

## Chapter 11: Database Recovery

IBM DB2 Universal Database V8.1  
Database Administration Certification Preparation Course

Maintained by Clara Liu

IBM Software Group

## Objectives

- In this section, we will cover:
  - ▶ Types of Recovery
  - ▶ Database Logging
  - ▶ Backup Utility
  - ▶ Restore Utility
  - ▶ Database Backup and Recovery
  - ▶ Table Space Backup and Recovery
  - ▶ Recovery History File
  - ▶ Dropped Table Recovery
  - ▶ Quiesce Command
  - ▶ High Availability Support

## Chapter 11: Database Recovery

### Types of Recovery

#### Database Logging

Database Backup and Recovery

Table Space Backup and Recovery

Other Backup and Recovery Considerations

High Availability Support

IBM Software Group

## Types of Recovery

- Crash / Restart Recovery
  - ▶ An automatic database recovery if the database was terminated abnormally (e.g. power failure)
  - ▶ Uses the RESTART DATABASE command or the automatic restart enable configuration parameter (autorestart) to protect a database from being left in an inconsistent state
  - ▶ Example:
    - RESTART DATABASE sample
- Version/Image Recovery
  - ▶ Uses the BACKUP command in conjunction with the RESTORE command to put the database in a state that was previously saved
  - ▶ Used for nonrecoverable databases or databases for which there are no archived logs
- Rollforward Recovery
  - ▶ Uses the BACKUP command in conjunction with the RESTORE and ROLLFORWARD commands to recover a database or table space to a specified point in time
- The logging mechanism used in the database determines whether the database is recoverable or nonrecoverable
- Recoverable databases are databases that can be restored by CRASH, VERSION, or ROLLFORWARD recovery techniques
- Nonrecoverable databases only support CRASH and VERSION recovery

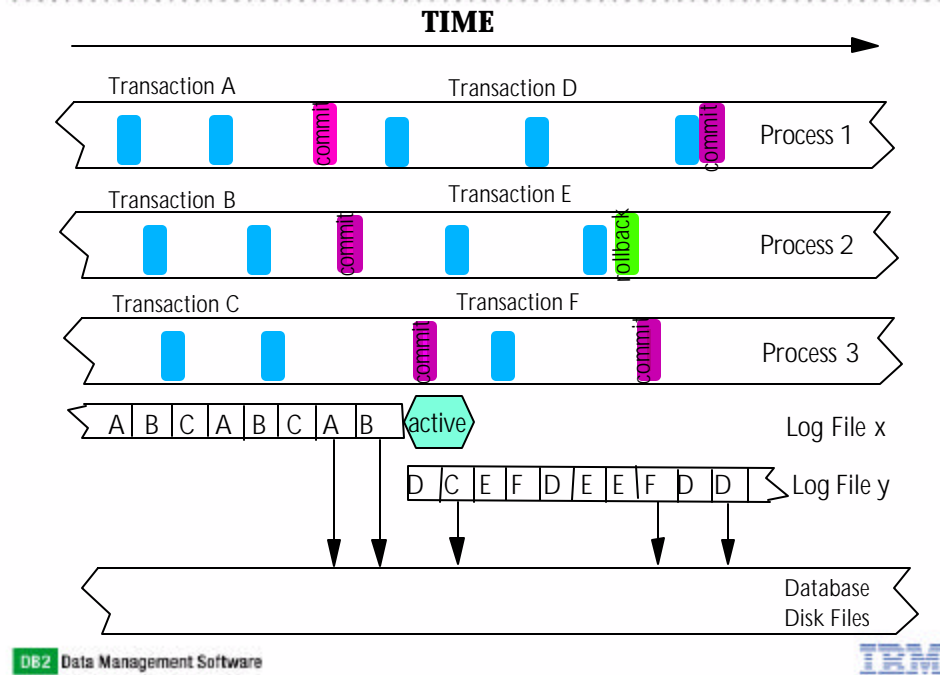
## Transaction Log Files

- Logs keep record of all changes made to database objects and data
  - ▶ Maximum log space allowed is 256GB
- Logs can be written to raw devices
  - ▶ Use the database configuration parameter NEWLOGPATH to point to the raw device
- Transaction log files are used for:
  - ▶ Rollback
    - Uses log files to terminate a unit of work and back out any changes
  - ▶ Crash Recovery
    - Consists of 2 phases
    - Reapply all transactions (regardless of commit)
    - Rollback changes NOT committed
    - Database configuration parameter AUTORESTART (default ON)
  - ▶ Roll Forward Recovery
    - Command can be applied to database or table space
    - Minimum Point In Time (PIT) for table spaces ???

## Transaction Log Files

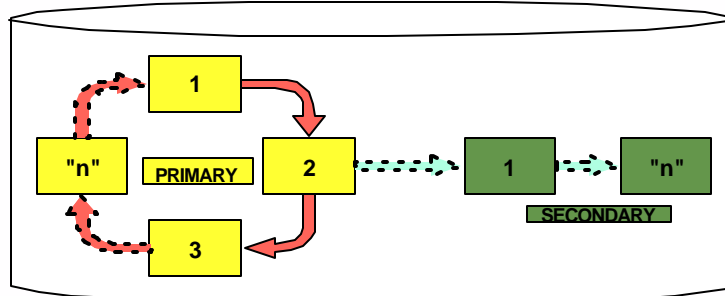
- Log files are stored in the path specified under the 'Path to log files' entry in the database configuration file
- The NEWLOGPATH database configuration parameter can be used to change the log file location
- The numbering scheme for logs starts with S0000000.LOG, and continues through S9999999.LOG, accommodating a potential maximum of 10 million log files
- Log Buffers
  - ▶ All changes are written to log buffers first before being flushed to disk
  - ▶ Size of log buffer is defined with the database configuration parameter LOGBUFSZ
  - ▶ LOGBUFSZ is allocated from a memory area called the database heap (DBHEAP)

## Transaction Log File Usage



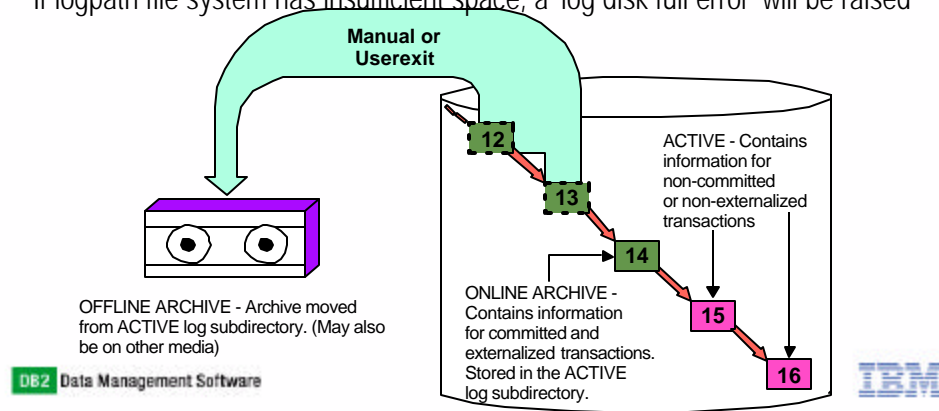
## Circular Logging

- Default logging method
- Primary log files used to record all changes; reused when changes are committed
- Secondary log files allocated when limit of primary log files reached
  - Secondary log files are not used if logging uses raw devices
- Crash and version recovery possible; roll-forward recovery not possible
- If both primary and secondary log limit is reached, an error code is returned
- If logpath file system has insufficient space, a 'log disk full error' will be raised
- Related database configuration parameters: logprimary, logsecond, logfilesiz



## Archive Logging

- Archive logging is used specifically for rollforward recovery
- Enabled with the LOGRETAIN and/or USEREXIT database configuration parm
- Archived logs are logs that are no longer active (i.e. not required for crash recovery)
- Log files are not deleted; they are kept online or offline
- Rollforward recovery and online backup are possible
- If both primary and secondary log limit is reached, an error code is returned
- If logpath file system has insufficient space, a 'log disk full error' will be raised



## How are retained logs handled ?

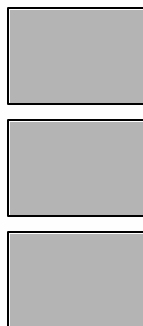
- With log retention, all logs will be kept in the log path unless userexits are enabled or they are moved manually
- Userexits are used to archive the log files to another path/drive or tape device
- Logs will be archived once they are full, they may still be active
- They will be marked as "reusable" once they are in active (i.e. no longer required for restart / recovery)

## Userexit

- The userexit will be called by the DB2 system controller for every log file as soon as it is full
- When the log file is no longer required for restart / recovery DB2 will mark the log file as "reusable"
- Any time after this point the log file can be renamed to one of the new (higher numbered) log files
- During roll forward, userexit may be called to get the log file if it is not in the current log path
  - ▶ Only done for full database restore, not table space
- Sample userexits included with DB2
  - ▶ Can be modified for any installation
  - ▶ Can be found under `../sqlib/samples/c/db2uext2`

## Userexit

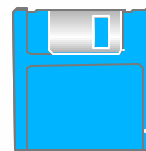
### Primary Logs



Userexit



DB2

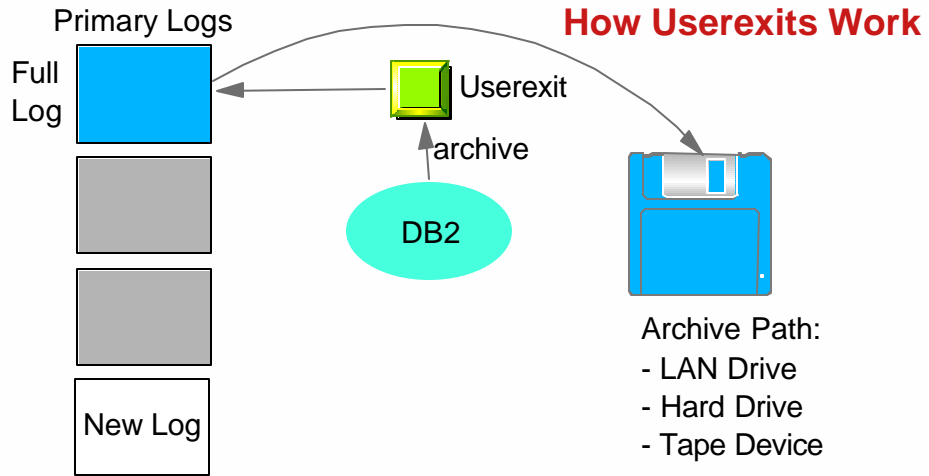


Archive Path:

- LAN Drive
- Hard Drive
- Tape Device
- ADSM

### A Typical Userexit Environment

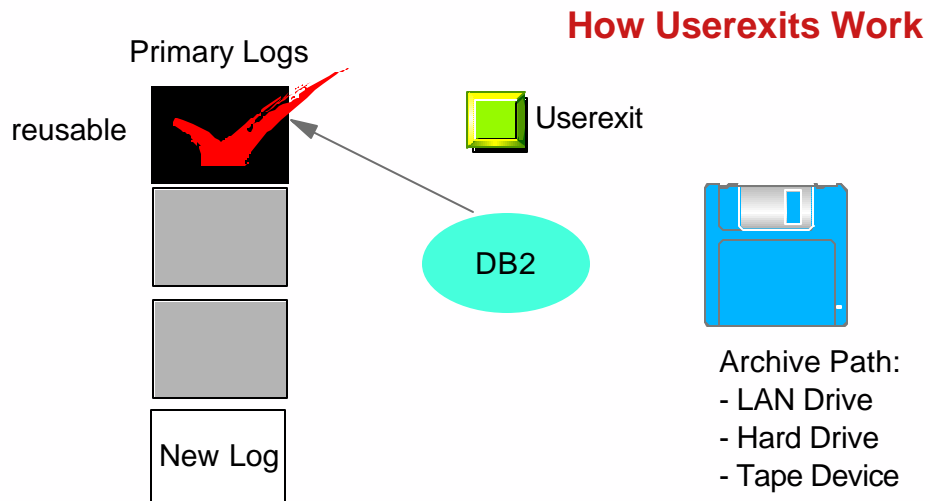
## Userexit



DB2 will signal the userexit to copy a log file to the specified archive path once the log file becomes full. After the log file is copied, a new log file is created.

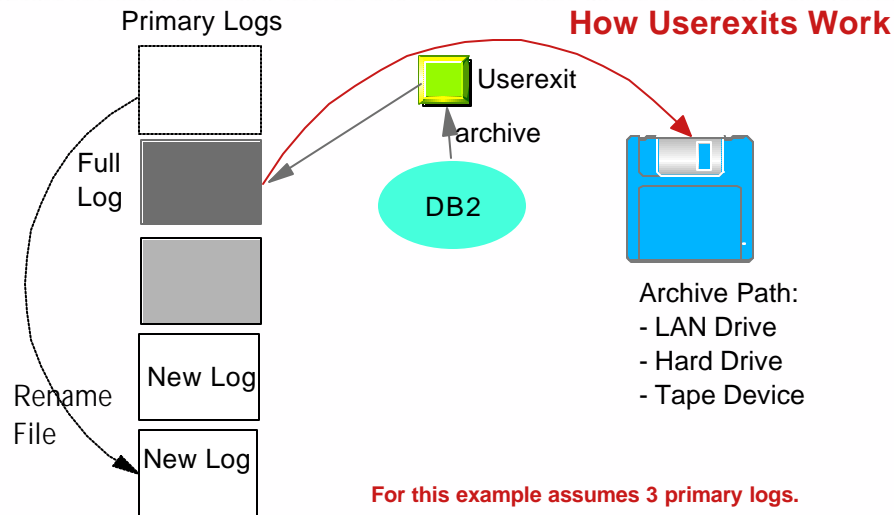
**For this example assumes 3 primary logs.**

## Userexit



DB2 will mark the log file as "reusable/renamable" once the log file is no longer needed for restart/recovery.

## Userexit



DB2 will signal the userexit to copy a log file to the specified archive path once the log file becomes full. After the log file is copied, a new log file is created. In this case, since a previous log file has been marked reusable, it will be renamed. This saves a lot of I/O.

DB2 Data Management Software



## Block Transactions When Log Directory is Full

- If the log path is full and no new log files can be created to record database changes, the transaction will receive a log disk full error and will be rolled back
- To increase database availability, set the database configuration parameter `BLK_LOG_DSK_FUL` to YES so that DB2 will not report any error because of disk full in the active log path
  - ▶ DB2 will attempt to create the log file again after 5 minutes
  - ▶ DB2 will continue this behavior until log is successfully created
  - ▶ DB2 will write a message to the administration notification log if log creation failed
- Until the log file is successfully created, any user application that attempts to update table data will not be able to commit transactions
  - ▶ Read-only queries are normally not affected
  - ▶ Read-only queries may block if reading query is dependent on rows being locked by an update transactions
- `BLK_LOG_DSK_FUL` is not enabled by default

DB2 Data Management Software





## Infinite Active Logging

- Allow a single transaction to use an infinite number of log files
- An active unit of work is able to span the primary logs and archive logs
- Enable infinite active logging by setting database configuration parameter logsecond to -1
- Userexit is used to archive log files, therefore userexit has to be enabled for infinite active logging
- If DB2 needs to read log data from a log file (e.g. rollback) but it is not in the active log path, DB2 will invoke the userexit program to retrieve the log file from the archive to the active log path
  - ▶ DB2 will manage the retrieval, caching, and removal of these log files as required
- Performance considerations:
  - ▶ Rollback, crash recovery could be slow due to the need to retrieve log files from the archive
  - ▶ A warning is written to the administration notification log indicating that the current set of active units of work has exceeded the primary log files

## Log Mirroring

- The MIRRORLOGPATH database configuration parameter allows the database to write an identical second copy of log files to a different path
- Recommended to place the secondary log path on a physically separate disk
- If an error is encountered on either path, that log path will no longer be used until the DB attempts to access the next log file
- Mark the failing path as "bad"
- Message written to the administration notification log
- Future log records will be written only to the remaining log path
- No attempt to synchronize the log paths but DB2 keeps information about access errors that occur, so that the correct paths are used when log files are archived
- if a subsequent failure occurs while writing to the remaining "good" path, the database shuts down

## On Demand Log Archive


- After an online backup is complete, DB2 will now close the current active log
- Use the ARCHIVE LOG command to close and archive an active log for a recoverable database at any time
- This allows you to collect a complete set of log files up to a known point, and then to use these log files to update a standby database
- Issuer of command cannot have a connection to the specified database, although other users may be connected
- Performance may be impacted during execution of the command
- If other applications have transactions in progress, a slight performance degradation will be noticed as the ARCHIVE LOG command flushes the log buffer to disk
- If user exits are enabled, an archive request is issued after the logs are closed and truncated
- Authority Required
  - ▶ User must have DBADM, SYSMANT, SYSCTRL or SYSADM authority
- Example:
  - ▶ ARCHIVE LOG FOR DATABASE sample

**DB2** Data Management Software



**DB2** Data Management Software



 business software

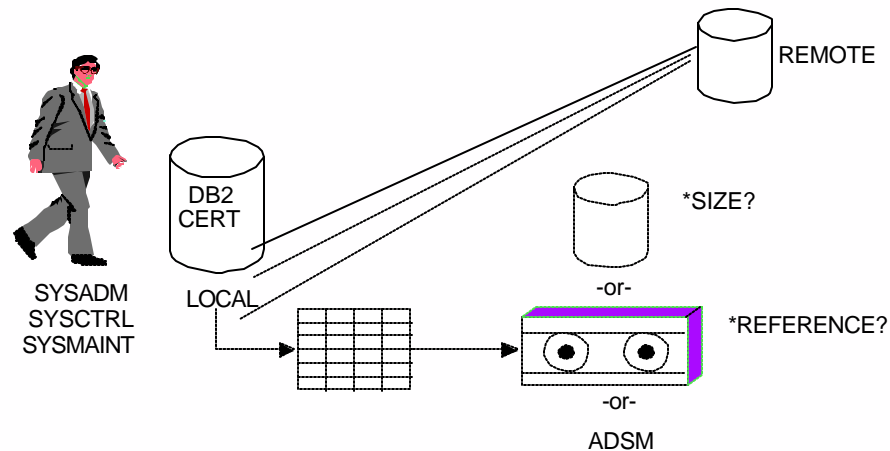
## Chapter 11: Database Recovery

- Types of Recovery
- Database Logging
- Database Backup and Recovery**
- Table Space Backup and Recovery
- Other Backup and Recovery Considerations
- High Availability Support

IBM Software Group

## Backup Utility

- Creates a backup copy of a database or a table space
- Offline and Online backups supported
- SYSADM, SYSCTRL, or SYSMANT authority is required



DB2 Data Management Software



## Backup a Database

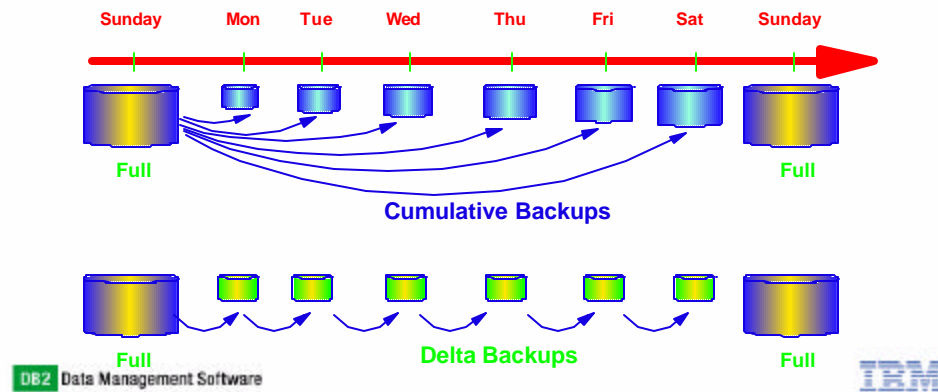
- Full offline database backup
  - ▶ Exclusive database lock is required, no other database operation can be performed
  - ▶ Example:  
**BACKUP DATABASE sample**  
**TO /db2backup/dir1, /db2backup/dir2**  
**WITH 4 BUFFERS BUFFER 4096**  
**PARALLELISM 2**
- Full online database backup
  - ▶ Database must have LOGRETAIN and/or USEREXIT enabled
  - ▶ Other database operations are allowed
  - ▶ Transaction log files are used to capture changes during the online backup
  - ▶ Example:  
**BACKUP DATABASE sample ONLINE TO /dev/rdir1, /dev/rdir2**

DB2 Data Management Software



## Incremental Backup

- Two kinds of incremental backups
  - ▶ Incremental - backup all of the data that has changed since the last full backup
  - ▶ Delta - backup all of the data that has changed since the last successful full, incremental, or delta backup
- Supports both database and table space
- Smaller backup size, reduced amount of data read during backups
- Ability to recover without doing frequent full backups or lengthy log processing



## Backup Files

- File name for images on disk or diskette contains:
  - ▶ database alias
  - ▶ type of backup (0=FULL, 3=TS, 4=Copy from LOAD)
  - ▶ instance name
  - ▶ database node (always 0 for single partition database)
  - ▶ timestamp of backup
  - ▶ sequence number
- Exact naming convention varies slightly by platform
- Tape images are not named, but contain the same information in the backup header for verification purposes
- Use the db2ckbcp utility to display information about existing backup images
  - ▶ Test the integrity of a backup image and determine whether or not it can be restored
  - ▶ Display information that is stored in the backup header
  - ▶ Display information about the objects and the log file header in the backup image

## Example of Backup Filenames

### Intel

Alias Instance Year Day Minute Sequence  
**DBALIAS.0\DB2INST.0\19960314\131259.001**  
Type Node Month Hour Second

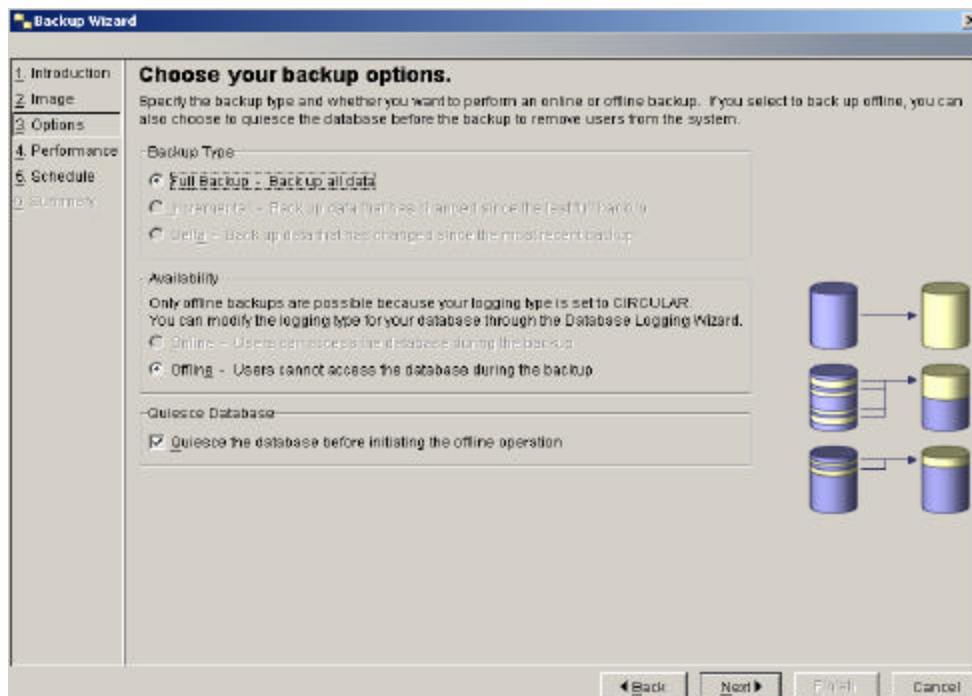
### UNIX

Alias Instance Year Day Minute Sequence  
**DBALIAS.0.DB2INST.0.19960314131259.001**  
Type Node Month Hour Second

DB2 Data Management Software

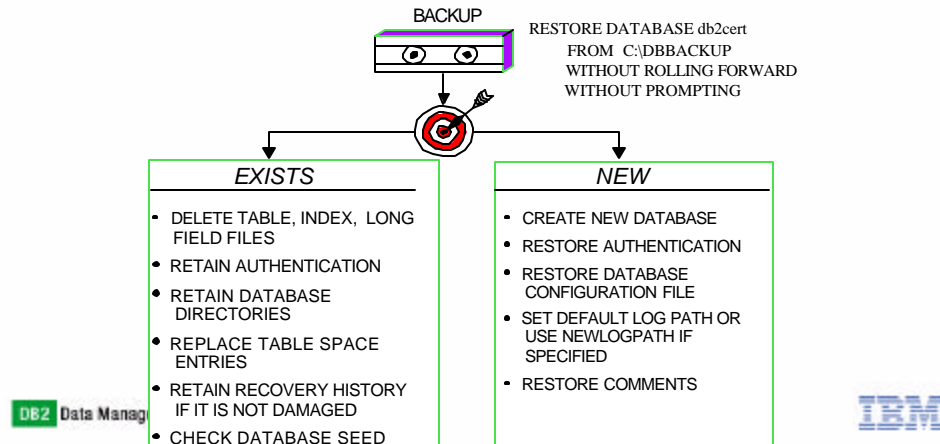


## Backup Using the Control Center



## Restore Utility

- Rebuilds a database or table space that has been backed up using the backup utility
- Incremental restore is supported
- Require SYSADM, SYSCTRL, or SYSMAINT authority to restore to an existing database from a full database backup
- Require SYSADM or SYSCTRL authority to restore to a new database



## Restore a Database

- Full database restore must be offline
- Example:

```
RESTORE DATABASE sample
FROM /dev/rdir1, /dev/rdir2
TAKEN AT xxx
INTO sample
NEWLOGPATH /db2/sample/newlog
REPLACE EXISTING
WITHOUT PROMPTING
```
- TAKEN AT
  - ▶ Specify the time stamp of the database backup image
  - ▶ Backup image timestamp is displayed after successful completion of a backup
- NEWLOGPATH
  - ▶ Specify the directory where active log files will be stored after the restore
  - ▶ Can be used when the log path in the backup image is no longer valid after the restore
- WITHOUT PROMPTING
  - ▶ Specifies that the restore operation is to run unattended

## Roll Forward a Database

- To perform a rollforward recovery, execute the ROLLFORWARD command after the database backup image is restored
- If the WITHOUT ROLLING FORWARD option is specified in the RESTORE DATABASE command, it is equivalent in performing a version recovery
- Note that the WITHOUT ROLLING FORWARD option cannot be used if the database backup was taken online
- Require SYSADM, SYSCTRL, or SYSMAINT to roll forward a database
- During roll forward processing, DB2 will:
  - ▶ Look for the required log file in the current log path
  - ▶ If found, reapply transactions from the log file
  - ▶ If log file is not found in the current path, the userexit is called to retrieve the log file from the archive path
  - ▶ The userexit will only be called to retrieve log files if rolling forward a full database restore
  - ▶ Once the log is in the current log path, the transactions will be reapplied
- Example:
  - ▶ ROLLFORWARD DATABASE sample TO END OF LOGS AND COMPLETE
  - ▶ ROLLFORWARD DATABASE sample TO timestamp1 USING LOCAL TIME
  - ▶ ROLLFORWARD DATABASE sample COMPLETE

DB2 Data Management Software



## Rollforward a Database

- Example:
  - ▶ ROLLFORWARD DATABASE sample TO END OF LOGS AND COMPLETE
  - ▶ ROLLFORWARD DATABASE sample TO timestamp1 USING LOCAL TIME AND COMPLETE
- Specify the recovery timestamp in one of the following ways:
  - ▶ Timestamp expressed in Coordinated Universal Time (CUT)
    - CUT can be evaluated by subtracting CURRENT TIMEZONE from the local time
  - ▶ Local time
- Example:
  - ▶ ROLLFORWARD DATABASE sample TO cut-timestamp
  - ▶ ROLLFORWARD DATABASE sample COMPLETE
- Need to specify COMPLETE or STOP to take the database out of the roll forward pending state
  - ▶ Include keyword COMPLETE or STOP in the ROLLFORWARD command or execute it as a separate command
- Once the ROLLFORWARD command is issued with the COMPLETE or STOP option, it is not possible to roll forward additional changes

DB2 Data Management Software



## Restore a Database with Incremental Backup Images

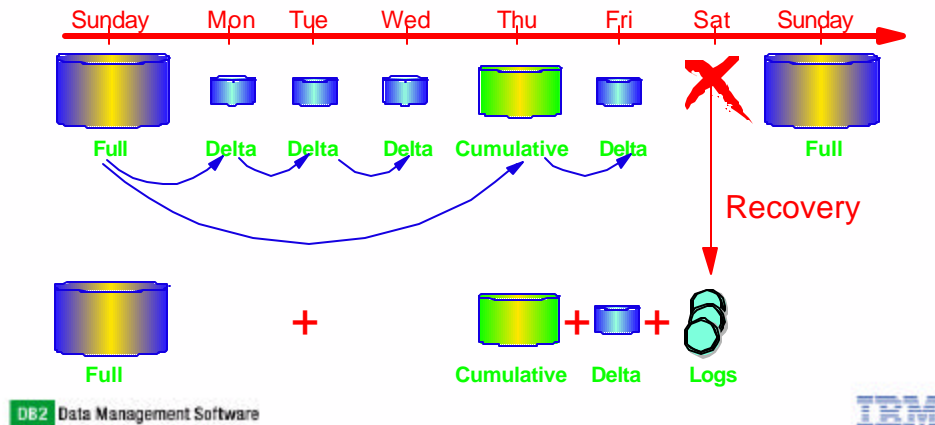
- Full recovery requires full backup, last cumulative backup, all delta backups, and any log files before failure

- Example:

```
RESTORE DATABASE sample INCREMENTAL AUTOMATIC
```

```
FROM /db2backup/dir1, /db2backup/dir2 ;
```

```
ROLLFORWARD DATABASE sample TO END OF LOGS AND COMPLETE ;
```



## INCREMENTAL Restore Options

- INCREMENTAL

- ▶ Without additional options, it specifies a manual cumulative restore
- ▶ During manual restore the user must issue each restore command manually for each image involved in the restore

- ▶ Example:

- RESTORE DB prod INCREMENTAL TAKEN AT friday INTO test
- RESTORE DB prod INCREMENTAL TAKEN AT monday INTO test
- RESTORE DB prod INCREMENTAL TAKEN AT wednesday INTO test
- RESTORE DB prod INCREMENTAL TAKEN AT friday INTO test

- INCREMENTAL AUTOMATIC

- ▶ Specifies an automatic cumulative restore operation
- ▶ The restore utility uses the backup history file to backup image needed

- ▶ Example:

- RESTORE DB prod INCREMENTAL AUTOMATIC TAKEN AT friday INTO test

- INCREMENTAL ABORT

- ▶ Abort an in-progress manual cumulative restore



## Redirected Restore

- Record of table space containers is kept during backup
- If containers not available during restore, new containers can be specified through the use of the restore utility
- Use the REDIRECT option in the RESTORE command and the SET TABLESPACE CONTAINERS command

### **EXAMPLE**

```
RESTORE DATABASE DB2CERT FROM C:\DBBACKUP  
INTO NEWDB REDIRECT WITHOUT ROLLING FORWARD
```

SQL1277N Restore has detected that one or more table space containers are inaccessible, or has set their state to 'storage must be defined'.

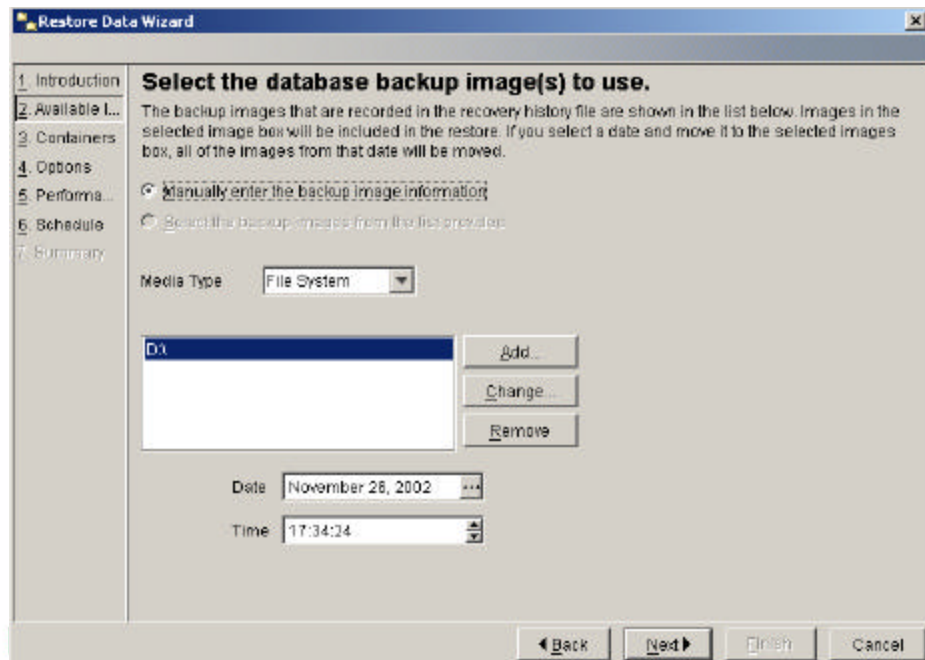
DB20000I The RESTORE DATABASE command completed successfully.

```
SET TABLESPACE CONTAINERS FOR 0 USING (FILE "d:\newdb\cat.dat" 5000)  
SET TABLESPACE CONTAINERS FOR 1 USING (PATH "d:\newdb2")  
RESTORE DATABASE DB2CERT CONTINUE
```

## Restoring a Backup on a Different Platform or Word Size

- To make a copy of a database created on a different operating system or word size (32-bit or 64-bit) use the db2move utility EXCEPT:
  - ▶ Database created and backed up in 32-bit Windows can be restored to a 64-bit Windows platform, or vice versa
  - ▶ Database created and backed up in 32-bit Linux can be restored to a 64-bit Linux platform, or vice versa
  - ▶ Database backups created on AIX, HP, and Sun operating systems in 32-bit or 64-bit, may be restored on AIX, HP, and Sun operating systems in either 32-bit or 64-bit
    - The backup image format in V8 is the same for AIX, HP, and Sun platforms
- Incremental and delta images cannot be restored when there is a difference in operating systems or word size (32-bit or 64-bit)
- Following a successful restore from one environment to a different environment, no incremental or delta backups are allowed until a non-incremental backup is taken

## Restore Using the Control Center



**DB2** Data Management Software

**IBM**

business software

## Chapter 11: Database Recovery

- Types of Recovery
- Database Logging
- Database Backup and Recovery
- Table Space Backup and Recovery**
- Other Backup and Recovery Considerations
- High Availability Support

IBM Software Group

## Table Space Backup

- To perform a table space level backup, the database must be enabled for forward recovery
- Use the BACKUP DATABASE command with the TABLESPACE option:  
BACKUP DATABASE sample  
TABLESPACE ( syscatspace, userspace1, userspace2 ) ONLINE  
TO /db2tbsp/backup1, /db2tbsp/backup2
- Support both online and offline backup
- Concurrent backup operations on the same table space are not permitted
- A backup operation will fail if a list of the table spaces to be backed up contains temporary table space
- Table space level incremental backup is supported

## Table Space Recovery

- To restore a table space, use the RESTORE DATABASE command with the TABLESPACE option:  
RESTORE DATABASE sample TABLESPACE ( userspace1 ) ONLINE  
FROM /db2tbsp/backup1, /db2tbsp/backup2
- After a table space is restored, it is always in rollforward pending state
- Table space must be rolled forward to a minimum Point In Time (PIT)
- Minimum PIT ensures the table space and logs agree with what is in the system catalogs
- Minimum PIT can be obtained from:
  - ▶ LIST TABLESPACES SHOW DETAIL command OR
  - ▶ Table space snapshot - GET SNAPSHOT FOR TABLESPACE ON dbname
- System catalog table space must be rolled forward to the end of logs and in offline mode

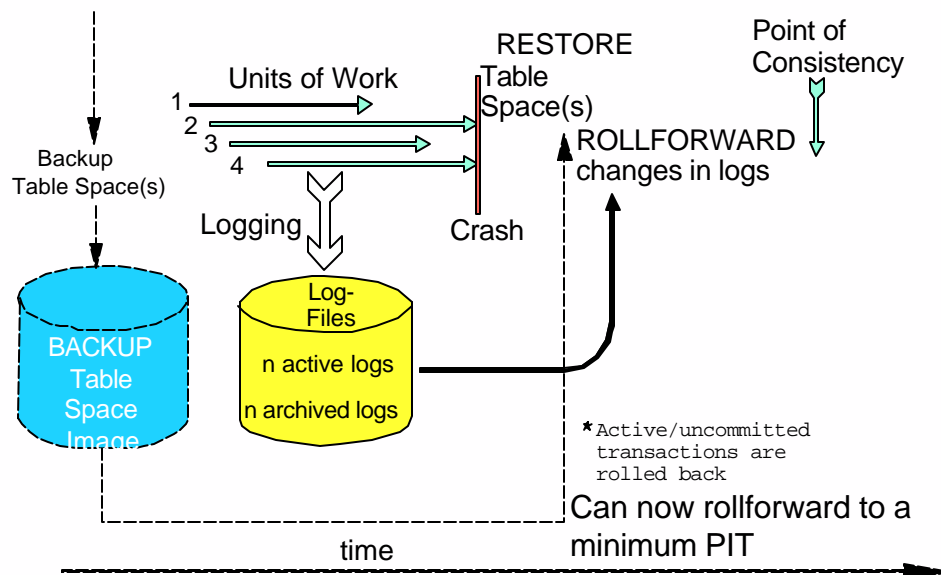
## Table Space Rollforward

- Only log files required to recover the table space are processed
  - Rollforward will skip over log files that are not required
- Minimum PIT is updated when DDL statements are run against the table space, or against tables in the table space
- Example:
 

```
ROLLFORWARD DATABASE sample
TO END OF LOGS AND COMPLETE
TABLESPACE ( userspace1 ) ONLINE
```
- May use the QUERY STATUS option to list log files that the database manager has rolled forward, the next archive file required, and the timestamp of the last committed transaction since rollforward processing began
- Example:
 

```
ROLLFORWARD DATABASE sample QUERY STATUS USING LOCAL TIME
```
- After a table space point-in-time rollforward operation completes, the table space is put in backup pending state
- Must take a backup of the table space, because all updates made to it between the point in time recovered and the current time have been removed

## Table Space Recovery

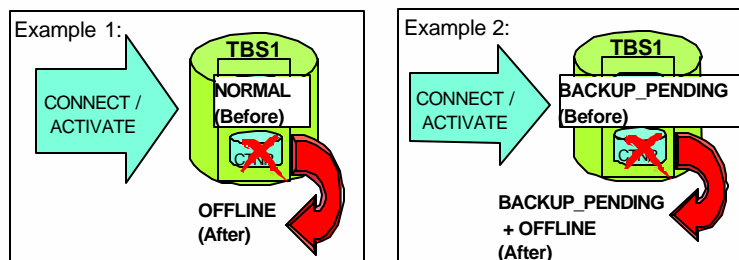


## Table Space Change History File

- By default, DB2 will process all log files even if they do not contain log records that affect that table space
- To skip the log files that does not contain any log records of affecting the table space, additional information is collected, set the registry variable `DB2_COLLECT_TS_REC_INFO = ON`
- The table space change history file - `DB2TSCHG.HIS`, keeps track of which logs should be processed for each table space
- View the contents of the change history file with the `db2logsForRfwd` utility
- Use the `PRUNE HISTORY` command to remove entries in the table space change history file

## Table Space in OFFLINE State

- If container not accessible, table space placed in OFFLINE state
- Rest of the database and table spaces are still accessible
- If you can correct the problem, table space can be made available with:
  - ▶ `ALTER TABLESPACE userspace1 SWITCH ONLINE`
- Supported in both circular and archival logging



- At database restart time, table space with bad containers can be placed in drop pending state with this command:
  - ▶ `RESTART DATABASE dbname DROP PENDING TABLESPACES`
- Can only drop a table space that is in drop-pending state

## Summary of Table Space States

- Rollforward pending
  - ▶ After a table space restore, the table space is placed in rollforward pending
  - ▶ Rollforward to the end of the logs or to a point in time or specify COMPLETE to get it out of the pending state
- Rollforward-in-progress
  - ▶ A table space is put in this state when a rollforward on that table space is in progress
- Restore pending
  - ▶ If a rollforward operation on that table space encounters an unrecoverable error, in which case the table space must be restored and rolled forward again
- Backup pending
  - ▶ A table space is put in this state after a point-in-time rollforward operation, or after a load operation with the no copy option
  - ▶ The table space must be backed up before it can be used
  - ▶ If it is not backed up, the table space cannot be updated, but read only operations are allowed
- Drop pending
  - ▶ Nothing can be done with the table space except to drop it
- Offline
  - ▶ Table space is offline because the database manager cannot access one or all of its containers

DB2 Data Management Software



DB2 Data Management Software



business software

## Chapter 11: Database Recovery

- Types of Recovery
- Database Logging
- Database Backup and Recovery
- Table Space Backup and Recovery
- Other Backup and Recovery Considerations**
- High Availability Support

IBM Software Group

## Recovery History File

- Created with each database and is automatically updated whenever there is a:
  - ▶ Database or table spaces are backed up
  - ▶ Database or table spaces are restored
  - ▶ Database or table spaces are rolled forward
  - ▶ Table space is created
  - ▶ Table space is altered
  - ▶ Table space is quiesced
  - ▶ Table space is renamed
  - ▶ Table space is dropped
  - ▶ Table is loaded
  - ▶ Table is dropped
  - ▶ Table is reorganized
- To see the entries in the recovery history file, use the LIST HISTORY command:
  - ▶ LIST HISTORY BACKUP SINCE timestamp FOR sample
  - ▶ LIST HISTORY ROLLFORWARD CONTAINING user1.staff FOR sample
  - ▶ LIST HISTORY REORG ALL

## Managing Recovery History File

- Data contained in the history is used directly by automatic restore operations
- If the backup images required by the automatic restore operations have been relocated since they were created, recommended to update the current location with the UPDATE HISTORY command
  - ▶ UPDATE HISTORY FOR <backup image in timestamp format>
    - WITH LOCATION <new location> DEVICE TYPE <device type>
- Size of the file is controlled by the rec\_his\_retentn configuration parameter
- Specifies a retention period (in days) for the entries in the file
- To manually delete all entries in the recovery history file with timestamps equal to or less than the one specified, use the PRUNE command:
  - ▶ PRUNE HISTORY timestamp
- Every backup operation (database, table space, or incremental) includes a copy of the recovery history file
- Dropping a database deletes the recovery history file
- Restoring a database to a new location restores the recovery history file
- Restoring to an existing database does not overwrite the existing history recovery file unless the file that exists on disk has no entries or damaged

## Dropped Table Recovery

- Permits recovery of accidentally dropped tables
- Requires setting DROPPED TABLE RECOVERY option on table space
- Need to restore database or table space to recover dropped table
- DDL for dropped table is stored in Recovery History File
- Extract using the LIST HISTORY DROPPED TABLE command
- Involves generating a 'loadable' image of the table and then loading delimited ASCII data back into table
- Image is generated by:
  - ▶ Restore a database or table space backup image
  - ▶ Roll forward the database or table space with the RECOVER DROPPED TABLE option, this will export data of the dropped table to file(s)
- Recreate the table
- Import the data
- Triggers, summary tables, unique constraints, referential constraints, and check constraints not recovered

## Quiesce an Instance or a Database

- The QUIESCE command forces all users off the specified instance and database and puts it into a quiesced mode
- To quiesce an instance, SYSADM or SYSCTRL authority is required
- To quiesce a database, SYSADM or DBADM authority is required
- In quiesced mode, only users with certain authority are allowed to attach or connect to the instance/database, administrative and maintenance tasks (such as database backup) that require exclusive access can be performed
  - ▶ Users with SYSADM, SYSMANT, and SYSCTRL authority always have access to an instance while it is quiesced
  - ▶ Users with SYSADM authority always have access to a quiesced database
- Examples:
  - ▶ QUIESCE DATABASE sample IMMEDIATE
    - Quiesce the database sample, all database objects will be in quiesce mode
  - ▶ QUIESCE INSTANCE db2inst1 IMMEDIATE FOR GROUP admingroup
    - Quiesce the instance db2inst1, all databases in the instance will be in quiesce mode
    - All transactions are immediately terminated and rolled back
    - Besides users with SYSADM authority, the group admingroup is also allowed to access the instance while it is quiesced
- To unquiesce:
  - ▶ UNQUIESCE DATABASE sample
  - ▶ UNQUIESCE INSTANCE db2inst1



## Quiesce Table Space for a Table

- A table space can also be quiesced
- Few quiesced mode can be obtained:
  - ▶ QUIESCE TABLESPACES FOR TABLE user1.employee SHARE
  - ▶ QUIESCE TABLESPACES FOR TABLE user1.employee EXCLUSIVE
  - ▶ QUIESCE TABLESPACES FOR TABLE user1.employee INTENT TO UPDATE
  - ▶ QUIESCE TABLESPACES FOR TABLE user1.employee RESET
- QUIESCE SHARE
  - ▶ An intent share locks for the table spaces and a share lock for the table are acquired
  - ▶ The table cannot be changed but other share mode requests to the table and table spaces are allow
- QUIESCE EXCLUSIVE
  - ▶ The transaction requests super exclusive locks on the table spaces and the table
  - ▶ No other access to the table spaces is allowed
- QUIESCE INTENT TO UPDATE
  - ▶ The table spaces are locked in intent exclusive (IX) mode, and the table is locked in update (U) mode
- QUIESCE RESET
  - ▶ To unquiesce the table space and the table

## Index Recreation

- If data in the database is corrupted or damaged, it is possible that the indexes associated are also damaged or marked as invalid by the database manager
- The configuration parameter INDEXREC indicates when the database manager will attempt to rebuild invalid indexes
- INDEXREC is defined in both the database manager and database levels
- There are three possible settings for this parameter:
  - ▶ SYSTEM
    - Can only be specified in the database configuration file
    - It means use the INDEXREC setting specified in the database manager configuration file
  - ▶ ACCESS
    - Invalid indexes are rebuilt when the index is first accessed
  - ▶ RESTART
    - Invalid indexes are rebuilt during database restart via the RESTART DATABASE command

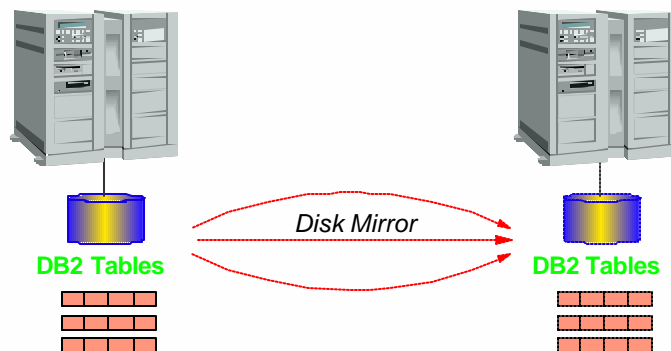
## Chapter 11: Database Recovery

- Types of Recovery
- Database Logging
- Database Backup and Recovery
- Table Space Backup and Recovery
- Other Backup and Recovery Considerations
- High Availability Support**

IBM Software Group

### I/O Suspend, Split/Mirror

- Customer cannot afford doing Off-line or On-line Backups on a TB live database
- On-line Split Mirror
  - split off a consistent mirror of a database while OLTP keeps on running on the primary (live) database
  - only a limited performance hit
  - Backups and System copies can be done from a mirror image



## Making a Clone Database

- Clone the primary database so that it can be used for read only purposes
  - ▶ Running reports
  - ▶ Analysis
- Suspend I/O on primary system:
  - ▶ SET WRITE SUSPEND FOR DATABASE
- Split the mirror
  - ▶ Use operating system level command to split the mirror from the primary database
- Resume I/O on primary system
  - ▶ SET WRITE RESUME FOR DATABASE
  - ▶ Database on the primary system should be back to a normal state

## DB2INIDB Command

- Once a database has been split, the db2inidb utility must be used to indicate the intended use of the database
- Snapshot
  - ▶ Crash recovery is performed - making the database consistent
  - ▶ Database will be fully available for any operations (e.g. backup)  
db2inidb <db\_name> as snapshot
- Standby
  - ▶ Places the database in rollforward pending state
  - ▶ Allow authorized users to perform an off-line backup of either the entire database or a subset of the tablespaces  
db2inidb <db\_name> as standby
- Mirror
  - ▶ Mirrored copy will be used to replace the original database, the database is placed in rollforward pending and the WRITE SUSPEND state is turned off  
db2inidb <db\_name> as mirror