is a software product that provides storage management services for data, primarily backup, restore, archive, and retrieve, by using a client/server model. In general terms, Tivoli Storage Manager backup-archive clients are installed on each system (such as file servers, database servers, client workstations). Using a configured network transport such as TCP/IP, each Tivoli Storage Manager client sends copies of its files as either backup or archive objects to a Tivoli Storage Manager server. The server stores the client files in a centralized storage system (typically consisting of large amounts of disk or tape storage).

A DB2 database is a self-contained data file that you can back up and restore using Tivoli Storage Manager. Tivoli Storage Manager restores a DB2 database in its entirety because it is just a file for Tivoli Storage Manager. If a DB2 database is deleted or corrupted, Tivoli Storage Manager can restore the most recent or any previous backup version of this database from the Tivoli Storage Manager server to the DB2 server or client.

Tivoli Storage Manager API

The Tivoli Storage Manager application programming interface (API) provides a library of functions that allow independent software applications and custom-built applications to back up and archive their data to a Tivoli Storage Manager server. The DB2 DBMS also uses the Tivoli Storage Manager API for backup and restore operations.

DB2 provides its own backup utility that can be used to back up data at the table space level or the database level. If you set up this utility to use Tivoli Storage Manager as the backup media, DB2 communicates with the Tivoli Storage Manager API for backup and restore operations. Thus, both the Tivoli Storage Manager API client and backup-archive client work together to provide full data protection for your DB2 environment. The API client and the backup-archive client can run simultaneously on the same DB2 server. The Tivoli Storage Manager server considers them separate clients.

Basic DB2 and Tivoli Storage Manager architecture

The Tivoli Storage Manager backup-archive client can back up, restore, archive, and retrieve client file system data. The client can back up any files and uses standard operating system functions to access files within file systems. This method affects how the DB2 and other database systems are backed up. Each database appears as an individual file on the server or client file system.

To manage database or table space-level backup processing, the DB2 database manager can use the Tivoli Storage Manager API. A Tivoli Storage Manager client API must be installed on each DB2 database server.

Figure 2. DB2 Tivoli Storage Manager architecture

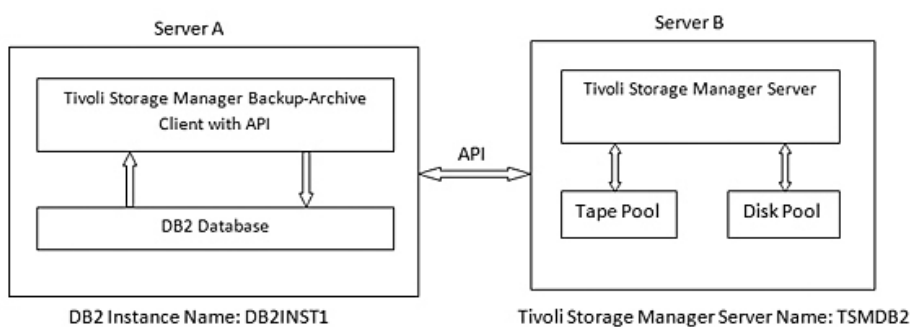


Figure 2 shows one method that you can use to back up a DB2 database. The Tivoli Storage Manager server is on Server B. Server A contains the DB2 database and the Tivoli Storage Manager backup-archive client and API.

In Figure 2, the following Tivoli Storage Manager components are used:

- *Tivoli Storage Manager backup-archive client*: Provides a tool for users to back up versions of files to a Tivoli Storage Manager server, which can be restored if the original files are lost or damaged. Users can also archive files for long-term storage and retrieve the archived files when necessary.

- *Tivoli Storage Manager API*: Allows independent software applications and custom-built applications to back up or archive its application data to a Tivoli Storage Manager server. DB2 calls functions provided by the Tivoli Storage Manager API to send database and table space backups directly to the Tivoli Storage Manager server.
- *Tivoli Storage Manager server*: Provides services to store and manage the client's data. The server can store data on disk and tape storage devices. Tivoli Storage Manager server Version 5.5 has a proprietary database. Tivoli Storage Manager Version 6 and later uses a DB2 database to track information about server storage, clients, client data, policies, and schedules.

Installing and configuring the Tivoli Storage Manager backup-archive client

Before you can back up a DB2 database using Tivoli Storage Manager, you must install and configure the Tivoli Storage Manager backup-archive client and the Tivoli Storage Manager server. For this article, the client and server are both set up on AIX® systems, therefore, the following components are required:

- Tivoli Storage Manager backup-archive client Version 6.1, or later, for AIX
- Tivoli Storage Manager server Version 6.1, or later, for AIX

Installing the Tivoli Storage Manager backup-archive client

As shown in [Figure 2](#), the Tivoli Storage Manager backup-archive client and API must be installed on a DB2 server and the Tivoli Storage Manager server must be installed on a separate system. In this example, the DB2 server is located on an AIX system, so the AIX System Management Interface Tool (SMIT) is used to install the backup-archive client.

Choose the following file sets when you install the backup-archive client:

- tivoli.tsm.client.ba64
- tivoli.tsm.client.api.32bit
- tivoli.tsm.client.api.64bit
- GSKit8.gskcrypt64.ppc.rte and GSKit8.gskssl64.ppc.rte (required by the 64-bit client API)

After you install the Tivoli Storage Manager backup-archive client, run the AIX `ls1pp` command to verify the following installed file sets:

```
$ ls1pp -L | grep "tivoli"

tivoli.tsm.client.api.32bit
tivoli.tsm.client.api.64bit
tivoli.tsm.client.ba.64bit.base
tivoli.tsm.client.ba.64bit.common
tivoli.tsm.client.ba.64bit.hdw
tivoli.tsm.client.ba.64bit.image
tivoli.tsm.client.ba.64bit.nas
tivoli.tsm.client.ba.64bit.snphdw
tivoli.tsm.client.ba.64bit.web
```

The Tivoli Storage Manager backup-archive client and API default installation directories are:

- Backup-Archive client directory: `/usr/tivoli/tsm/client/ba/bin64`
- API directory: `/usr/tivoli/tsm/client/api/bin64`

Configuring the Tivoli Storage Manager backup-archive client

To configure the Tivoli Storage Manager backup-archive client, complete the following tasks.

1. Set the environment variables

Set the following DSMI environment variables in either the operating system shell or the `/home/instance_home_dir/sqlplib/userprofile` file.

DSMI_DIR

DSMI_CONFIG

DSMI_LOG

Important: The DB2 DBMS reads these environment variables during the DB2 instance startup. If you change the variables, you must restart the DB2 instance for the changes to take effect.

DSMI_DIR

This variable points to the API installation directory. The `dsmtca` file, the `dsm.sys` file, and the language files must be in the directory pointed to by the `DSMI_DIR` environment variable. Setting the `DSMI_DIR` variable is optional. If it is not specified, the default directory is `/usr/tivoli/tsm/client/api/bin64` (for AIX 64-bit).

DSMI_CONFIG

This variable points to the fully qualified path and file name of the Tivoli Storage Manager `dsm.opt` client options file. This file contains the name of the server to be used.

DSMI_LOG

This variable points to the directory path where the error log file, `dsierror.log`, is to be created. Add the following lines in the `userprofile` file.

```
export DSMI_CONFIG=/usr/tivoli/tsm/client/api/bin64/dsm.opt
export DSMI_LOG=/home/db2inst1
export DSMI_DIR=/usr/tivoli/tsm/client/api/bin64
```

Log off and log in again as an instance user and run the `.profile` file.

```
$ ~/.profile
```

2. Create the `dsm.sys` client options file

Log in using the root user ID and create the `dsm.sys` file with the following entries (customize for your installation). This file must be in the directory that is specified by the `DSMI_DIR` environment variable.

```
servername tsmdb2           //name of this stanza
commmethod tcpip
tcpserveraddress x.xx.xxx.xxx //IP address of Tivoli Storage Manager server
tcpport 1500                //Port where Server is listening
nodename tsmdb2             //Must match nodename on Tivoli Storage Manager server
passwordaccess generate
```

3. Create the `dsm.opt` client options file

Log in using the root user ID and create the `dsm.opt` file. This options file has only one line in it, which is a reference to the server stanza in the `dsm.sys` file. This file must be in the directory specified by the `DSMI_CONFIG` environment variable.

```
Servername    tsmdb2
```

4. Recycle the DB2 instance

Stop and start the DB2 instance.

```
$ db2stop
SQL1064N DB2STOP processing was successful.
$ db2start
SQL1063N DB2START processing was successful.
```

5. Set the API password

To access the Tivoli Storage Manager server, client users (called *nodes*) must have a password to access the server. The DB2 `dsmapiw` program uses the Tivoli Storage Manager API to create the encrypted password file. The DB2 application includes the `dsmapiw` utility, which is installed in the `/home/instance_home_dir/sqllib/adsm` directory.

Log in as root user to run the `dsmapiw` utility. Before you run `dsmapiw`, you must set the DSMI environment variables similar to that on the DB2 instance, as shown in the following example:

```
export DSMI_CONFIG=/usr/tivoli/tsm/client/api/bin64/dsm.opt
export DSMI_DIR=/usr/tivoli/tsm/client/api/bin64
```

To set the password, run the `dsmapiw` utility from the `/home/instance_home_dir/sqllib/adsm/` directory. When prompted by the `dsmapiw` utility, specify the password for the Tivoli Storage Manager node that is stored on the Tivoli Storage Manager server.

```
$ ./dsmapiw
*****
* Tivoli Storage Manager                               *
* API Version = 6.1.0                                  *
*****
Enter your current password:
Enter your new password:
Enter your new password again:

Your new password has been accepted and updated.
```

Tip: If you use the `passwordaccess prompt` option, you do not have to run the `dsmapiw` utility.

Tivoli Storage Manager server considerations

The Tivoli Storage Manager server acts as a central storage repository for backup and archive data from one or more Tivoli Storage Manager clients. The server maintains a database to track client data, users (nodes, administrator IDs), data retention policies, and Tivoli Storage Manager server resources. The data retention policy manages the following settings:

- How long to keep an archive
- How long to keep a backup
- How many copy versions of a backup to maintain

The server also controls the Tivoli Storage Manager server storage (storage pools). These storage pools store the client's backup and archive data. Each storage pool represents a single type of storage media. For example, one storage pool can represent a pool of random access disks,

another can represent a pool of sequential access tapes, and a third can represent a pool of sequential access optical platters.

Complete the following steps on the Tivoli Storage Manager server:

1. [Define the policy domain.](#)
2. [Define the policy set.](#)
3. [Define the management class.](#)
4. [Assign the default management class.](#)
5. [Define the copy group.](#)
6. [Validate and activate the policy set.](#)
7. [Register the client node.](#)

1. Use the following command to define the policy domain. When a client node is registered, it is assigned to an existing domain and this node or domain association defines how that node's data is managed by the Tivoli Storage Manager server.

```
tsm: TSM6120>define domain dbdomain
ANR1500I Policy domain DBDOMAIN defined.
```

2. Use the following command to define the policy set.

```
tsm: TSM6120>define policysset dbdomain dbpolicy
ANR1510I Policy set DBPOLICY defined in policy domain DBDOMAIN.
```

3. Use the following command to define the management class.

```
tsm: TSM6120>define mgmtclass dbdomain dbpolicy dbmgmtclass
ANR1520I Management class DBMGMTCLASS defined in policy domain DBDOMAIN, set
DBPOLICY.
```

4. Use the following command to assign a default management class.

```
tsm: TSM6120>assign defmgmtclass dbdomain dbpolicy dbmgmtclass
ANR1538I Default management class set to DBMGMTCLASS for policy domain
DBDOMAIN, set DBPOLICY.
```

5. Use the following command to define the copy group.

```
tsm: TSM6120>define copygroup dbdomain dbpolicy dbmgmtclass type=backup dest=ltopool
VEREXISTS=1 VERDEL=0 RETEXTRA=0 RETONLY=0
ANR1530I Backup copy group STANDARD defined in policy domain DBDOMAIN, set
DBPOLICY, management class DBMGMTCLASS.
```

All objects that the DB2 backup stores on the Tivoli Storage Manager server are given a unique name based on a timestamp for the object. Because these objects are unique, they are never automatically expired from the Tivoli Storage Manager server. The `db2adut1` tool can be used to delete (mark inactive) the unwanted backup objects from the Tivoli Storage Manager server.

To remove the DB2 backups from the Tivoli Storage Manager Server as soon as possible after they are marked inactive, the backup copy group must have the following retention settings: `VEREXISTS=1`, `VERDEL=0`, `RETEXTRA=0`, `RETONLY=0`.

6. Use the following commands to validate and activate the policy set.

```
tsm: TSM6120>validate policysset dbdomain dbpolicy
tsm: TSM6120>activate policysset dbdomain dbpolicy
```

7. Use the following command to register the client node. To access the Tivoli Storage Manager server, nodes must log in to the Tivoli Storage Manager server using their respective node

name and password. Issue the following command to register the DB2 node with a password to the server and also assign the node to the policy domain.

```
tsm: TSM6120>register node db2node password domain=dbdomain
```

Backup techniques

After the Tivoli Storage Manager server and client are configured, DB2 can back up data onto the Tivoli Storage Manager server. Use the `USE TSM` option to specify that Tivoli Storage Manager information be used during the database backup operation. When the `USE TSM` option is specified on the DB2 backup operation, the API is used to direct the database backup to the Tivoli Storage Manager server. The `USE TSM` option instructs DB2 to use the DB2 API library interface that calls Tivoli Storage Manager. For example:

```
DB2 BACKUP DATABASE database_name USE TSM
```

There are several ways that you can back up the DB2 database using Tivoli Storage Manager:

- [Offline backup](#)
- [Online backup](#)
- [Table space backup](#)

Offline backup

An offline backup involves shutting down the database before you start the backup operation and restarting the database after the backup is complete. Offline backups are relatively simple to administer. However, users and batch processes cannot access the database while the backup is taking place. You must schedule sufficient time to perform the backup operation.

The following example shows how to run an offline DB2 database backup. The example uses the DB2 command-line interface (CLI) to back up the SAMPLE database.

1. Log in as an instance owner, or with higher authority.
2. Ensure that there are no applications connected to the database that you want to back up. Use the `DB2 LIST APPLICATIONS` command to check if any applications are connected to the database.

```
$ db2 list applications for db sample
```

Auth Id	Application Name	Appl. Handle	Application Id	DB Name	# of Agents
db2inst1	db2bp	245	*LOCAL.DB2INST1.120731143027	SAMPLE	3
db2inst1	db2med	132	*LOCAL.DB2INST1.120731162426	SAMPLE	2

3. Log off all applications that are connected to the database. The previous step shows that two applications are connected to the database. Use the `FORCE APPLICATION` command to disconnect the applications. To specify more than one application on the `FORCE APPLICATION` command, use a comma to separate the handles.

```
$ db2 "force application ( 245,132 )"
```

```
DB20000I The FORCE APPLICATION command completed successfully.
```

```
DB21024I This command is asynchronous and might not be effective immediately.
```


Tip: If many applications are connected to the database, you can use the `db2 force application all` command to log them all off. However, be careful when using this command, because it logs off all the applications connected to any database in the instance.

4. Reissue the `LIST APPLICATIONS` command (see [step 2](#)) and verify that there are no more users connected to the database.
5. Use the `BACKUP DATABASE` command with the `USE TSM` option to back up the database.

```
$ db2 backup db sample use tsm
```

```
Backup successful. The timestamp for this backup image is : 20120806185756
```

Online backup

In a DB2 database system, you can back up data while the database is started and is in use. Clearly, if a database is being backed up while users are updating it, it is likely that the data backed up might be inconsistent. The DB2 DBMS uses log files during the recovery process to recover the database to a fully consistent state. This process requires that log files be gathered at the start of the database backup and when the backup is completed.

At the end of an online backup, the current active log is archived and is included in the backup image. By using the online method, you can collect the set of log files that are required to recover the database.

Restriction: During an online backup, various applications can work on the database. However, some utilities, such as `REORG TABLE`, are not compatible with an online backup and cannot be performed while the backup is active.

The following example shows how to run an online DB2 database backup. The example uses the DB2 CLI to back up the `SAMPLE` database.

1. Log in as the database administrator, or with higher authority.
2. Use the `BACKUP DATABASE` command with the `ONLINE` and `USE TSM` options to back up the database.

```
$ db2 backup db sample online use tsm
```

```
Backup successful. The timestamp for this backup image is : 20120805223343
```

Table space backup

The DB2 DBMS facilitates taking a table space-level backup in certain situations where you do not want to do a full database backup. For example, you completed your full database backup and later a user performs a large insert or load to important tables in a specific table space. The user wants the new data backed up. Because you do not want to repeat the full database backup, you can do a table space-level backup. You can perform a table space-level backup either offline or online. You must enable roll-forward recovery to perform an online backup.

The following example shows how to run a table space backup. The example uses the DB2 CLI to back up the `DATA1` table space in the `SAMPLE` database.

1. Log in as the database administrator, or with higher authority.
2. Use the `BACKUP DATABASE` command with the `TABLESPACE` and `USE TSM` options to back up the table space. If you want to do an online backup, include the `ONLINE` option.

```
$ db2 backup db sample tablespace DATA1 online use tsm
```

```
Backup successful. The timestamp for this backup image is : 20120805223621
```

Recovery techniques

The main function of any DBMS is to recover the database from the point where it failed. The DB2 DBMS has various features and functions that you can use to manage your database and perform standard recovery whenever needed. Although it is impossible to recover the database for every failure, you can ensure that recovery is possible for most situations with a minimum loss of data. There are various methods available to restore the database from a backup image.

Version recovery

Use version recovery to restore a previous version of the database from a backup image that was created using the `BACKUP DATABASE` command. When you restore the database, the entire database is rebuilt using a backup of the database made at an earlier point. You can use the backup image of a database to restore the database and create a copy of the database that was there when the backup image was generated.

Only databases that are not enabled for roll-forward recovery can do version recovery. They are said to be *non-recoverable databases* (that is, all the transactions that were held after taking the backup image are lost). All users must be disconnected from the database for version recovery to take place.

Follow these steps to recover a database using version recovery. These steps use the DB2 CLI to recover the `SAMPLE` database.

1. Use the `LIST HISTORY` command with the `BACKUP` option to find the backup image that you want to restore.

```
$ db2 list history backup all for sample
```

```
Op Obj Timestamp+Sequence Type Dev Earliest Log Current Log Backup ID
-----
B D 20120805223343001 N A S0000002.LOG S0000002.LOG
```

```
-----
Contains 7 tablespace(s):
```

```
00001 SYSCATSPACE
00002 USERSPACE1
00003 IBMDB2SAMPLEREL
00004 IBMDB2SAMPLEXML
00005 TEST1
00006 DATA1
00007 SYSTOOLSPACE
```

```
-----
Comment: DB2 BACKUP SAMPLE ONLINE
Start Time: 20120805223343
End Time: 20120805223401
Status: A
-----
```

```

EID: 15 Location: adsm/libtsm.a

Op Obj Timestamp+Sequence Type Dev Earliest Log Current Log Backup ID
-----
B D 20120805223343002 N A S0000002.LOG S0000002.LOG
-----
Contains 7 tablespace(s):

00001 SYSCATSPACE
00002 USERSPACE1
00003 IBMDB2SAMPLEREL
00004 IBMDB2SAMPLEXML
00005 TEST1
00006 DATA1
00007 SYSTOOLSPACE
-----
Comment: DB2 BACKUP SAMPLE ONLINE
Start Time: 20120805223343
End Time: 20120805223401
Status: A
-----
EID: 16 Location: adsm/libtsm.a

```

2. Use the `RESTORE DATABASE` command with the `USE TSM` option to restore the database. The `USE TSM` option tells the DBMS to use the Tivoli Storage Manager API to read the backup file and restore it.

```

$ db2 restore db sample use tsm taken at 20120805223343
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.

```

Database roll-forward recovery

Database roll-forward recovery enhances version recovery by using full database backups and log files to restore a database or selected table spaces to a particular point. Such databases are called *recoverable databases*. With a full database backup image as a base line, if you have all the log files available from the time of backup to the current time, you can apply all the transactions on any or all of the table spaces in the database, up to any point within the time period covered by the logs. With recoverable databases, you can restore the database and apply the logs to the point of failure, and no transactions are lost.

In DB2, roll-forward recovery is specified at the database level and must be enabled explicitly.

Full database restore and roll forward

The database can be restored and then rolled forward to the end of the logs. The following steps use the DB2 CLI to recover the `SAMPLE` database.

1. Use the `RESTORE DATABASE` command with the `USE TSM` option and specify the timestamp of the backup image that needs to be restored.

```

$ db2 restore db sample use tsm taken at 20120805223343
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.

```

2. Use the `ROLLFORWARD DATABASE` command to apply all the logs.

```
$ db2 rollforward db sample to end of logs and stop

                                Rollforward Status

Input database alias              = sample
Number of nodes have returned status = 1

Node number                       = 0
Rollforward status                 = not pending
Next log file to be read           =
Log files processed                 = S0000002.LOG - S0000003.LOG
Last committed transaction         = 2012-08-05-17.06.21.000000 UTC

DB20000I  The ROLLFORWARD command completed successfully.
```

Table space restore and roll forward

With recoverable databases, you can recover the whole database or only the table spaces that need to be recovered. The table space can be recovered either offline or online. You can use a backup image from either a previous table space backup or a previous database backup.

The following steps use the DB2 CLI to recover the DATA1 table space in the SAMPLE database.

1. Use the `RESTORE DATABASE` command with the `TABLESPACE` and `USE TSM` options and specify the timestamp of the backup image.

```
$ db2 "restore db sample tablespace (DATA1) online use tsm taken at 20120805223621"
DB20000I  The RESTORE DATABASE command completed successfully.
```

2. Use the `ROLLFORWARD DATABASE` command with the `TABLESPACE` option to apply all the logs for the recovered table space.

```
$ db2 "rollforward db sample to end of logs and stop tablespace (DATA1) online"

                                Rollforward Status

Input database alias              = sample
Number of nodes have returned status = 1

Node number                       = 0
Rollforward status                 = not pending
Next log file to be read           =
Log files processed                 = -
Last committed transaction         = 2012-08-05-17.06.21.000000 UTC

DB20000I  The ROLLFORWARD command completed successfully.
```

Roll forward to a point in time

This type of roll-forward operation rolls the database or table spaces forward to a specific point. During an online backup, logs are regularly being updated from new transactions. Rollback removes uncommitted transactions in the database. However, when the transaction is committed, rollback is not possible. To remove unwanted data that is mostly due to user errors, you can perform a point-in-time (PIT) recovery. A PIT recovery, however, cannot be done on table spaces that contain system catalogs.

For PIT recovery, you first restore the database or table spaces. Then you apply the logs up to a specific point in time. In the following example, PIT recovery is performed for the DATA1 and TEST1 table spaces in the SAMPLE database.

1. Use the `RESTORE DATABASE` command with the `TABLESPACE` and `USE TSM` options and specify the timestamp of the backup image.

```
$ db2 "restore db sample tablespace (DATA1, TEST1) online use tsm
taken at 20120805223343"
DB20000I The RESTORE DATABASE command completed successfully.
```

2. Use the `ROLLFORWARD DATABASE` command with the `TABLESPACE` option to apply the logs for the recovered table spaces up to a specific point in time. Specify the point in time in coordinated universal time (UTC) format. The point in time must be greater than the minimum recovery time.

```
$ db2 "rollforward db sample to 2012-08-05-16.26.48.000000 and stop
tablespace (DATA1, TEST1) online"
```

Rollforward Status

```
Input database alias           = sample
Number of nodes have returned status = 1

Node number                    = 0
Rollforward status             = not pending
Next log file to be read       =
Log files processed             = -
Last committed transaction     = 2012-08-05-17.06.21.000000 UTC
```

```
DB20000I The ROLLFORWARD command completed successfully.
```

The two table spaces are now in backup pending state (0x0020).

```
$ db2 list tablespaces show detail
```

```
.....
.....
Tablespace ID                 = 5
Name                          = TEST1
Type                          = Database managed space
Contents                       = All permanent data. Large table space.
State                          = 0x0020
  Detailed explanation:
    Backup pending
Total pages                    = 4096
Useable pages                  = 4064
Used pages                     = 96
Free pages                     = 3968
High water mark (pages)       = 96
Page size (bytes)             = 8192
Extent size (pages)           = 32
Prefetch size (pages)         = 32
Number of containers           = 1

Tablespace ID                 = 6
Name                          = DATA1
Type                          = Database managed space
Contents                       = All permanent data. Large table space.
State                          = 0x0020
.....
.....
```

3. Use the `BACKUP DATABASE` command with the `USE TSM` and `TABLESPACE` options to back up the table spaces.

```
$ db2 "backup db SAMPLE tablespace (DATA1, TEST1) online use tsm"
```

```
Backup successful. The timestamp for this backup image is : 20120806115021
```

Now you can successfully access and work on the DATA1 and TEST1 table spaces in the SAMPLE database.

Conclusion

This article introduced you to how you can use Tivoli Storage Manager to back up and restore a DB2 database. You learned how to configure the Tivoli Storage Manager client and server, and saw considerations that you must be aware of during the initial setup. The article also provided step-by-step instructions that showed you how to perform backup operations and how to perform restore and roll forward operations using a backup image stored in the Tivoli Storage Manager server. Now that you understand how to perform these operations, you can take advantage of Tivoli Storage Manager data-management features for your DB2 databases.

Related topics

- [Tivoli Storage Manager](#) provides a wide range of storage management capabilities.
- [Tivoli software from IBM](#) is an excellent starting point for information about Tivoli products and services.
- See [DBA Central](#) to expand your database administration and tuning skills.

© Copyright IBM Corporation 2012

(www.ibm.com/legal/copytrade.shtml)

[Trademarks](#)

(www.ibm.com/developerworks/ibm/trademarks/)