

Unit 10

A few more Functions that Improve IMS Availability

What this unit is about

In the last few units, we have discussed IMS Online Change and DRD, major features designed to improve IMS availability. In this unit, we will briefly discuss several other IMS features that reduce the need for planned, and unplanned, outages.

This chapter talks about Enhanced Scheduling in IMS, I/O error behavior and some utilities for IMS DB backup.

IMS Enhanced Scheduling (1 of 2)

- A PSB used by a dependent region might reference hundreds of databases:
 - Many (or most) of these databases might be unavailable (or even non-existent)
 - In releases of IMS prior to 2.1, during scheduling IMS verified the usability of all databases referenced in the PSB (that is, no databases had a status of *STOPPED*)
 - If a database was unavailable, SCHEDULING WOULD FAIL
 - With current IMS Versions, the usability of Full Function database is not verified during scheduling:
 - IMS will defer checking for a Full Function database's usability until the first application DL/I call to the database
 - The usability of Fast Path databases is still verified as part of IMS scheduling of PSBs
 - The net result of this change is to greatly increase the likeliness of a successful PSB schedule

Discussing here :

- DB pre open ... status captured across (warm) restart
- Service calls (INIT STATUS ...)

IMS Enhanced Scheduling (2 of 2)

- An unmodified application that issues a DL/I call to an unavailable IMS database will receive:
 - *IMS TM Transactions will Pseudo ABEND U3303*
 - Here we have deferred our failure beyond schedule time
 - However, the IMS TM Input message that triggered this U3303 will be placed on the *Suspend Queue* and eventually be scheduled again
 - CICS and BMP PSBs will receive U0456 and be *STOPPED*
- If the application program had been modified to issue either:
 - *INIT STATUSGROUPA or INIT STATUSGROUPB*, the application will not ABEND
 - Instead the application will receive a Status Code informing it of the Unavailable database condition
 - The application can decide what action to take in response to this condition
- Application programs that do not issue the *INIT STATUS GROUPx* will be no worse off than before this change was made
 - Modified programs will be able to more fully achieve *Enhanced Scheduling*

The SCHEDULING in an IMSPLEX differs from single IMS environments !!
It's topic of the IMS performance workshops

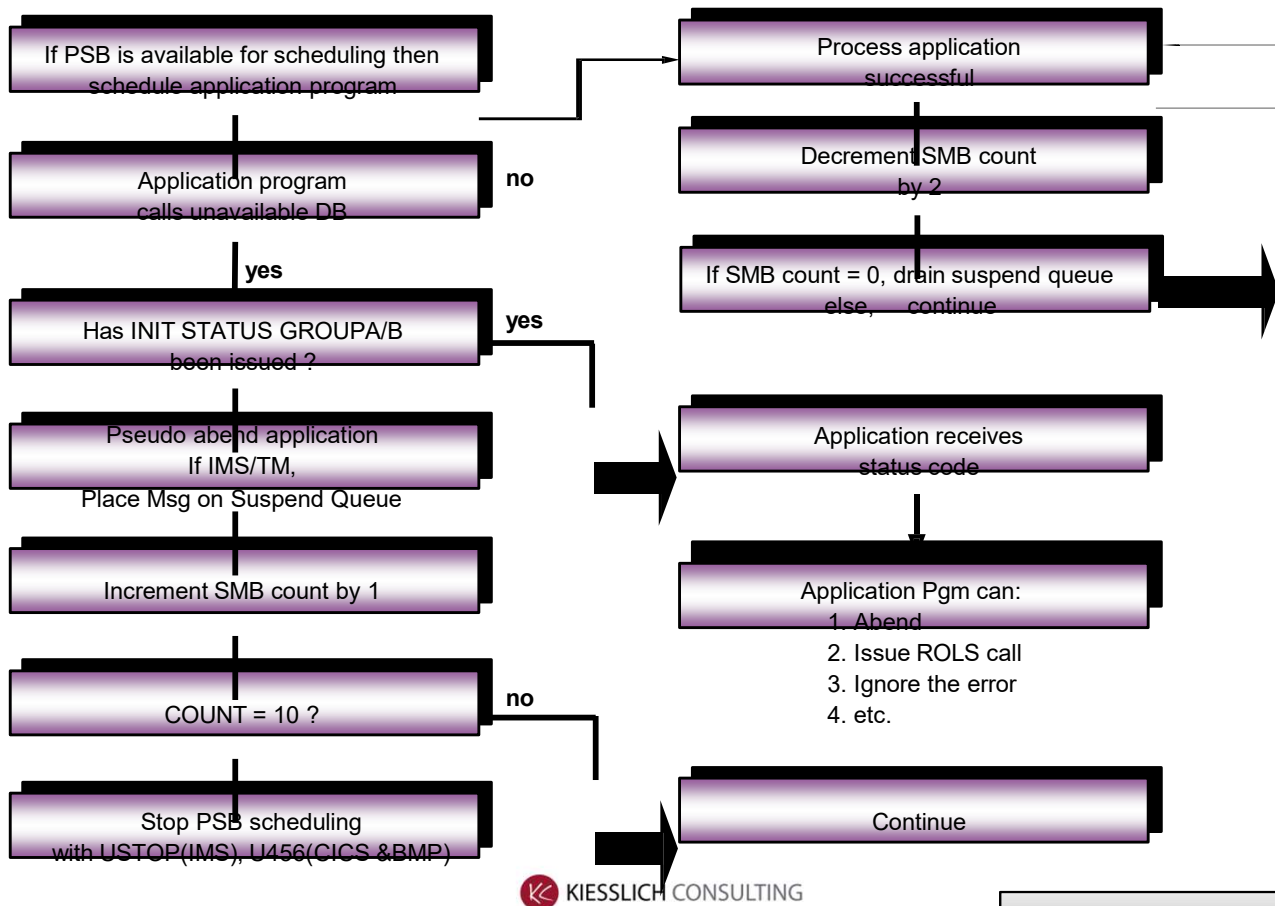
Pseudo ABEND processing and the Suspend Queue

- An unmodified program issuing a DL/I call to an unavailable database receives pseudo ABEND U3303
 - Additionally, for these unsuccessful executions, a special counter in the transaction's SMB is incremented by 1
 - If this counter reaches a value of 10, the PSB is *USTOPped*
 - IMS will not attempt further scheduling of USTOPped transactions
 - After each successful execution, counter in SMB **decremented by 2** until counter = 0
 - That can be differing processing performed by individual transaction instances
 - IMS *rewards* successful transaction scheduling by making scheduling more likely to continue
- In response to a /STA DB command to transactions referencing the STArted database:
 - Counter in SMB is reset to 0
 - Transaction is no longer USTOPped
 - *Suspend queue* is drained (transactions reprocessed) in IMS TM
 - This reprocessing will almost certainly **not** be in FIFO sequence

Notes:

With U3303 ABENDs and the Suspend Queue, we have a situation in which bad news (scheduling failure) has less of an impact than good news (scheduling success).

Enhanced Scheduling flow



Notes:

With Enhanced Scheduling, IMS provides an application that experiences a read error the opportunity to try again later without systems programmer, DBA, or operator involvement.

DLI I/O error processing (1 of 2)

- The basic approach of IMS is to postpone the need to recover from database I/O errors as long as possible:
 - Database in error is not stopped
 - Defer database recovery to more convenient time
- When write errors occur, the blocks or CIs are saved in IMS storage and across restarts and reloaded for continued use
- IMS will retry failed I/Os at a later point in time
- At the time of the I/O error, IMS will provide more detailed information on I/O error status
 - There will always be an:
 - DFS0451 Message issued for first occurrence of each I/O error per database block/CI
 - For VSAM message:
 - DFS0451I DFSDVSM00 dbdname ddname i/o_error_type will be issued.



Discussing here :

- EEQE a.o. remain .. Even across restart (logged) ... see restart handling a few foils later

DLI I/O error processing (2 of 2)

- For OSAM, these messages will be issued:
 - DFS0451I DFSDBHM00 dbdname ddname i/o_error_type
AND
OSAM: DFS0762I OSAM write ERROR - FUNC=08 ..
or
OSAM: DFS0762I OSAM read ERROR - FUNC=01

Different response to Read and Write I/O errors

- Write errors from an IMS *Online System*:
 - No status code
 - Error is logged (EEQE) Virtual Buffer is allocated and logged and block/CI is moved into it. It resides in ECSA.
 - This does not happen with Batch (DLI BATCH or DB Batch)
 - Batch job will fail
 - Notify DBRC if DB is registered
 - Later, if this block is requested (read) by THIS IMS, the block/CI is copied into database buffer from ECSA Virtual Buffer
- Read errors:
 - For first reference from this system:
 - 'AO' status code
 - Error is logged (EEQE)
 - Notify DBRC if DB is registered
 - For subsequent read from THIS IMS, data is returned from Virtual buffer copy as described above
 - See next page for data sharing considerations

Notes:

The enhanced protection from write I/O errors requires some kind of an IMS Control Region (DB/DC or DBCTL); no Virtual Buffers with Batch.

Surely it fails in batch : should run fast thru and no „high avail“ for more than the batch PSB expected!

Extended Error Queue Elements (EEQEs), both read and write, are created and logged for the first instances of an I/O error to a block.

I/O errors and DBRC

- If database is **registered** with DBRC:
 - Recovery needed flag = on (in DBDS-Record)
 - Recovery needed counter incremented by 1 (in DB-Record)
 - EEQE (Extended error queue element) with block number/RBA of block/CI in RECON (DBDS-Record)
 - If database is not registered to DBRC, any batch job would not be aware of I/O errors in the database
- Access by another Data Sharing subsystem:
 - I/O errors experienced accessing a database by one IMS subsystem do not globally stop access to the database by other subsystems
 - Most database blocks/CIs are available for read and write from other sharing IMS Subsystems
 - Exception: **write-error** block/CI
 - For sharing subsystems, reads to the write-error block/CI is treated as a read error:
 - DBRC is checked for EEQEs before read is done
 - No I/O operation is even attempted for the write-error (EEQE) block
 - 'AO' status code returned to application

Notes:

In a Data Sharing environment, a Sharing IMS will first check with DBRC for Extended Error Queue Elements (EEQEs) before reading a block; if there is an EEQE for a requested shared block, the AO status code will be returned without even attempting to perform the read since the data on disk is an old version of the data. The current version of the changed error block is only available from the Virtual Buffer in the system that first experienced the write I/O error.

IMS I/O error logging

- Log record X'25' EEQE:
 - Block/CI address in error
 - RBA, DBD and DCB
- Log record X'26' I/O toleration Virtual Buffer (if write error):
 - Block number / RBA of error block / CI
 - Virtual buffer record, corresponds to block/CI
 - DBD,
 - DCB, and
 - Data of block/CI
- Close time, if all retries read and write errors successful:
 - Message DFS615I: *All I/O errors successfully corrected for database 'dbdname' written*
 - No recovery needed
 - DBRC recovery needed flag turned off
 - DBRC recovery count zeroed
 - EEQE deleted

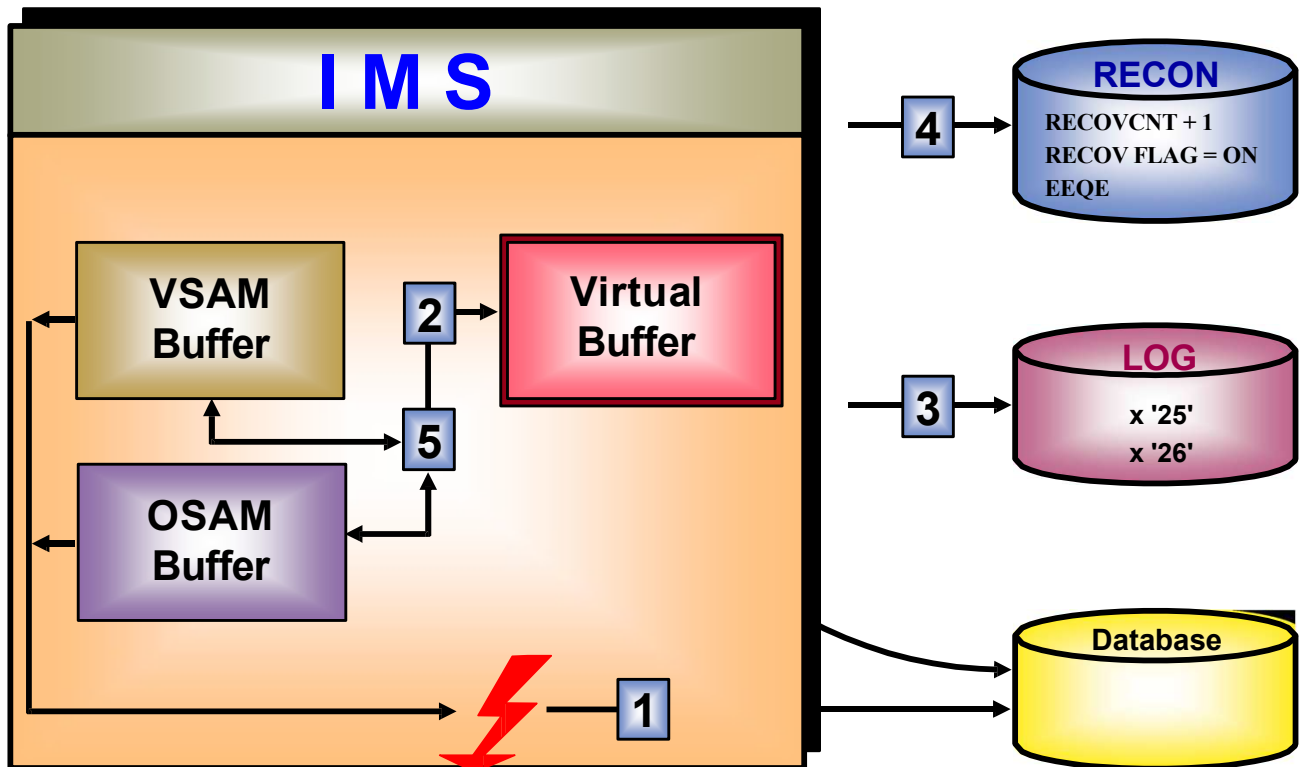
Notes:

Log records corresponding to I/O errors are written to the IMS log as the errors occur. For write errors from a control region, the creation of virtual buffers is also logged. IMS will retry I/O at database close, which will occur at (least at) IMS shutdown time.

Write errors and Restart processing

- Virtual buffer available across IMS restarts from checkpoint records
- At Warm start or ERE of IMS:
 - IMS looks for X'26' I/O toleration buffer
 - If database registered
 - Compares block number/RBA against EEQE in DBDS record of RECON:
 - Match with RECON Create virtual buffer
 - No match with RECON:
 - > If EEQE not in RECON, assume recovery run and no virtual buffer creation
 - > If EEQE in RECON, but no match in X'26,' EEQE belongs to another subsystem
 - > Create EEQE and Reads will receive an 'AO' status code for that block/CI
 - Database not registered:
 - Virtual Buffer Recreated from Log
 - EEQE considered valid until /DBR command for the database with one or more EEQEs
 - /DBR Command DELETES all the EEQEs from the log and the Virtual buffers
 - > This happens for registered databases also
 - The /DBR command has no impact on the EEQEs in the RECON for registered databases

Write error processing



Notes:

Database write error

Move block/CI to virtual buffer for future reference

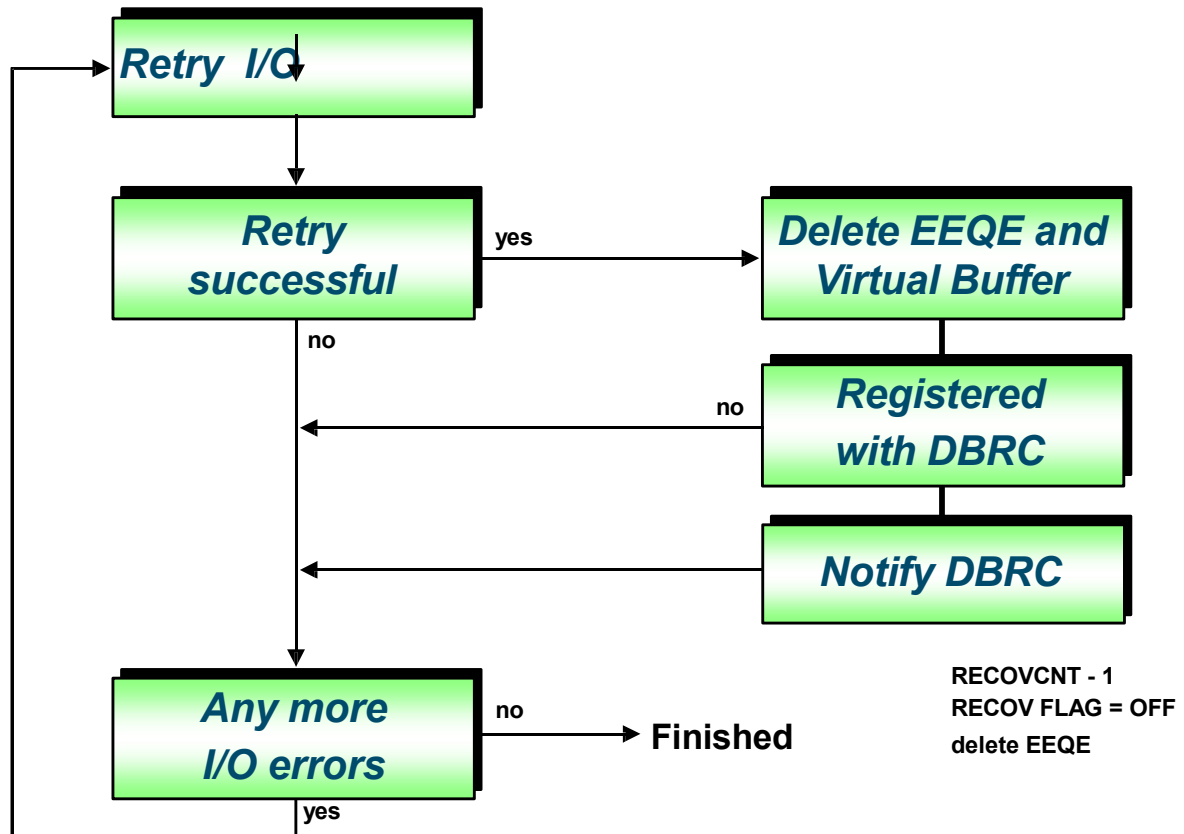
Log virtual buffer when created or changed and at checkpoint

Update RECON

Records in error will be read from virtual buffer

Retry write at database close

Database close processing



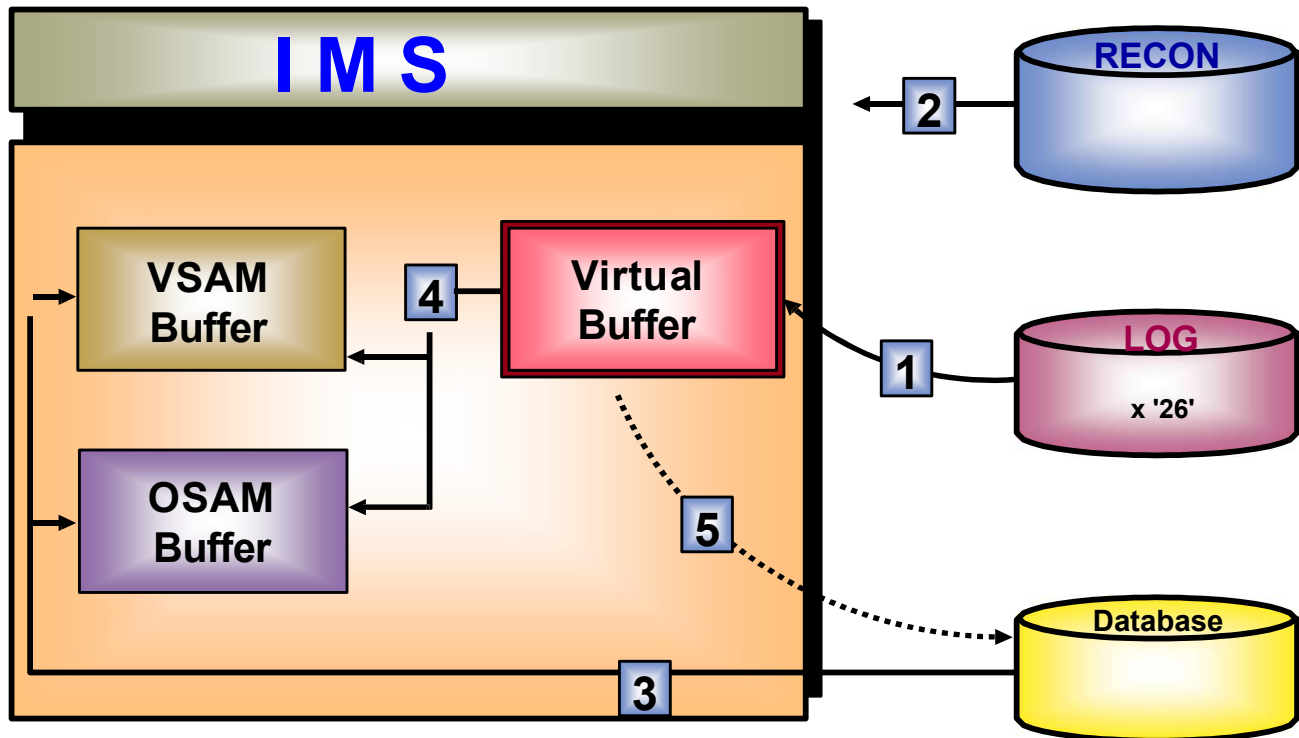
Notes:

At Database close, the writes are retried. If successful, the Virtual Buffers are released (by writing a log record indicating that the write was successful) and DBRC is informed of the fact and removes the indicator of the error from the RECONS.

Write errors and Restart processing ...

- Checkpoint time:
 - x'4025' EEQE log record in checkpoint
 - x'4026 I/O toleration virtual buffer in checkpoint
- Cold start of IMS:
 - No virtual buffer creation during Cold start
 - DB needs recovery prior to Cold start for valid access to error blocks:
 - Otherwise, read attempts will be made to old data on disk
 - If database is registered, RECON EEQEs and other flags and counters will still be set after Cold start:
 - DBRC will prevent invalid read attempts
 - Access to error blocks will be prevented until after recovery

Warm or Emergency Start of IMS and EEQEs



Notes:

- Read buffer from the log checkpoint record
- Read RECON EEQE (*) Resume processing
- Read good data from database
- Read error data from virtual buffer
- Retry write at database close

IMS Online Image Copy (1 of 2)

- This is a *fuzzy* backup
 - All database blocks are not on the online image copy data set in a consistent state
 - RECOVERY will use OLIC data set **and** log **and/or** CA
- The Online Image Copy is a batch message processing program (BMP) and runs under the control of IMS
 - Unlike *normal* BMPs, the OLIC BMP requests special DBRC authorization
- The subsystem under whose control the OLIC runs can concurrently update the database:
 - **Only** the subsystem on which the OLIC job runs can be authorized with access intent = UP for the databases being copied
 - All other Subsystems can only be authorized with access intent = RD/RO even if Block Level Sharing is implemented
- The OLIC BMP requires a PSB with the OLIC=YES parameter and PROCOPT=GO. The PSB must be defined in the IMS SYSGEN.

Notes:

Online Image Copy has been available for a long time and does not require the use of DBRC. However, its use can impact the performance of your IMS system since it uses buffers that could otherwise be used for productive processing; it is also not suitable for databases that are involved in IMS Data Sharing.

IMS Online Image Copy (2 of 2)

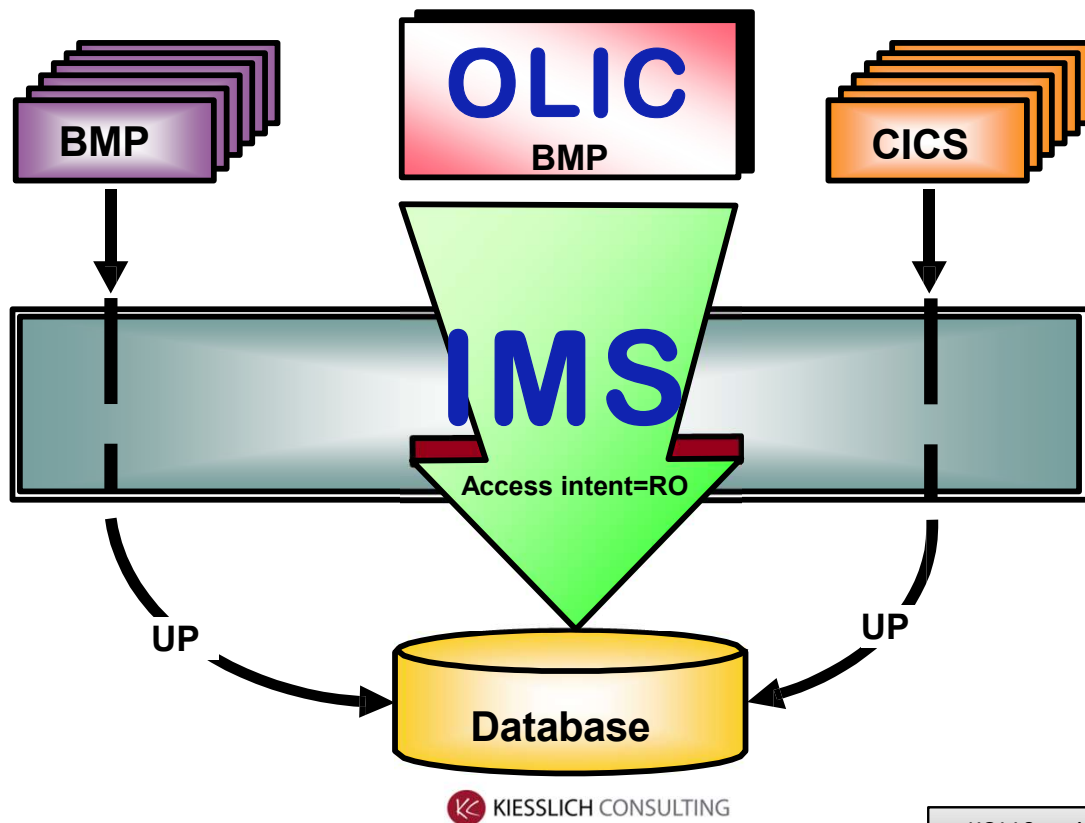
- When making an online image copy of a non-recoverable database, the database access intent must be set to RD or RO (/DBD command)
 - Like batch image copy
- PRILOG compression can take place during *archiving* or 'DELETE.LOG INACTIVE' dependent on LOGRET(...) parameter

But:

Usually a non recov index isn't that important and will be rebuilt anyway

Discussing more about PRILOG Compression (since V8) when and why

IMS Online Image Copy processing



Notes:

Only the IMS Subsystem on which OLIC runs can update the database while OLIC runs. This makes it impractical for use in a data sharing environment.

IMS Concurrent Image Copy

- Output is a *fuzzy* image copy like online image copy
- Same program as for batch image copy (DFSUDMP0) but with an additional parameter:

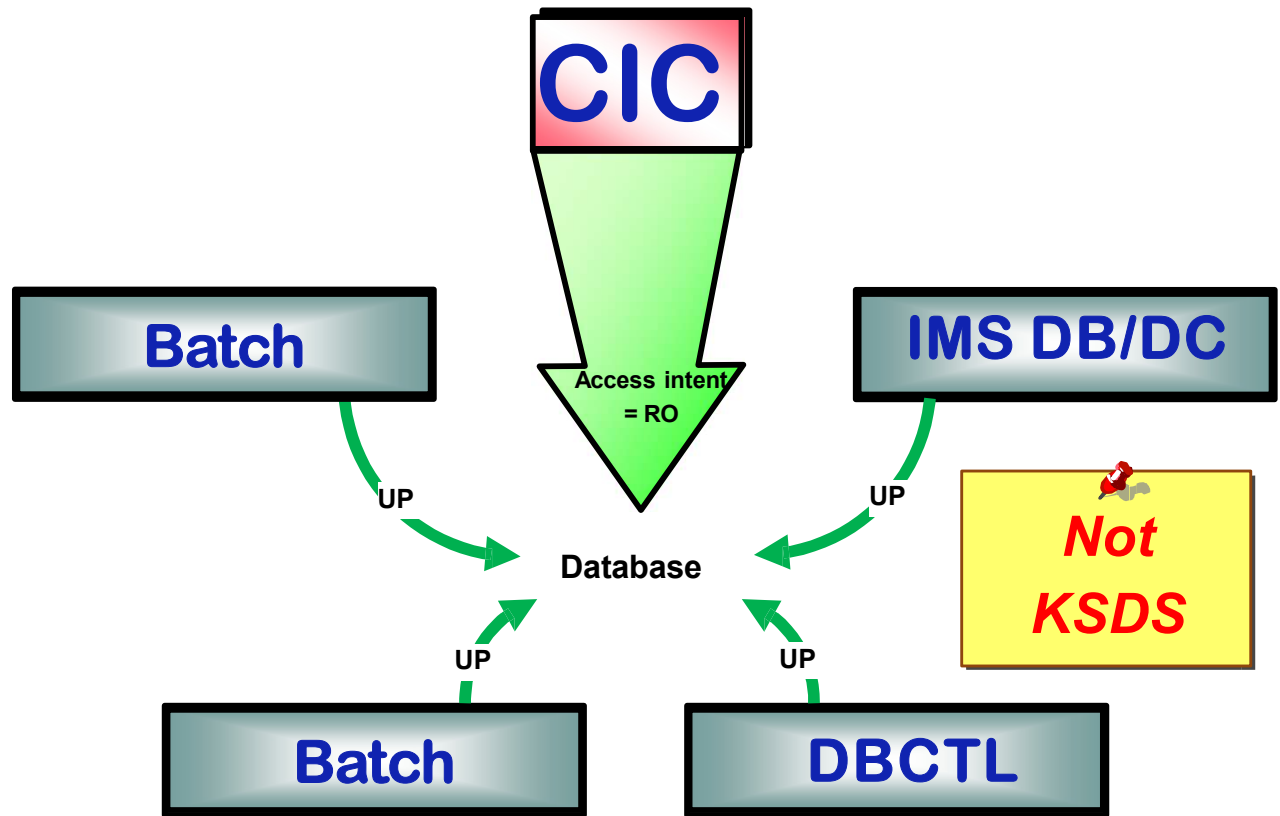
```
//IC1      EXEC PGM=DFSUDMP0,REGION=800K,PARM='DBRC=Y,CIC'
```

- Runs parallel to other updating IMS subsystems
- CIC requires that:
 - DB is registered in RECON
 - Minimum DBRC share level is 1
 - Parallel subsystems run under DBRC
- CIC does not support:
 - VSAM KSDS
 - Non-recoverable databases
- GENJCL.RECOV supports CIC by including the correct logs as input to the recovery job

Notes:

The restriction that requires the use of DBRC should not be a problem for most customers. The restriction that prevents the use of CIC for KSDS data sets can be addressed by using other tools (for example, IBM's Index Builder) to recreate, instead of recover, index databases.

IMS Concurrent Image Copy processing



IMS ImageCopy 2

- A concurrent backup which supports OSAM, KSDS, ESDS
- DFSUDMT0 - Database Image Copy 2 Utility
 - Invokes DFSMSdss DUMP to create a copy of the data set
 - Can produce a clean (like batch IC) or fuzzy (like OLIC/CONCUR IC) image copy
 - Registers two copies with DBRC
- Two phases:
 - Logical Copy phase: Storage subsystem to maintain track status
 - Physical Copy phase: When logical copy completes, copies target data set to image copy data set
 - Each phase issues distinct WTO messages that can be responded to
- DBRC support (required):
 - SMSNOCIC (Clean image copy)
 - SMSCIC (Fuzzy image copy)
- Database Recovery (DFSURDB0):
 - Invokes DFSMSdss RESTORE for data set
 - Recovers using restored data set with logs and/or change accumulation

Notes:

Image Copy 2 works with DFSMS, and Disk Subsystem hardware, to produce either crisp or fuzzy image copies; it can be used for any IMS database type, but requires that the database be registered to DBRC.

The only utility which can work against groups

The utility which interferes to DFSMSdss (ADRDSSU) based backups ... and furthermore usable to kick off flashcopy / snapshotcopy technologies

Especially embedded into a QUIESCE

IMS Image Copy 2 processing

