
**Session:B01**

# An A-Z of DB2 for z/OS Backup and Recovery

*Phil Grainger*

IDUG
The Worldwide DB2 User Community

**Monday October 5th 11:30 – 12:30**
**Platform:z/OS**

# Agenda

- Active logs
- Archive logs
- BSDS
- Business requirements
- Catalog and Directory
- Consistency
- Deferred recovery
- Hardware services
- Image copies
- Index copies
- Log tools
- Logging and NOT logging
- Logical recovery
- Mandated requirements
- MERGECOPY
- Mirroring
- Parallelism
- Partitioned objects
- Priorities
- Quiesce
- Restart issues
- Strategies
- SYSCOPY
- System backup/restore

# Active Logs – How many?

- DB2 insists that you have at least TWO logs
  - One is in use
  - One is in reserve, to be switched to when the first fills up
- Ideally these are also duplexed
  - In case of loss or damage
  - DB2 can switch into ‘single logging’ mode
  - Instead of abending

# Active Logs – How many?

- Only 2 logs is dangerous though
  - If one log has not yet been archived
  - And the second fills up
  - DB2 will hang
  - Only a cancel or IRLM will terminate DB2

# Active Logs – How Many?

- So at least 3 would be safer
  - Or even more
  - During recovery, if DB2 needs log records from an archive
  - And the active log that the archive was created from is not yet overwritten
  - DB2 will utilise the “old active” instead of the archive

# Active Logs – Dual logs?

- I often hear that dual logs are not needed
  - As ‘modern DASD duplexes everything anyway’
- If you run in single logging mode
- And DB2 loses access to the log
- DB2 WILL abend
  - He cannot continue without a log
- Do you want to risk that?

# Archive logs

- Archive logs should be kept for “as long as needed”
- To assist in recovery, for example
- Or for auditing purposes
- Missing archive logs will stop a log apply from proceeding

# BSDS

The BSDS contains

- Details of the defined active logs
- All the known archive logs
- All system checkpoints
- Some data sharing information

- Do NOT lose your BSDS!!!

# Business requirements

- You CANNOT build a recovery strategy without knowing what your business needs
- Someone must know:
  - How quickly things need to be back to normal
  - How much data it is acceptable to lose
  - How much impact a backup can have on day-to-day running
- Without this knowledge any strategy is just guesswork

# Business Requirements

- So, how much does downtime cost
  - Per hour for example

# Catalog & Directory backup/recovery

- Recovery of the catalog and directory is made more complex by a defined sequence of object recoveries
  - There is also a defined sequence for the backups!
- These sequences change between different DB2 versions as new catalog objects are introduced

# Catalog & Directory backup/recovery

- The recovery sequence is documented in the Utility Guide
  - Under “Recovering catalog and directory objects”
- The backup sequence does not appear to be documented anywhere
- But it can be derived from the recovery sequence
  - Back things up in the reverse order they need to be recovered

# Change accumulation (ISV)

- Change accumulation takes log accumulation one step further
- Accumulates log records and merges them with an image copy
- The results is a NEW image copy
  - As if it were taken at the end point of the accumulate
- If accumulating to a consistent point
  - The result is a SHRLEVEL(REFERENCE) copy
  - Otherwise a SHRLEVEL(CHANGE) copy

# Change accumulation

- The created copies will be registered in SYSCOPY
  - As if they were taken with the Image Copy utility
- BUT copies are taken WITHOUT any outage to the application concerned
  - Even for SHRLEVEL(REFERENCE)
- The rba at which the copy has been “taken” is the rba that is accumulated to
  - NOT the current rba of the system

# Change accumulation

- Therefore
- It is possible to take a quiesce at a convenient point
- THEN later, when time permits, accumulate to that quiesce rba
- Resulting in a SHRLEVEL(REFERENCE) copy
- Accumulate multiple objects to the same rba
  - To get a consistent SET of copies
  - WITHOUT making all objects unavailable whilst this is done

# Recover with consistency (DB2 9)

- DB2 has always allowed you to recover to ANY RBA (or LRSN) you wish
- It is up to YOU to recover to a sensible (aka consistent) point
- In fact, DB2 (prior to DB2 9) will allow you to recover into the middle of a unit or work
- You could have uncommitted updates in your recovered object
  - It's what you asked for so you got it!

# Recover with consistency (DB2 9)

- DB2 9 modifies the recovery mechanism a little
- If you recover to a point where there are in-flight Units Of Work updating the object(s) you are recovering
- DB2 will scan the log back to the start of those UOWs, undoing the uncommitted updates
- Taking your recovery point back to a point of transactional consistency

# Recover with consistency (DB2 9)

- You will see log scan and log apply phases if this is happening
- Of course , as discussed earlier, transaction consistency does NOT mean business consistency
  - You could be back to a mid-job COMMIT point
- BUT at least you now don't have any uncommitted updates

# Deferred restart

- If your subsystem availability is critical, the above restart delay may be unacceptable
  - Even though things are much faster than they used to be
- In this case, the availability of the subsystem may be more important than the availability of individual tables

# Deferred restart

- If so, dsnzparm can be changed to speficy DEFER,ALL
  - Instead of the more usual RESTART,ALL
- This means that during an abnormal restart
- DB2 will still look for in-flight UOWs
- BUT will not recover them
  - They are placed in LPL instead

# Automation of deferred recovery

- Two dsnzparm parameters allow DB2 to take control of this deferred recovery
- LIMIT BACKOUT
  (zparm LBACKOUT)
- BACKOUT DURATION
  (zparm BACKODUR)

# Hardware services

- Don’t forget the possibilities for hardware assisted backups
  - And recoveries
- DB2 can utilise these for snapshot copies
  - And for BACKUP/RESTORE SYSTEM, of course
- ISVs also make use of these capabilities

# Image Copy

- IBM provides the COPY utility
- This will copy table spaces and copyable indexes
- LISTDEF can be used to copy groups of objects at the same point
  - Perhaps logically related groups
- At it's simplest we can

```
COPY ........ FULL YES|NO
```

- FULL YES gives us a full copy of ALL the pages
- FULL NO only copies pages changed since last copy

# Incremental Copy

- In the past, incremental copies rapidly became slower than full copies
  - Because FULL YES could take advantage of prefetch
- Today, FULL NO can use LIST PREFETCH
- Benchmark to see where the breakpoint is
  - It can still be slower to incrementally copy a large proportion of a page set

# Incremental Copy - Changelimit

- You can also specify

```
... CHANGELIMIT (percent1, percent2)
  ....
```

- If less than *percent1*% of the pages have changed
  NO copy will be taken
- If more than *percent2*% of the pages have changed
  A full copy will be taken
- Otherwise
  An incremental copy will be taken

# Incremental copy - Changelimit

- Be aware of the first point
- It is possible NEVER to have image copies taken if only a very small amount of data changes
- If using CHANGELIMIT, perhaps schedule a regular FULL copy as well

# Index copies

- DB2 Indexes can now be specified as COPY YES
- If so, they may be copied in the same way as table spaces
- Copying of indexes can speed up recovery
  - By allowing index recovery to take place concurrently with table space recovery
  - Writing out a set of index pages can be quicker than rebuilding a complete index structure from scratch

# Index copies

- HOWEVER
- Recovery of Indexes from copies MUST be to the same point as table space recovery
- Otherwise REBUILD INDEX will be needed to synchronise index entries
- If indexes and table spaces are recovered in different jobstreams, REBUILD PENDING may result

# Index copies – Why?

- Recovery of an index is likely to be faster than a rebuild
- Recovery can take place in parallel with table recovery
  - Rebuild must take place AFTER table recovery
- In recovery situations, the most parallelism is the best

# Log tools

- ISV log tools can be INVALUABLE during recoveries
  - To assist with determining recovery points
  - To determine exactly WHAT to recover
  - By providing additional recovery possibilities

# Log accumulation (ISV)

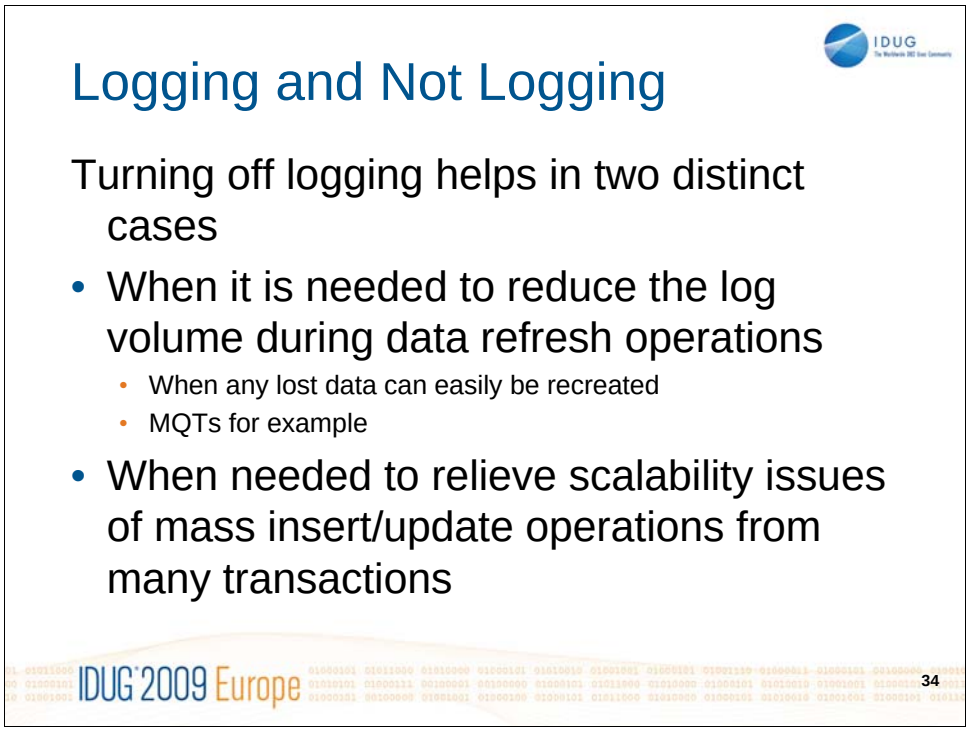
- Log accumulation is the action of collecting log records for specific objects
- And accumulating them in an independent log file
- Some tracking mechanism is needed to allow recovery to “find” these accumulated logs
- This technique (only available from ISVs) can speed up log apply recover of critical objects

# Log accumulation

- If the accumulate is done to a point of consistency
  - Like a quiesce point
- It can be used as a recovery point as well

# Logging and Not Logging

- DB2 already had NOT LOGGED for LOB table spaces
- DB2 9 allows this for XML and normal table spaces as well
  - And propagates this to their indexes
- Objects can be created NOT LOGGED
- Or be altered to/from NOT LOGGED

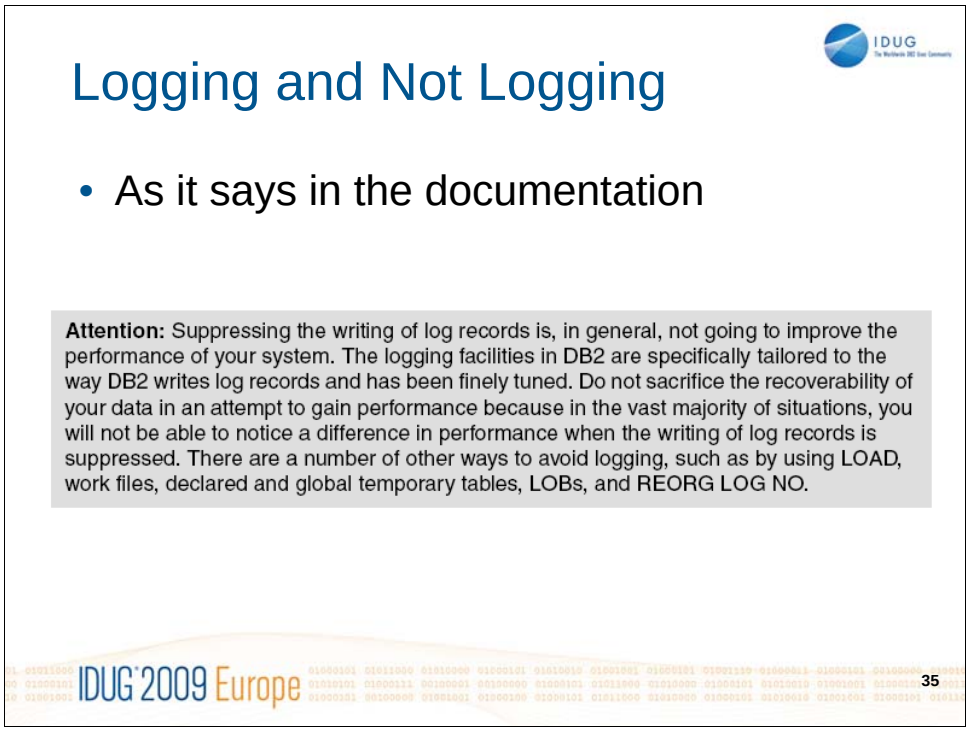



Note though that turning off logging should not be regarded as a way of getting increased performance – in most cases is won't

Te only times that IBM suggest that NOT LOGGED might be beneficial are:

- When it is advantageous to reduce the volume of log data and when any lost data can easily be recreated (such as refreshing summary data or Materialized Query Tables)

- When mass insert/update/delete operations from many transactions are failing to scale due to the level of logging




Read this note carefully before going near NOT LOGGED objects!

# Logical/SQL recovery

- Similar to change backout
- It is possible to read log records to forward recover objects
- Normally, DB2 will do this
- But if you have a badly corrupted file system, an alternative may be needed
- Also, if you are recovering a dropped object
- Changes to internal ids (obid, psid etc.) may make DB2 log apply invalid

# Mandated requirements

- Are there industry or government mandated requirements for recovery
- Do you have to be back in less than a specific time
- Do you have to lose less than a specified amount of data

- If you do, then money should be no object ☺
- BUT you had better be sure you can fulfil these requirements

# Incremental copy – MERGECOPY

- The MERGECOPY utility can do two things

1. Merge a series of incremental copies to form one consolidated incremental copy
2. Merge incremental copies with a full copy to make a new full copy

- Either of these techniques will help to speed up recovery where multiple incremental copies are involved

# MERGECOPY

- MERGECOPY can also be run on lists of objects created with LISTDEF
- Bear in mind that pages will need to be sorted and merged, so this is not a simple process
- But it does happen before you need the data for recovery
  - Otherwise all the sorting and merging takes place at recovery time

# Mirroring

- One word on mirroring
  - The synchronous or near-synchronous copying of DASD to a second location
- This is often viewed as a universal panacea to recovery
- “I don’t need to worry – All my data is mirrored”

# Mirroring

- But what about data corruptions?
- Many (most?) recoveries are done purely to recover from logically damaged data
- Mirroring just spreads the corruption across your duplicated DASD
- So mirroring only protects from PHYSICAL damage or loss
- And if you accidentally drop an object
  - The drop will also be mirrored!

# Parallelism in copy

- Either code parallelism into the jobs that you create
  - And run multiple jobs simultaneously
- Or utilise the parallelism inherent in processing sets of objects
  - Either with IBMs LIST definitions
  - Or ISV wildcarding
- Make sure you understand how to control the degrees of parallelism though

# Parallelism in recovery

- Parallelism in recovery is perhaps even more important
- You are definitely now concerned about minimising elapsed time
- Parallelism can be:
  - Multiple partitions
  - Multiple page sets
  - Table spaces and Indexes
  - Any combination of the above

# Partition level copy or not

- With partitioned objects you have another choice
- Copy the entire table space
  - Or index space
- Or copy partition by partition
  - For partitioned indexes too

# Partition level copy or not

- Copying at pageset level makes for an easy life
- Only one copy statement/job per object
- Also simplifies recovery of the entire object
- BUT....

# Partition level copy or not

- Copying at partition level increases flexibility
- ANY partition can be recovered independently
- Copies can also be run in parallel
  - Reducing the elapsed time of the copy
- Recoveries can also be run in parallel
  - Reducing the elapsed time of the recovery

# Partition level copy or not

- Partitioning the copies and recoveries is possibly the best choice
- For flexibility
- BUT at the cost of increased complexity
  - New partitions have to be added to the backup/recovery jobs

# Priorities

- Most IT staff consider a recovery sequence of:
  - Catalog + Directory
  - ISV utilities
  - User data
- To be sufficient for the needs of the business

# Priorities

- But this does not take into account the differing business requirements
- Some APPLICATIONS may be more critical than others
- Some applications may have critical OBJECTS
- Only the business knows these things
  - But IT knows how they interrelate
  - Business may require an object to be recovered as a high priority
    - But IT knows other objects are pre-requisites

# Priorities

- So it must be a consultation between the business and IT
  - Prioritise applications
  - Prioritise objects within applications
- And build these prioritisations into the recovery strategy

# Priorities

- These categorisations may also then drive revisions of the backup strategy
- Do these high priority objects require special treatment
  - More frequent full copies?
  - Change or Log accumulation processes?
  - Concurrent copies?
  - Mirroring?

# Quiesce

- A quiesce point is a point in the DB2 log where an object
  - Or more usually a set of objects
- Is not currently being updated by an in-flight unit of work
- In other words
- Is a point of CONSISTENCY to which objects may be recovered
- The quiesce utility is normally used to determine and register quiesce points

# Quiesce - Real or Virtual

- Virtual quiesce points can be discovered from the log
  - Even manually with DSN1LOGP!
- As well as be created with the QUIESCE utility

# DB2 Restart

- BUT if the prior shut down was NOT normal
  - i.e. there is no shut down check point
- DB2 knows that something is wrong
- Now the log must be scanned back to the most recent checkpoint
- This shows which UOWs were in flight
- Now each and every one of these must be undone before restart can be completed

# DB2 Restart

- As DB2 is processing these, compensation log records are also being created
  - In case DB2 fails during the restart
  - There has to be some way of picking up the pieces and starting again
- Once all of the in-flights have been recovered, the DB2 subsystem is now available

# Strategies for Recovery

- A recovery strategy is a situation where data loss or corruption has occurred
- And you have a plan to recover the data
- Even if they are not explicitly defined, most people have an idea of some of these
  - “What happens if I lose a volume”
  - “What happens if I delete all the data”
  - Etc.

# Strategies for Recovery

- But many recovery situations are not expected
- It is dangerous to ONLY have recovery plans to fix anticipated problems
- Even more dangerous to build BACKUP plans around expected failures
  - What happens if you have an unexpected failure
  - At worst, you will be unable to recover from it!

# Strategies for Recovery

- The safest recovery scenario to plan for is
- “We lost everything – Now what do we do”

- Any problem you have will then be a subset of this
- Hopefully, you will never actually lose everything

# Strategies for Recovery

- Although not part of this presentation, I would also suggest taking a BACKUP before starting any recovery
- It seems strange to want to back up something that you know is damaged
- BUT
- It is always possible to make things WORSE
  - Perhaps by making incorrect decisions
  - And you WILL be working under stress

# Strategies for Recovery

- Taking a pre-recovery backup does a number of things
  1. It gives you some thinking time to determine what your plan should be
  2. It ensures that if things do go very badly, you can at least get back to where you started
- Perhaps also take backups at various stages if this is a complex recovery
  - But do NOT overwrite one set of backups with another

# Strategies for Recovery

- Also, part of your recovery planning should be
- “Is all data equal?”
- Are there some objects that are MORE critical to recover than others?
- This is not an IT decision but a business one

# Strategies for Recovery

- So, the order of recovery becomes
  1. Recover catalog and directory
  2. Recovery ISV tools
  3. Recover high priority data
  4. Recover remaining data
- How many people do NOT differentiate between (3) and (4)?
  - Most I suspect

# Strategies for Recovery

- Those high priority objects may be candidates for special log handling
  - Change or Log Accumulation

# SYSCOPY housekeeping

- SYSCOPY can quickly grow to an unmanageable size
- The MODIFY RECOVERY utility is used to keep things under control
- MODIFY RECOVERY will remove entries from SYSCOPY based on specific criteria

# SYSCOPY housekeeping

- DELETE .... AGE() or DATE()
- Will delete rows that are older than the specified number of days or specific date will be deleted

# SYSCOPY housekeeping

- RETAIN .... LAST(), LOGLIMIT or GDGLIMIT
- Retains the
  - Last N entries in SYCOPY
  - All entries that are more recent than the oldest archive log timestamp
  - All entries that are more generations behind than the GDG holding the copied object will support

# System level point-in-time recovery


- DB2 Version 8 introduced BACKUP and RESTORE SYSTEM utilities
- To support backing up (and restoring) a full DB2 subsystem
  - Either data only or data and catalog/system data

## System level point-in-time recovery

- However, there is a string of pre-requisites
- Z/OS V1R5 or above
- Disk controllers supporting ESS FlashCopy API
- Specific DB2 named HSM constructs
- Defined SMS target copy groups
- DB2 must be running in at least V8 New Function Mode



There are many pre-requisites before these utilities can be used, as they rely on DFDSShsm services and hardware assists

Also, all of the DB2 pagesets must be defined to SMS and the SMS storage pools must be named in a specific way

Finally, DB2 must be running in Version 8 New Function Mode

# Testing

- Don’t forget that things change

- Your business needs may change
- Government/Industry demands may change
- DB2 (etc.) WILL change

- So your backup/recovery plans well need constant revision

# Questions/Thoughts?

# Phil Grainger

philg@philgrainger.demon.co.uk