



Modernizing your DB2 environment with DB2 Utilities and Tools for z/OS

Hendrik (Hennie) Mynhardt – DB2 for z/OS Exec ITS

March 26, 2013





Disclaimer

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.



Agenda



- ▶ **DB2 Real Time Statistics (RTS)**
- ▶ **DB2 DSNZPARMS affecting DB2 Utilities**
- ▶ **Making copies of DB2 data**
- ▶ **Backing up and recovering DB2 data**
- ▶ **Managing your utilities**



DB2 Real Time Statistics



Objective of Real-time Statistics

- Current DBA and some home-grown monitor tools had no accurate data to identify objects that need maintenance
- IBM DB2 Automation Tool and DSNACCOX (Stored Procedure) exploit RTS – both recommended by IBM for object maintenance
- Performing maintenance on static and unchanged objects makes
 - inefficient use of DBA's time
 - waste batch window time
 - and waste CPU – the fastest utility is not to run the utility
- DB2 system becoming large and complex
 - A single DB2 for z/OS may have large amounts of tables/indexes - for ERP-packaged applications, it can be 60K +
 - Requires skilled DBAs to identify unused objects
- Goal is to self-managed or automate the maintenance process



History of Real-time Statistics (RTS)

- Introduced in DB2 V7
- Contain space and as well as some access path statistics
- Part of DB2 Catalog – DB - SYSIBM.DSNRTSTS:
 - TB - SYSIBM.SYSTABLESPACESTATS
 - TB - SYSIBM.SYSINDEXSPACESTATS
- Intended to eliminate running utilities to collect statistics – run utilities by exception
- Not replacing RUNSTATS
- Runs in the background. Automatically updates statistics, as the data in DB2 databases is modified.



RTS Tables – SYSTABLESPACESTATS

Global Statistics	Incremental Statistics		
	REORG	COPY	RUNSTATS
NACTIVE NPAGES EXTENTS SPACE TOTALROWS DATASIZE UNCOMPRESSED DATASIZE UPDATESTATSTIME HASHLASTUSED DRIVETYPE LPFACILITY	LASTTIME INSERTS UPDATES DELETES DISORGL0B UNCLUSTINS MASSDELETE NEARINDREF FARINDREF REORGSCANACCESS REORGHASHACCESS REORGCLUSTERSENS	LASTTIME UPDATEDPAGES CHANGES UPDATELRSN UPDATETIME	LASTTIME INSERTS UPDATES DELETES MASSDELETE

DB2 10



RTS Tables – SYSINDEXSPACESTATS

Global Statistics	Incremental Statistics		
	REORG	COPY	RUNSTATS
NACTIVE NLEVELS NPAGES NLEAF EXTENTS SPACE TOTALENTRIES LASTUSED UPDATESTATSTIME DRIVETYPE	REBUILDLASTTIME REORGLASTTIME REORINSERTS REORUPDATES REORGDELETES APPENDINSERT PSEUDODELETES MASSDELETE LEAFNEAR LEAFFAR NUMLEVELS REORGINDEXACCESS	LASTTIME UPDATEDPAGES CHANGES UPDATELRSN UPDATETIME	LASTTIME INSERTS DELETES MASSDELETE



DB2 10

RTS – In memory collection

- **Allocate RTS blocks – pages in memory**
 - Table Spaces - at first **update** - since the page set/partition is opened
 - Index Spaces - at **open time** - DB2 updates SYSINDEXSPACESTATS.LASTUSED
- **In DBM1 Address Space**
 - ~140 bytes per page set/partition – above bar in V9+
 - Maximum 346 bytes per page set/partition – above the bar in V9+
- **Free RTS blocks when**
 - Page sets/partitions are closed
 - After statistics are written to RTS tables
- **In a data sharing group, statistics are collected by each member**
- **In-memory statistics are always collected even if RTS is not enabled**
 - Order active statistics blocks in clustering order
 - Insert/update rows in the RTS tables via the clustering index

RTS – Eternalization of in-memory blocks

- **On timer intervals**
 - ZPARM – **STATSINT** - default is 30 minutes
 - REAL TIME STATS in DSNTIPO install panel
 - Range: 1 to 1,440 minutes
- **RTS manager runs under a system task in DBM1 address space**
 - CPU time is included in DBM1's SRB time
 - The system task is created during START DB2
- **STOP/START DATABASE SPACENAM command**
 - Flush in-memory statistics for all target objects
- **STOP DB2 MODE(QUIESCE) & STOP DB2 MODE(FORCE)**
- **DB2 utilities trigger externalization of in-memory statistics**
 - Each data sharing member externalizes its own statistics
 - Via synchronous NOTIFY messages





RTS – what may not be captured?

- DB2 Directory (DSNDB01), SYSLGRNX and SYSUTILX – for some operations like COPY
- LISTDEFS containing RTS objects – data may not be externalized
- Only some counters for work files / DSNDB04
 - EXTENTS, SPACE and NACTIVE
- RECOVER utility DOES NOT populate RTS
- DROP will remove rows from RTS tables



RTS History – use for trending

- There is currently no historical capability in RTS
 - Create a history table manually

CREATE

SYSIBM.(TABLE | INDEX)SPACESTATS_HIST

LIKE

SYSIBM.SYS(TABLE / INDEX)SPACESTATS

- then *add*

CAPTURE_TIME AS TIMESTAMP NOT NULL WITH DEFAULT column

- Periodically insert into RTS history tables with a sub select from the RTS tables those rows that aren't already in the history tables; and delete old information.
 - Some customers do this weekly, others monthly – depending on needs

Some uses for RTS

Object activity

How active are my objects?

- What activity has taken place for a specific time for TS' and IX's
- Use UPDATESTATSTIME

```
SELECT DBNAME, NAME, PARTITION, UPDATESTATSTIME
FROM SYSIBM.TABLESPACESTATS
WHERE (JULIAN_DAY(CURRENT DATE) –
      JULIAN_DAY(UPDATESTATSTIME)) <= 14
      AND NAME = xxx;
```

- Use DB2 Administration Tool which contains performance queries to run against RTS



Some uses for RTS

Utilities – when was it last run against my objects

- When was COPY, REORG, LOAD REPLACE, and RUNSTATS last executed against my objects ..

```
SELECT DBNAME, NAME, PARTITION, TOTALROWS, NACTIVE, SPACE,  
EXTENTS, UPDATESTATSTIME, STATSLASTTIME,  
LOADRLASTTIME, REORGLASTTIME, COPYLASTTIME  
FROM SYSIBM.TABLESPACESTATS  
ORDER BY DBNAME, NAME, PARTITION
```

- Use DB2 Administration Tool for reporting
- For object maintenance, use DB2 Automation Tool or the free stored procedure DSNACCOX



Some uses for RTS

BACKUPS

- For copies based upon changes, consider INCREMENTAL / FULL
- For sizing – eg. Use DB2 FCIC based upon TEMPLATE switching
- Use RTS history tables for RTO planning :
 - Use SPACE or CHANGES to group objects for backup purposes
- Use RTS / history UPDATETIME to determine when backups are run – take back control
 - Determine **when** and **how many times** backups run
 - Remove duplicates – where needed

COPYLASTTIME

last IC timestamp

COPYUPDATEPAGES

of pages updated since last IC

COPYPAGES

of rows changed since last IC



Some uses for RTS

REORGS

- For scheduled REORGS – why bother with RTS?
- Consider using RTS for conditional REORGS
 - EXTENTS
 - REORGUNCLUSTINS - # of rows inserted out of sequence
 - REORGINSERTS, -UPDATES and -DELETES - # of rows ~ since last
- Best REORG is no REORG – no benefit, why REORG



Some RTS stats worth a serious look...

Unused or (in)activity of INDEXES

- LASTUSED column in *SYSINDEXSPACESTATS*
 - Is a **date** field
 - Consider using for identifying which IXs to drop
 - The date indicates the index is last used for SELECT, FETCH, searched UPDATE, searched DELETE, or used to enforce referential integrity constraints.
 - The default value is 01/01/0001.
- REORGINDEXACCESS column in *SYSINDEXSPACESTATS*
 - # of times the IX was accessed with DML since last reorg or since creation
 - NULL denotes never used



Usage of RTS – some examples

- History of space allocation
 - When trend analysis is needed – for capacity planning and also:
 - Backup strategy based upon size
 - Reorg strategy – Partitioning decisions
 - Purging / archiving old data – contain growth

- What about using RTS stats info your own table
 - Capture the thresholds violating the conditions used to “trigger” the reorganizations.
 - Keep this in a DB2 table
 - Object name, partition, timestamp, condition, threshold and actual value
 - which caused the reorg to happen

- Use the DB2 Automation Tool

RTS - Determining which objects are missing stats

- * objects that don't exist in **systablespacestats**
- * -> Run REORG TABLESPACE

```

SELECT P.DBNAME, P.TSNAME, P.PARTITION
FROM SYSIBM.SYSTABLESPACE S
INNER JOIN SYSIBM.SYSTABLEPART P
ON      P.DBNAME = S.DBNAME
AND     P.TSNAME = S.NAME
WHERE NOT EXISTS
        (SELECT 1
         FROM SYSIBM.SYSTABLESPACESTATS T
         WHERE S.DBID=T.DBID
              AND S.PSID=T.PSID
              AND P.PARTITION=T.PARTITION)
ORDER BY P.DBNAME, P.TSNAME, P.PARTITION
FOR FETCH ONLY WITH UR;

```

- * objects that don't exist in **sysindexspacestats**
- * -> RUN REORG INDEX

```

SELECT P.IXCREATOR,
       P.IXNAME,
       P.PARTITION
FROM SYSIBM.SYSINDEXES X
INNER JOIN SYSIBM.SYSINDEXPART P
ON P.IXCREATOR = X.CREATOR
AND P.IXNAME = X.NAME
WHERE NOT EXISTS
        (SELECT 1
         FROM SYSIBM.SYSINDEXSPACESTATS T
         WHERE X.DBID=T.DBID
              AND X.ISOBID=T.ISOBID
              AND P.PARTITION=T.PARTITION)
ORDER BY P.IXCREATOR, P.IXNAME, P.PARTITION
FOR FETCH ONLY WITH UR;

```



DSNZPARMS affecting DB2 Utilities



DSNZPARMS and Utilities

ZPARMS affecting SORT operations:

- **UTSORTAL** – (Yes / No) Enable dynamic allocation of SORTWK datasets. Uses DB2 RTS to estimate the number of rows to sort.
- **IGNSORTN** – Overrides (Yes / No) if SORTWK DDs are allocated in JCL

ZPARMS affecting temporary dataset allocation:

- **VOLTDEFT** - Specifies the device type or unit name that is to be used by DB2 utilities for dynamically allocating temporary data sets. Used for COPY CONCURRENT (DSS) datasets
- **UTIL_TEMP_STORCLAS** – Specifies the SMS storage class that the CHECK INDEX, CHECK DATA, and CHECK LOB utilities are to use when allocating temporary shadow data sets. (These utilities allocate shadow data sets when the SHRLEVEL CHANGE option is used).

DSNZPARMS and Utilities

ZPARMS affecting COPY Utility:

- **FLASHCOPY_COPY** – (Yes / No) Specifies that the COPY utility uses FlashCopy technology when the FLASHCOPY option / keyword is not specified in the control statement.
- **FLASHCOPY_LOAD** – (Yes / No) Specifies whether LOAD utility uses FlashCopy technology when the FLASHCOPY option is not specified in the control statement – inline copy
- **FLASHCOPY_REORG_TS & FLASHCOPY_REORG_IX** – (Yes / No) – Inline copy at load phase for REORG – when not coded on LOAD control statement
- **FLASHCOPY_REBUILD_IX** – (Yes / No) – Can Rebuild IX use FlashCopy by default when not specified on the REBUILD INDEX utility control statement

DSNZPARMS and Utilities

ZPARMS affecting **BACKUP / RESTORE SYSTEM Utilities**

- **SYSTEM-LEVEL_BACKUPS** – (Y/N) - Enables the RECOVER utility to use SLB as input for object-level recoveries.
- **RESTORE_RECOVER_FROMDUMP** – (Y/N) - Specifies if RESTORE SYSTEM and RECOVER can use SLB dump on tape as input for recovery. Yes – use tape. No – use only disk
- **MAXIMUM TAPE UNITS** – **RESTORE_TAPEUNIT** (1-255). The number of tapedrives RESTORE SYSTEM allocates for restore from dump.



DSNZPARMS and Utilities

ZPARMS affecting the **RECOVER** utility:

- **REC_FASTREPLICATION** – (Required, Preferred, None) - Specifies whether recovery from a FlashCopy image copy should use FlashCopy
 - **REQUIRED** - The **RECOVER** utility forces use of FlashCopy when performing recovery from a FlashCopy image copy, to ensure that recovery occurs as quickly as possible. However, this option will cause RECOVERY to fail if Flash Copy cannot be used
 - **PREFERRED** - The **RECOVER** utility uses FlashCopy only if FlashCopy support is available.
 - **NONE** - The **RECOVER** utility will use standard input/output to restore the FlashCopy image copy. This setting is not permitted when the FLASHCOPY_PPRC parameter is set to PREFERRED or REQUIRED.
- **LOGAPSTG** (No longer used in DB2 10) – specify 100MB (old ROT)
 - Specify 10 MB of log apply storage for each concurrent RECOVER job that you want to have faster log apply processing. The default value of 100 MB provides log apply storage for 10 concurrent RECOVER jobs



DSNZPARMS and Utilities

ZPARMS affecting **RECOVER** utility:

- **REC_FASTREPLICATION** – Should the RECOVER utility use fast replication to recovery from DB2 10 FCIC
 - None – Uses standard I/O to restore from FCIC
 - Preferred – uses FC to recover from FCIC if avail
 - Required – Forces RECOVER to use FC – will fail if FC is not avail

ZPARAM for **DB2 Sort** utility:

- **DB2SORT** – (Enable / Disable) - specifies whether DB2 utilities are to use DB2 Sort instead of DFSORT for utility sort processing when DB2 Sort is installed

DSNZPARMS and Utilities

ZPARMS affecting REORG utility:

- **REORG_LIST_PROCESSING** – (Parallel / Serial) – This parm controls the PARALLEL option for REORG. Used during LISTDEF processing – ‘parallel yes or no’
- **REORG_IGNORE_FREESPACE** – (Y/N) - Controls whether DB2 uses the PCTFREE and FREEPAGE values for PBG
 - **YES** - DB2 will set 0 for the PCTFREE and FREEPAGE
 - **NO** – DB2 will use the PCTFREE and FREEPAGE values
- **REORG_PART_SORT_NPSI** – (Y/N) – For REORG TABLESPACE PART – To sort all of the keys of a non-partitioned secondary index (y/n). Ignored for non-PART reorgs
 - Default of N means only keys for NPSI are sorted



DSNZPARMS and Utilities

ZPARMS affecting multiple utilities:

- **FCCOPYDDN** - Used for the FLASHCOPY option for COPY, LOAD, REBUILD IX, REORG IDEX, REORG TABLESPACE. Specified the template for the FCIC name
 - Default - HLQ.&DB..&SN..N&DSNUM..&UQ
 - FCIC uses VSAM copies – no GDG's etc



Making copies of data



Copying data – many options

- **DSNTIAUL**
- **Unload Utility**
- **High Performance Utility (HPU)**
- **Copy Utility**
- **DB2 Cloning Tool**



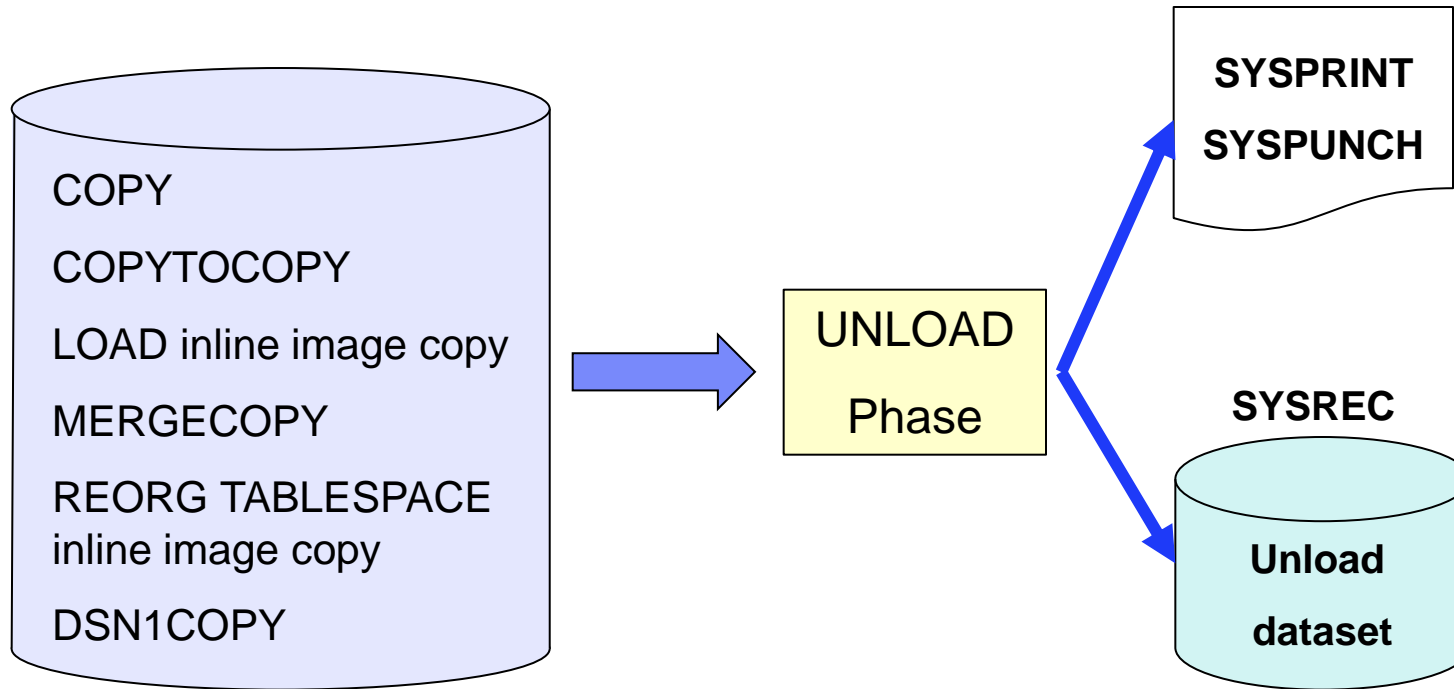
DSNTIAUL - DB2 sample program

▪ DSNTIAUL

- The DB2 sample unload program - written in assembler and is a simple alternative to the UNLOAD utility
- Handle SQL statements up to 2 MB in size
- Uses multi-row FETCH
- To enable MULTI-ROW FETCH:
 - By default DSNTIAUL retrieves 100 rows per fetch
 - Use the value of 32767 for better performance

```
RUN PROGRAM(DSNTIAUL) PLAN(DSNTIAUL)  
PARMS('SQL,32767') LIB('DSN.RUNLIB.LOAD')
```

DB2 Unload utility – COPY inputs



DB2 10 FCIC input is not supported:

- Use COPYTOCOPY to make a sequential copy and use that as input.
- Or use HPU which does support DB2 10 FCIC



DB2 UNLOAD utility

Performance

- Invoke parallelism wherever possible:
 - Unload PARTs to different outputs
 - Optimally the # of subtasks will equal the # of parts unload
 - UNLOAD limits the degree of parallelism to one times the number of processors on the LPAR
 - DB2 no longer counts subtasks towards IDBACK and CTHREAD
 - but check IDBACK to prevent failures
- Use **UNLOAD FORMAT INTERNAL**
 - Unload and load data in true internal format
 - Avoids field processing
 - Significant savings – both CPU and ET
 - Customer: “UNLOAD of 654m rows went from 5 hrs to 50 mins. Both CPU and ET reduced by 82-85%”



DB2 Unload utilities

DB2 HPU

- **Accesses the DB2 linear VSAM data sets directly**
 - Huge performance benefits – No DB2 BP usage
 - ORDER BY, GROUP BY functions do not use DSNDB07
 - Uses DFSORT or DB2 Sort (if installed)
 - Uses VSAM buffering capabilities – allows an entire cylinder to be read in a single I/O
 - HPU allows you to run several unload jobs that access the same table space in parallel
 - CPU costs significant lower then DB2 Unload Utility – ET may be similar or slightly less
 - Has SQL 'interface' – for complex SELECTS (JOINS etc)

DB2 Unload utilities

DB2 HPU

- **Use INDEXSCAN – unload the data in clustering seq.**
 - If IX is very disorganized this option is disabled
 - Useful (AUTO=Y) when use with LOAD PRESORTED
 - Use with ORDER CLUSTER or an ORDER BY – not with WHERE clause

- **HPU supports internal format**
 - INTERNAL_FORMAT



DB2 LOAD utility

- **Keep maintenance current**
- **LOAD INTERNAL FORMAT**
 - 77% elapsed time, 56% CPU reduction on LOAD
 - DB2 UNLOAD and HPU support
- **LOAD PRESORTED**
 - Avoids sort overhead when data already sorted in clustering order
 - Up to 25% CPU reduction, 33% ET reduction depending on # of indexes
- **Utility Enhancement Tool PRESORT option**
 - Use PRESORT option in Utility Enhancement Tool
- **For LOAD REPLACE, consider loading into a “clone” then renaming tables or datasets**



DB2 LOAD utility

- **Allocate inline copy data sets to DASD or use FCIC (DB2 10)**
- **Split up input dataset and drive LOAD partition parallelism in a single LOAD**
 - Use parallelism functions in the unload utilities
- **Use SORTNUM elimination**
 - **UTSORTAL** has been around for a long time and RTS has been around since DB2 V7
 - OLD challenges were fixed:
 - One size does not fit all - many sorts in the same utility invocation have very different needs for sort work data sets
 - Utilities allow only a single SORTNUM specification which tells DFSORT into how many data sets it should split the sort work space
 - Large sorts require many data sets – same with smaller sorts – lots of waste and not optimized



DB2 LOAD utility

▪ **LOAD INDEXDEFER**

- Lets LOAD to bypass the sort and build of the indexes during the LOAD phase - puts the deferred indexes in Rebuild Pending (RBDP) status
- Do REBUILD IX afterwards
- Potential savings in CPU time

▪ **More parallel options**

- Use LOAD REPLACE with LOG NO and COPYDDN
 - Reduces log writes and TS is avail after load – no COPY PENDING
- For partitioned TS, always use one dataset per partition – UNLOAD and LOAD



Cloning Terminology

- **A clone is an exact but independent replica**
 - Clone a DB2 system by **volume**
 - Clone a table space by **data set**
- **DB2 system cloning and table space refresh**
 - The act of replicating the data, making the replica accessible, and then using the replica in lieu of the original data
- **DB2 system cloning automation**
 - Clones a complete DB2 system including all its databases
 - Lowest level is by DASD volume
- **DB2 table and index space refresh automation**
 - Refreshes specific table and index spaces
 - Lowest level is by data set



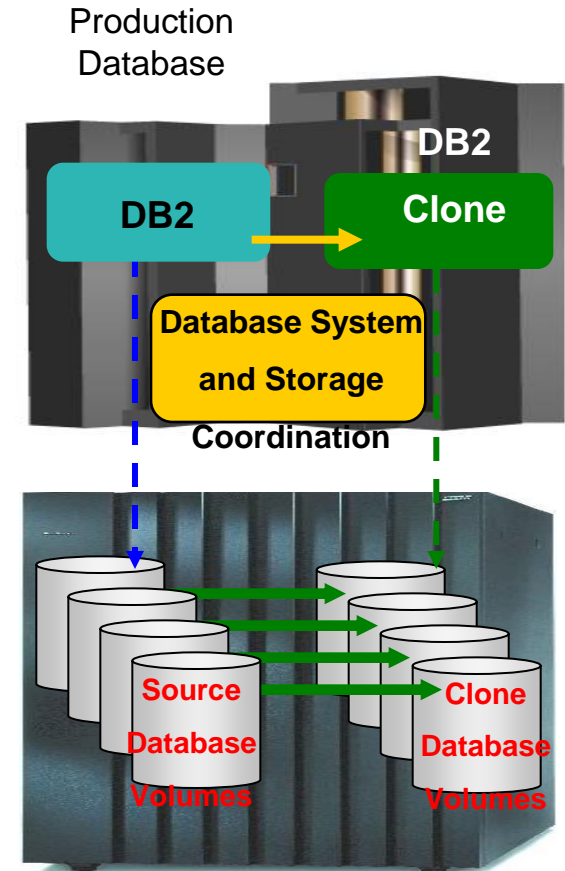
Use Cases for Refreshing DB2 Table and Index Spaces

- **Offload business reporting to a DB2 clone**
 - Improve production performance
- **Refresh data in a test, development or quality assurance environment**
 - Development refresh
 - To stage data-warehouse loads
- **To aid in problem determination**
 - Troubleshoot a production problem
- **Refresh table spaces into a previously cloned DB2 skeleton**

Clone DB2 Systems Using DB2 Cloning Tool

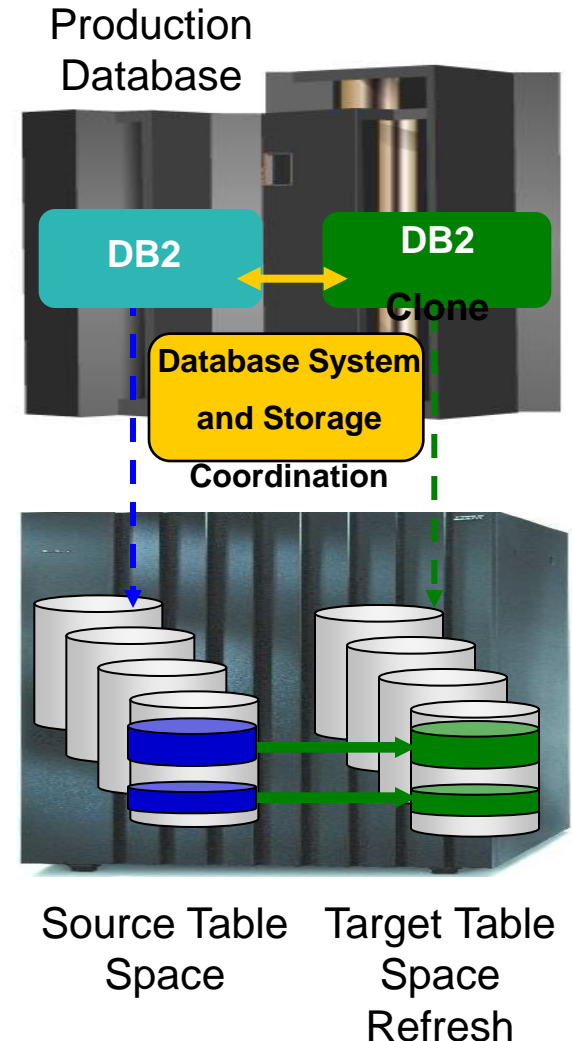
DB2 System Cloning Automation

- **Performs automated cloning of DB2 systems**
- **DB2 data copied using storage-based dataset fast-replication**
 - Data can be cloned while online or offline
- **Performs rapid volume reconditioning and data set renaming on cloned volumes to solve the data access challenges**
 - Target volumes retain their target volume label
 - Renames the VTOC, VTOCIX, and VVDS to match the target volume
 - Renames and catalogs all data sets to a new HLQ
- **Adjusts target DB2 system to accommodate and accept the cloned data**
 - DB2 catalog, directory, BSDS, active / archive Log
 - Makes data accessible on the same or shared LPAR



Refresh DB2 Table and Index Spaces Using DB2 Cloning Tool - Table and Index Space Refresh Automation

- **Performs automated DB2 table and index space refresh operations**
 - DB2 RI relationships, LOBS, and Identity columns
 - XML on DB2 V9 or greater
- **Verifies source target database compatibility**
- **DB2 data copied using storage-based dataset fast-replication**
 - Data can be cloned while online or offline
 - Slow copy mechanism (DFSMSdss etc.) can be used
- **Performs object ID translations and target system meta-data**





Target locations that DB2 Cloning Tool supports

- **Within the same DB2 system or to another DB2 system**
- **To the same database name or to another database**
- **To the same table space name, or to another table space name**
- **To the same Creator ID or to another Creator ID**



Data Masking

- **Option to mask one or more columns during the table space refresh process**
- **The masking changes are made during OBID translation step based on masking rules that are enabled during the copy**
- **All referential integrity (RI) columns can have same masking function applied**
 - Examples of fields that a user might change are US--- >Social Security numbers< --- credit card numbers, names and addresses



Data Masking - examples

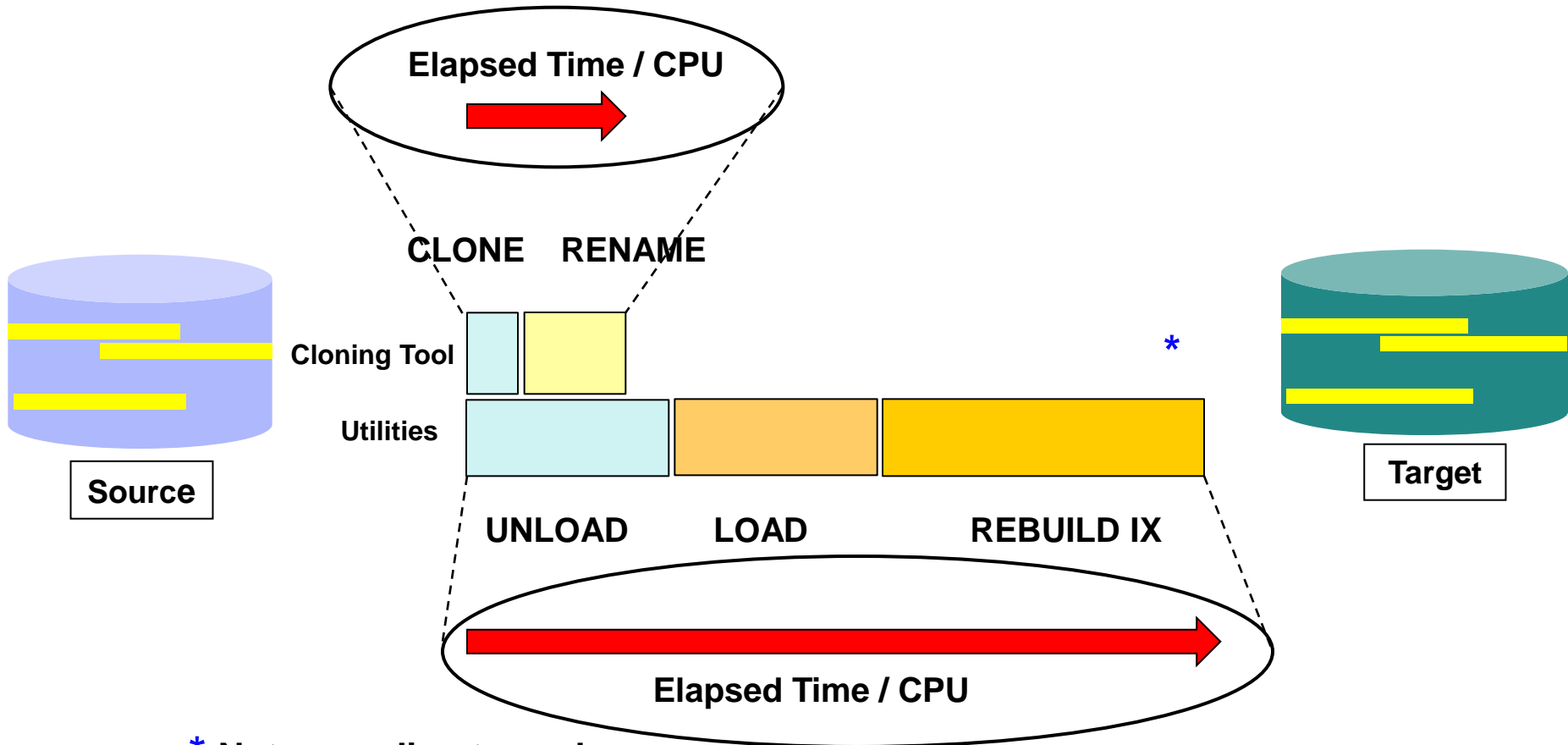
- **Types of data masking functions are provided, such as:**
 - *PATTERN RULE (Sir | Mr) Bill*
 - *SCRAMBLE RULE , SCRAMBLE(FIELD)*
 - *USER EXIT RULE , FIELD = USER_EXIT()*
 - *STATIC RULE , FIELD = CONSTANT VALUE*
 - *MASK RULE, FIELD = [a-z0-9]*10*
 - *RANDOM RULE, FIELD = RAND(1, 100)*
 - *SEQUENCE RULE , SEQ(1, 1)*
 - *CURRENT DATE, CURRENT TIME, CURRENT TIMESTAMP RULES*
 - *CURRENT USER RULE*



DB2 Cloning Tool benefits

- **DB2 Cloning Tool simplifies and automates database administration tasks**
 - Integrates and coordinates database and storage activities
- **Promote high availability solutions**
 - DB2 Cloning Tool clones DB2 systems fast and effectively
 - DB2 Cloning Tool provides fast and effective refresh operations
- **Integrates storage-based copy utilities (fast-replication or slow copy)**
 - Provides fast and non-intrusive cloning operations
- **Offloads copy to the storage processor (with fast-replication)**
 - Reduces host CPU and I/O resource utilization

Refresh – UNLOAD/LOAD/REBUILD IX vs CLONE

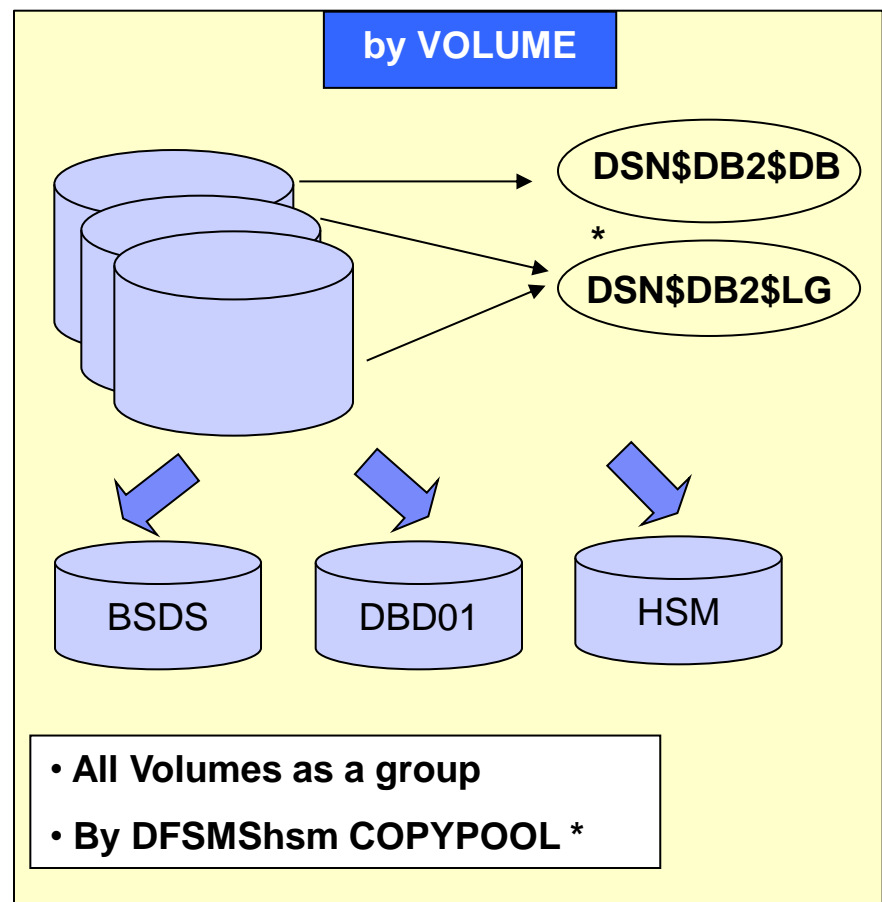
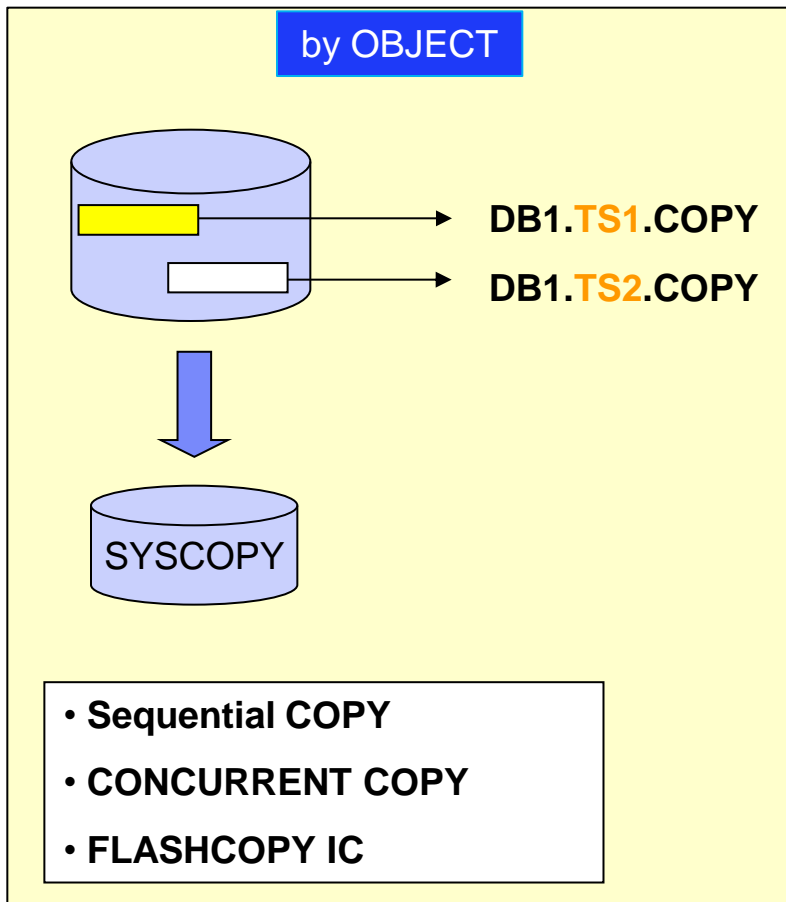
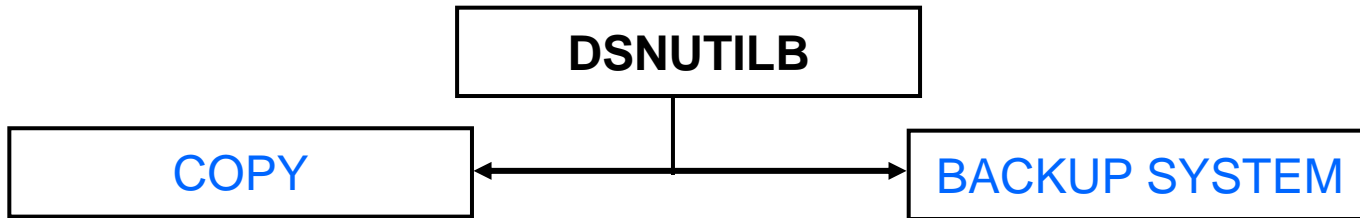


* Not according to scale



COPYing and RESTORing data

DB2 Backup utilities – two flavors



Fast Replication uses by DB2 – DB2 8, 9 and 10

DB2 V8

- BACKUP SYSTEM
- RESTORE SYSTEM
- Dataset FC support for CHECK INDEX SHRLEVEL CHANGE

DB2 9

- Incremental FC support for BACKUP SYSTEM
- Dataset FC for RECOVER with system-level backup (SLB) as input
- Dataset FC for CHECK DATA SHRLEVEL CHANGE and CHECK LOB SHRLEVEL CHANGE

DB2 10

- Dataset FC for COPY
- Dataset FC for inline copy in REORG TABLESPACE, REORG INDEX, REBUILD INDEX and LOAD
- FC image copies with consistency and no application outage (SHRLEVEL CHANGE)
- FCIC accepted as input to RECOVER, COPYTOCOPY, DSN1COPY, DSN1COMP and DSN1PRNT

DB2 COPY utility – sequential copies – 1 of 2

COPY TABLESPACE DSN8D81A.DDS1011B



```

14:14:46.55 DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = DNET963.DNET963D
14:14:46.56 DSNUGTIS - PROCESSING SYSIN AS EBCDIC
14:14:46.57 DSNUGUTC - COPY TABLESPACE DSN8D81A.DDS1011B DSNUM ALL FULL YES SH
14:14:46.72 DSNUBBID - COPY PROCESSED FOR TABLESPACE DSN8D81A.DDS1011B

  NUMBER OF PAGES=3
  AVERAGE PERCENT FREE SPACE PER PAGE = 32.66
  PERCENT OF CHANGED PAGES = 0.00
  ELAPSED TIME=00:00:00
16 14:14:46.79 DSNUBAFI - DB2 IMAGE COPY SUCCESSFUL FOR TABLESPACE DSN8D81A.DDS
  
```

You can use the output as input for:

- **DSN1COMP**
- **DSN1COPY**
- **DSN1PRNT**

Output format



DB2 COPY utility – sequential copies – 2 of 2

COPY TABLESPACE DSN8D81A.DDS1011B CONCURRENT



PAGE 0001 5695-DF175 **DFSMSDSS V1R12.0 DATA SET SERVICES** 2012.016 14:19 PARALLEL
 ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'PARALLEL '
 DUM OPT(3) DATAS(INCL(DSNSCAT.DSNDBC.DSN8D81A.DDS1011B.J0001.A%)) -
 CAN CONC SHA TOL(ENQF) WAIT(0,0) -
 OUTDD(SYSCOPY)
 ADR101I (R/I)-RI01 (01), TASKID 002 HAS BEEN ASSIGNED TO COMMAND 'DUM '
 ADR109I (R/I)-RI01 (01), 2012.016 14:19:16 INITIAL SCAN OF USER CONTROL STATEME
 ADR014I (SCH)-DSSU (02), 2012.016 14:19:16 ALL PREVIOUSLY SCHEDULED TASKS COMPL

You CANNOT use the output as input for:

- ~~DSN1COMP~~
- ~~DSN1COPY~~
- ~~DSN1PRNT~~

Output format



DB2 COPY utility – DB2 10 FCIC

COPY TABLESPACE DSN8D81A.DDS1011B FLASHCOPY YES



FLASHCOPY YES

DSNU421I 016 15:20:51.67 DSNUGFUM - **START OF DFSMS MESSAGES**

PAGE 0001 5695-DF175 DFSMSDSS V1R12.0 DATA SET SERVICES

ADR030I (SCH)-PRIME(0), DCB VALUES HAVE BEEN MODIFIED FOR SYSPRINT

COPY DATASET(INCLUDE(-

DSNBCAT.DSNDBC.DSN8D51A.DSN8S51E.I0001.A001 , -

...

RENAMEU(-

(DSNBCAT.DSNDBC.DSN8D51A.DSN8S51E.I0001.A001 , -

ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'COPY '

ADR109I (R/I)-RI01 (01), 2012.016 15:20:51 INITIAL SCAN OF USER CONTROL STATE

ADR050I (001)-PRIME(01), DFSMSDSS INVOKED VIA APPLICATION INTERFACE

ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK

ADR006I (001)-STEND(01), 2012.016 15:20:51 EXECUTION BEGINS

ADR711I (001)-NEWDS(01), DATA SET DSNBCAT.DSNDBC.DSN8D51A.DSN8S51E.I0001.A001

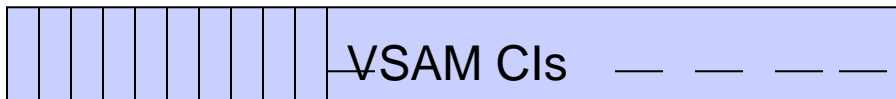
HLQ.DSN8D51A.DSN8S51E.N00001.DNDJZ8MW USING STORCLAS

USRMGMT

ADR755W (001)-PROTD(01), SOURCE DATA SET DSNBCAT.DSNDBC.DSN8D51A.DSN8S51E.I00

TARGET DATA SET HLQ.DSN8D51A.DSN8S51E.N00001.DNDJZ8M

Use COPYTOCOPY to make sequential copy when needed to DASD or TAPE

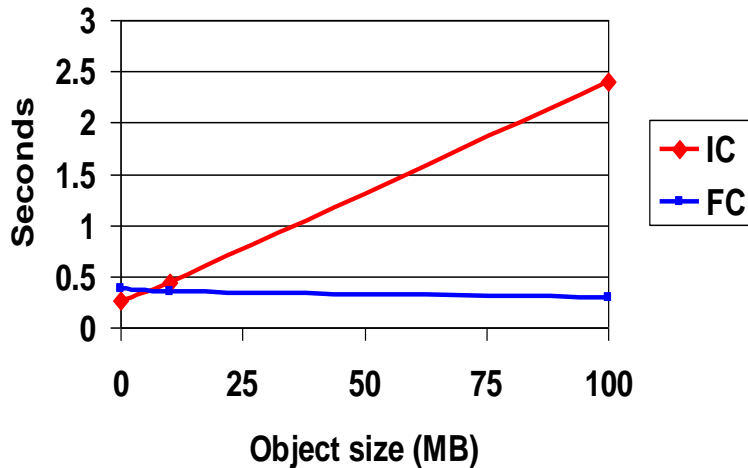


Output format VSAM Cluster

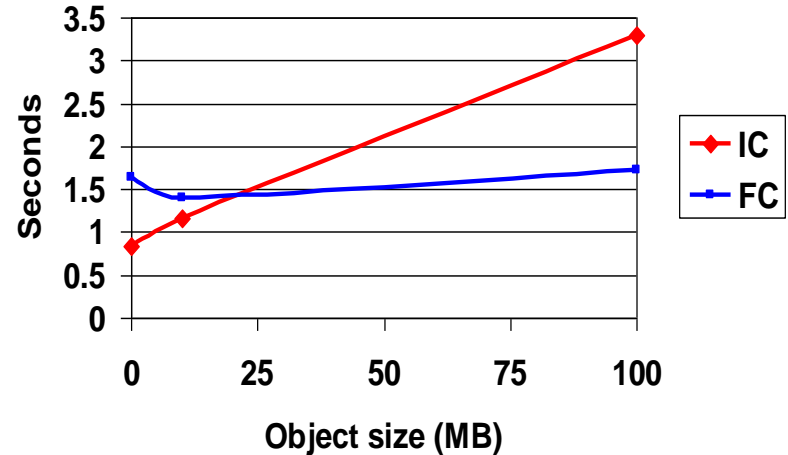
Data set level FlashCopy support

- **Significant CPU and elapsed time savings:**

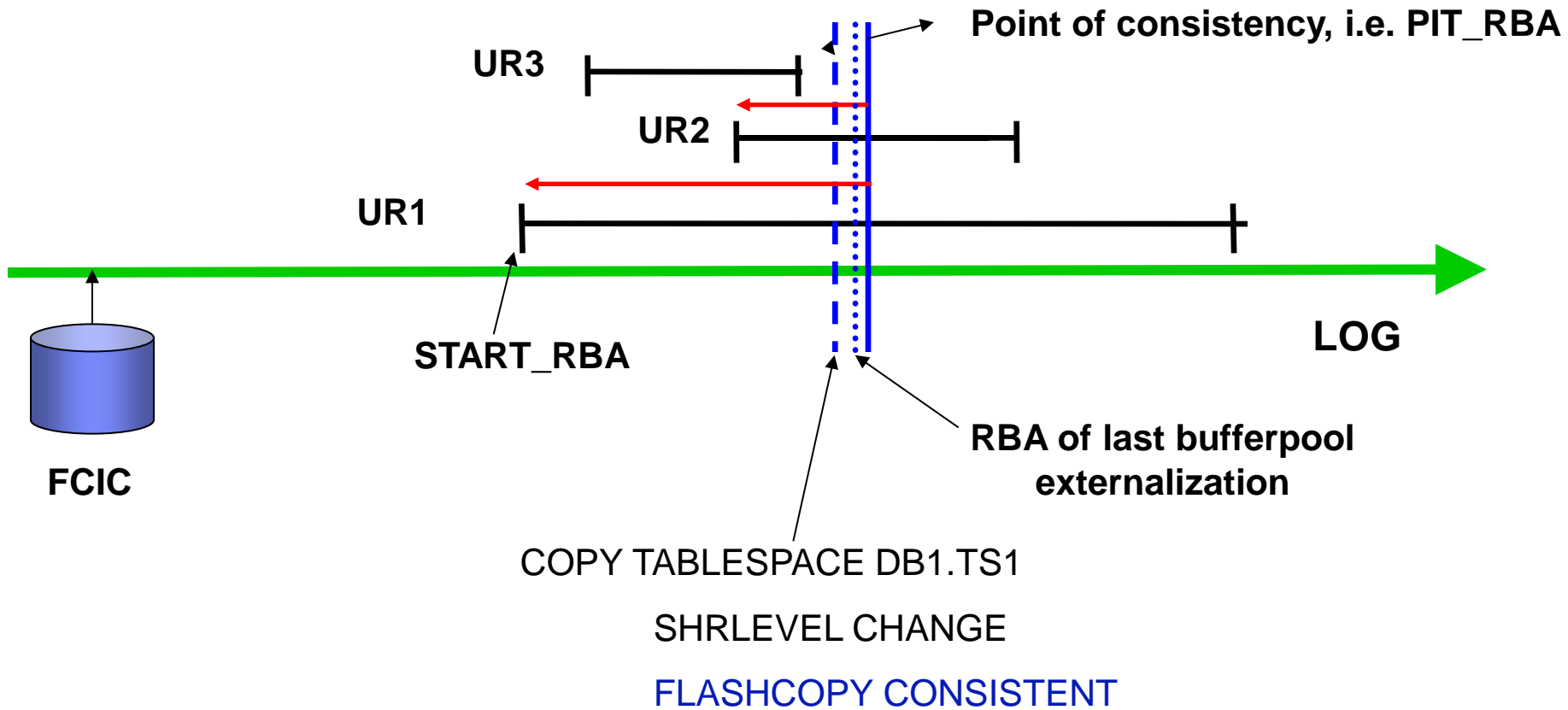
CPU time per object (z10)



Elapsed time per object (z10)



DB2 10 – FCIC CONSISTENT





DB2 10 – new SYSCOPY columns

- Records for FlashCopy copies have an ICTYPE=F and ICBACKUP=FC
- START_RBA is the point at which the pages for the object was externalized to disk
- PIT_RBA is the point of consistency, i.e. more or less the point when the FLASHCOPY relationship was established
- SYSCOPY records for each piece/part.
 - STYPE = T - FlashCopy copy is consistent
 - STYPE = N - FlashCopy copy is not consistent
 - STYPE = Q - Sequential copy is consistent
 - STYPE = U - Sequential copy is not consistent
 - TTYPE - one character indicating type of utility which made the copy
- The SYSCOPY record for a RECOVER with the BACKOUT keyword specified has a ICTYPE= P and a STYPE=B.



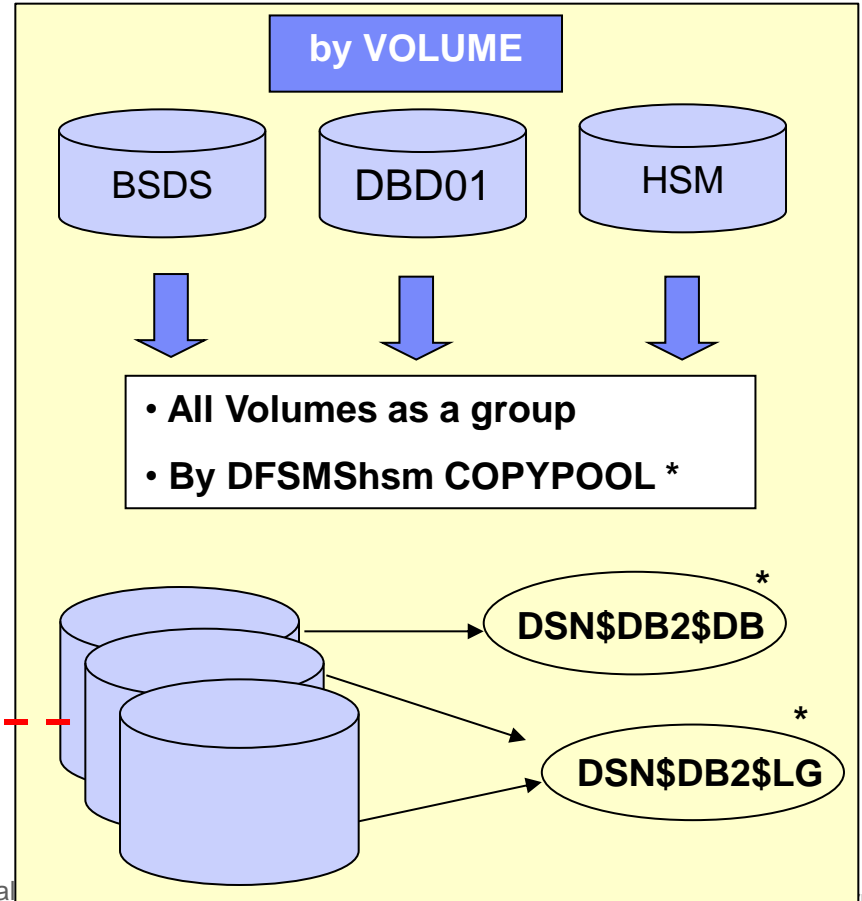
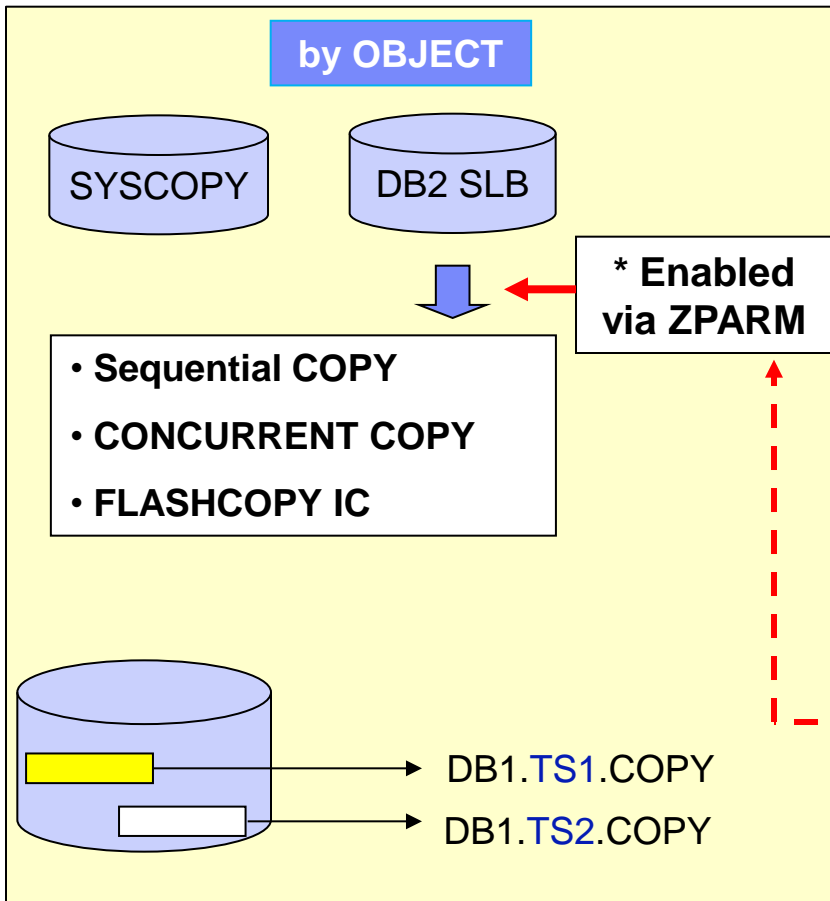
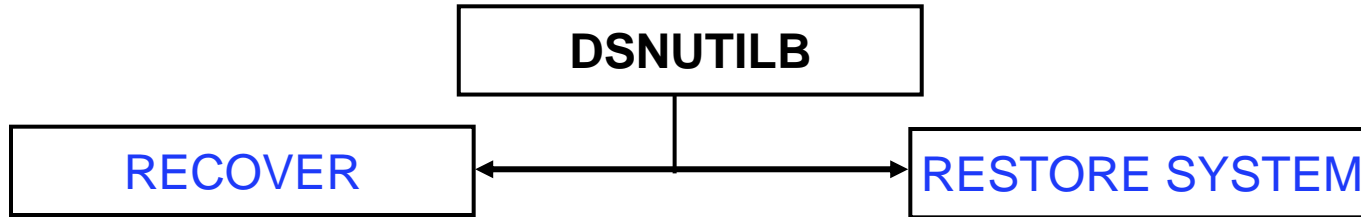
COPY Utility – FCIC considerations

- **FCIC output datasets are VSAM files**
 - To make it a sequential copy run COPYTOCOPY
 - Considerations:
 - DASD space is required for the output copy
 - FCIC to tape not possible
 - GDGs naming conventions not feasible
 - In data mirror environment, there maybe challenges depending on implementation – Metro-mirror, Global mirror (XRC.PPRC)
 - DB2 Recovery Expert solves some of these challenges

DB2 COPY utility vs. Recovery Expert – fast replication copies – differences

Function	COPY Utility	RE FR Copy
Object object level backup using fast replication requirements	Requires DFSMSdss and FC II enabled on DASD	Uses any fast replication for any of the following: IBM - DFSMSdss, IBM FC (all flavors) EMC TimeFinder, HDS Shadow Image (requires TimeFinder) Innovative's FDR
FlashCopy object level backup	FlashCopy Image Copy provided in DB2 10	Fast Replicated copies provided for all versions of DB2
	Registered in SYSCOPY	Registered outside of DB2 – in RE tables
	No GDG capability	Can register generations
	If copy is migrated, it is ignored by RECOVER utility	Supports migrated copy datasets
	Support LISTDEFs and TEMPLATES	Uses RE PROFILES – also support Automation Tool OBJECT PROFILES
	Provides a CONSISTENT SHRLEVEL CHANGE FCIC without impact to applications	Provides a consistent FR copy, but requires a QUIESCE since it runs outside of DB2

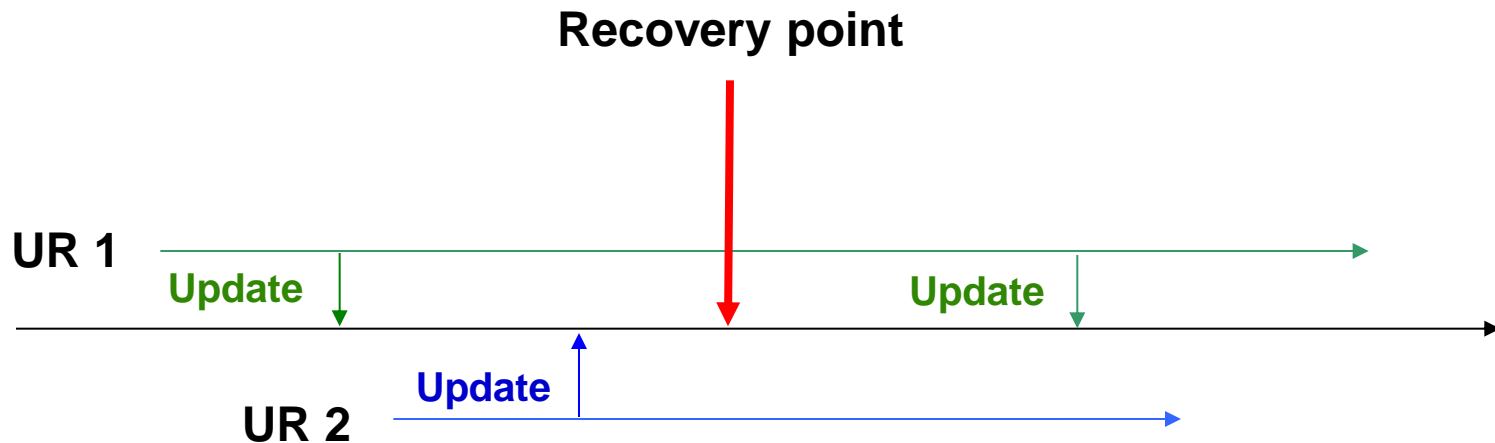
DB2 RECOVER and RESTORE SYSTEM



RECOVER utility

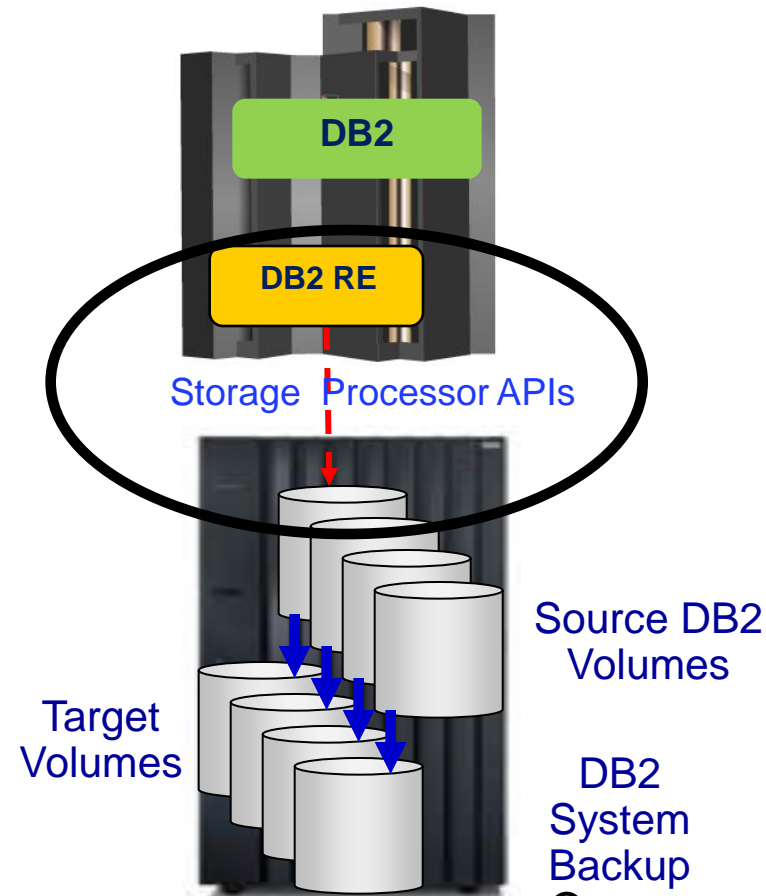
RECOVER to point in time with consistency

- Avoid need for QUIESCEs



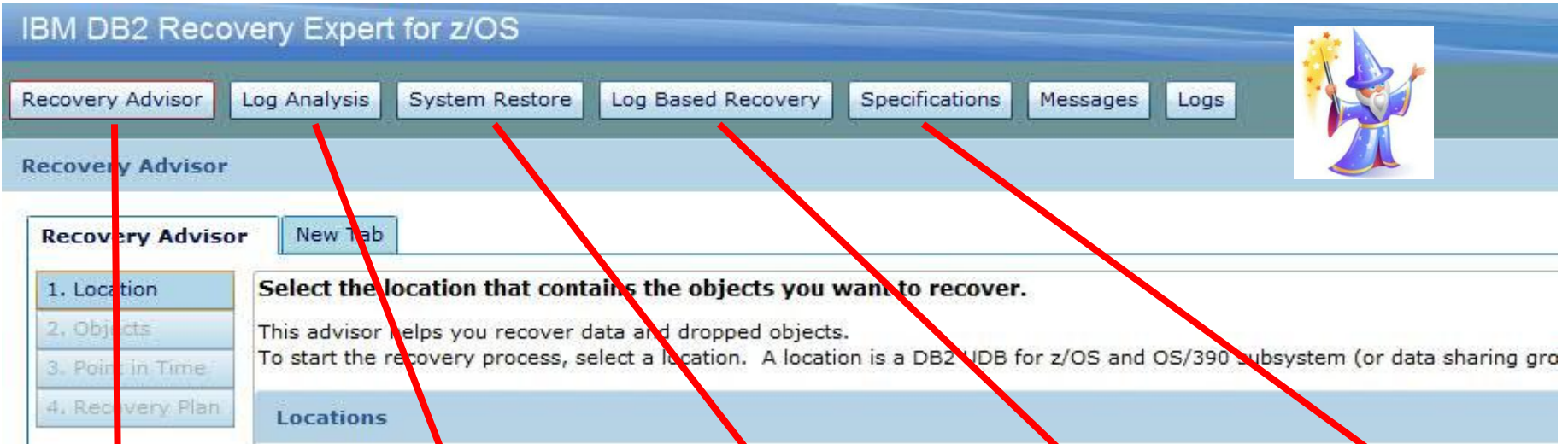
DB2 System Level Backup Overview

- **A System Level Backup is a backup of the entire DB2 environment at a point in time**
 - Can call DB2 BACKUP SYSTEM but does not require it
 - Full System Level Backups
 - Data system Level Backups
 - Partial System Level Backups
- **Leverages storage-based fast replication to drive the volume backup**
 - Backup in seconds
 - Offloading data copy process to the storage processor saves CPU and I/O resources
 - Faster than data set copies
- **Backup DB2 without affecting applications**
 - Backup windows reduced by replacing image copies
 - Extends processing windows
- **Data consistency ensures data is dependent-write consistent**
 - DB2 Suspend
 - Storage-based consistency functions
 - Equivalent to a power failure



Lab testing: 13 TB in .5 Seconds executed on storage processor!

DB2 Recovery Expert – Web GUI Interface



Recovery Advisor wizard allows you to build recovery for any object data and/or DDL

Result will be recovery plans with cost associated to each

Log Analysis wizard allows you to find and store QUIET times based for a set of objects and store this in an RE repository for later use

System Restore wizard allows you to build restore JCL using any SLB supported by RE.

This includes DFSMSdss

The Log Based Recovery Advisor will assist you through the steps of generating DDL and recovering dropped objects at a DB2 location using the DB2 log files.

The Specifications wizard allows you to build profiles of OBJECTS.

You can use these similar to the BACKUP PROFILES in ISPF



RECOVER utility vs. Recovery Expert – *Object level*

Function	RECOVER utility	Recovery Expert
Cost based analysis – “Expert” recoveries	No	Yes
UNDO / REDO recoveries	Provides this via the BACKOUT function using log records	Provides this via an SQL-based function – using log analysis function
Quiet time analysis reporting	No	Yes
Recover from migrated backup	No	Yes
Recovery Plans	No	Yes
Version Recovery	No	Yes
Recover objects from SLB	From DB2 9	Any version of DB2

RECOVER utility vs. Recovery Expert – *Object Level*

Function	RECOVER utility	DB2 Recovery Expert
Recover one table of multi table tablespace	No	Yes – Leaves other tables in tablespace untouched.
DDL Generation for objects and their dependent objects.	No	Yes
Dropped object recovery or DDL generation at all the new object levels	No	Yes. Support for Index, View, Synonym, Alias, Data Type, Trigger, Function, Stored Procedure, Roles
Restore object authorizations	No	Yes
Generate binds for associated Packages and Plans	No	Yes



Managing Utility processing



Take back control of utility processing

- **Most efficient way to manage utility costs is not to run the utility:**
 - Why run REORGs when you get no benefits?
 - Have you looked at your backup strategy to determine read-only/stable table spaces vs active, high-updates table spaces?
 - Do you really need your data in clustering sequence if SQL does not accesses it that way?
 - ...
- **Tools to help you:**
 - DB2 Utilities Enhancement Tool
 - DB2 Automation Tool
 - DB2 Sort
 - DB2 High Performance Unload

DB2 UET – How it helps you ...

- **Utility Monitor**
 - Change utility syntax at execution time
 - No need to make time-consuming syntax changes
 - Do you know where all your utility syntax resides?
 - Do you know when and how utilities are run?
 - Change job return codes based on DB2 messages in the utility SYSPRINT
- **Extend the functionality of the REORG, LOAD and CHECK DATA utilities**
- **Cancel threads**
 - In “Batch Mode” for programs other than DSNUTILB
 - Automatically through the DSNUTILB Intercept when DSNUTILB is run



DB2 UET – Take control of utility syntax

- **Changes utility syntax at run-time based on Policy rules**
- **Enforces the use or disuse of utility parameters**
 - **ADD** parameters that are not present in the utility syntax
 - **REMOVE** parameters that are present and should not be
 - **SUBSTITUTE** given parameters with different parameters
 - **FAIL** the utility based on object name, or user ID
- **Change Job Return Cods based on messages in SYSPRINT**
 - Control the severity of a message
- **Suppress repetitive messages in SYSPRINT**
- **Take no action and Audit who is doing what and when**
 - Identify utility execution and standardize



DB2 UET – Extending utility functions

- **REORG**
 - Automatically adding a MAPPINGTABLE at runtime
- **LOAD**
 - PRESORT data to reduce elapsed time
 - Change data by using CONSTANT or VALUEIF
- **CHECK DATA**
 - Discard data rows to a flat file
 - Flat file is in UNLOAD format
 - Post-process the flat file



DB2 UET – Managing Threads

- 1. Cancel threads automatically when DSNUTILB is invoked**
 - Policy rules determine which objects to cancel threads on
 - Policy enables many types of filter criteria
 - No JCL changes are required!
- 2. Cancel threads for any program using batch job steps**
 - Allows any program exclusive access to an object
- 3. ISPF Interface**
 - See all thread activity on the subsystem
 - Filter threads to find the one you want



DB2 UET - Utility Monitor (1)

Save your DBAs time and effort by enabling them to take complete control of your utility execution from end to end:

1. Cancel threads to ensure utilities execute the first time
2. Add another layer of security and prevent users from running utilities they shouldn't or with syntax they shouldn't
3. Customize job return codes for different DB2 messages to get the results you want



Db2 UET - Utility Monitor (2)

- **Provides an additional layer of security:**
 - Protect business critical objects:
 - Example: Prevent someone from running MODIFY against business critical objects
 - Protect your recovery strategy
 - Example: Ensure an object is recoverable by enforcing REORG with KEEPDICTIONARY
- **Take advantage of new DB2 10 utility syntax without changing utility syntax**
 - Save time and resources - avoid changing a batch job
- **Audit all DSNUTILB executions – see who is executing what**



DB2 UET - Extend the DB2 Utilities

- **Save on CPU and elapsed time by letting UET create the mapping table during a REORG**
 - Reduce DBA intervention by automatically dropping the mapping table and index when the REORG is done
- **Save on CPU and elapsed time by using PRESORT during a LOAD**
 - Integrate with DB2 Sort
- **Reduce DBA intervention by using DB2 Automation Tool to generate CHECK DATA on related sets of objects, then use UET to discard data rows to a flat file**



DB2 UET – Cancel Threads

- **Reduce application downtime by enabling access to business-critical objects**
- **Reduce repeated CPU consumption by ensuring utilities run successfully the first time**
- **Reduce DBA intervention by automatically canceling threads**



How It Works – Eg. DB2 9 Utility changes

```
<DSNUTILB_INTERCEPT>
```

```
<!-- DEFINE THE UTILITIES FOR DB2 9 ON WHICH TO CHANGE SYNTAX -->
```

```
<PRACTICE NAME="UTILITY_RULES_DB2V9">
```

```
<UTILITY NAME="RUNSTATS">
```

```
<MONITOR>
```

```
<SYNTAX ADD="SAMPLE 50"/>
```

```
</MONITOR>
```

```
</UTILITY>
```

```
<UTILITY NAME="LOAD">
```

```
<MONITOR>
```

```
<SYNTAX ADD="PRESORTED YES" OPTIONIF="PRESORT"/>
```

```
<SYNTAX VALUE="LOG %" SUBSTITUTE="LOG NO"/>
```

```
<SYNTAX ADD="KEEPDICTIONARY"/>
```

```
<SYNTAX ADD="REUSE"/>
```

```
<MESSAGE ID="DSNU1150I" RETURN_CODE="4"/>
```



How It Works – Eg. DB2 10 best practice

```
<!--DEFINE THE UTILITIES FOR DB2 10 ON WHICH TO CHANGE SYNTAX -->
<PRACTICE NAME="UTILITY_RULES_DB2V10">
  <UTILITY NAME="RUNSTATS">
    <MONITOR>
      <SYNTAX REMOVE="SAMPLE %"/>
      <SYNTAX ADD="TABLESAMPLE SYSTEM AUTO"/>
    </MONITOR>
  </UTILITY>
  <UTILITY NAME="LOAD">
    <MONITOR>
      <SYNTAX ADD="PRESORTED YES" OPTIONIF="PRESORT"/>
      <SYNTAX REMOVE="FORMAT SPANNED YES"/>
      <SYNTAX VALUE="LOG %" SUBSTITUTE="LOG NO"/>
      <SYNTAX ADD="KEEPDICTIONARY"/>
      <SYNTAX ADD="REUSE"/>
      <MESSAGE ID="DSNU1150I" RETURN_CODE="4"/>
    </MONITOR>
  </UTILITY>
</PRACTICE>
```



Tooling which compliments DB2 Utilities



DB2 Sort for z/OS

- **DB2 Sort provides high speed utility sort processing for data stored in DB2 for z/OS. It improves sort performance while optimizing overall system efficiency by exploiting the advanced facilities of the z/OS operating system and System z to provide the highest level of ROI.**
- **DB2 Sort leverages the strengths of the System z platform, DB2 for z/OS and the DB2 Utilities Suite to drive:**
 - Significant savings in elapsed time and CPU during utility sort processing, especially LOAD, REBUILD INDEX and REORG
 - Relief from application constraints of large volumes of data in highly-transactional workloads performing numerous insert, update and delete operations against DB2 for z/OS databases
 - Sophisticated disk allocations reduce Sort Capacity Exceeded errors caused by large data volumes and/or inaccurate statistics
 - Improved sort processing for other IBM products such as DB2 Utilities Enhancement Tool, DB2 High Performance Unload, DB2 Recovery Expert, DB2 Log Analysis Tool and DB2 Change Accumulation Tool

DB2 Sort 1.3 Performance Benefits

Use of DB2 Sort 1.3 with DB2 utilities, as compared with running DB2 utilities alone, may see: *

- Reduction of Sort CPU usage
 - Up to 74% reduction on machines with zIIP engines
 - Up to 43% reduction on machines without zIIP engines
- Reduction of Utility CPU usage
 - Up to 49% reduction on machines with zIIP engines
 - Up to 25% reduction on machines without zIIP engines
- Reduction of Utility Elapsed Time
 - Up to 50% reduction on machines with zIIP engines
 - Up to 49% reduction on machines without zIIP engines



* The information contained on this slide is distributed AS IS. Performance data and results presented were determined in various controlled laboratory environments, using specific, limited test configurations, and are for reference purposes only. Tests were run against the most current versions of DB2 Sort and DB2 Utilities Suite generally available as of Aug. 17, 2012. Results reported for machines with zIIP engines reflect a situation where all DB2 Sort program zIIP eligible instructions are successfully dispatched to execute on available zIIP processor(s). The results that may be obtained in other operating and production environments may vary significantly. Users of the product should verify the applicable results they might achieve for their specific environment.



DB2 Sort – many uses

- New capabilities:
 - Multi-record processing to reduce overhead between DB2 Sort and tools and utilities that call DB2 Sort
 - VALUE → Reduces CPU usage and enables greater zIIP offload.
 - Support for Native I/O to allow DB2 Sort to read SORTIN and write SORTOUT.
 - VALUE → Reduces CPU and elapsed time by reducing data movement.
 - Support for Change Accumulation Tool
 - VALUE → Support for more DB2 Tools enables better overall performance
- Extending your IBM DB2 Utilities and Tools investment with performance benefits:
 - **LOAD, REBUILD INDEX, REORG, RUNSTATS, CHECK INDEX / DATA / LOB**
 - **DB2 Utilities Enhancement Tool 2.2**
 - **DB2 High Performance Unload 4.2**
 - **DB2 Log Analysis Tool 3.3,**
 - **DB2 Recovery Expert 3.1**
 - **DB2 Change Accumulation Tool 3.1**
 - **DB2 Utilities Suite 9 and 10**



DB2 Sort 1.3 Beta - Customer Results

- European IT provider wants to reduce costs and maintain service levels as they anticipate growth in data and number of users

Percent Improvement Using DB2 Sort						DB2 Sort 1.3 Install Verification Program Results	
Object Size (Billion Bytes)	DB2 Utility	Sort CPU Time	Step CPU Time	Step Elapsed Time	DB2 Sort zIIP Offload %	Environment	
8	LOAD	68.4%	27.0%	5.0%	52.2%	LPAR Name	P11
8	REBUILD	71.6%	47.3%	10.0%	58.6%	Machine Type	2817 607
8	REORG	43.5%	39.6%	8.1%	26.6%	Operating System Level	1.12
10	LOAD	68.9%	27.1%	-3.0%	53.0%	DB2 Level	10
10	REBUILD	72.5%	47.6%	18.3%	59.4%	Number of Regular Processors	4
10	REORG	43.5%	39.5%	11.7%	27.6%	Number of zIIP Processors	2
25	LOAD	71.7%	38.0%	27.0%	52.8%	Amount of Memory	80G
25	REBUILD	75.5%	54.4%	49.8%	58.6%	"Above the Line" Region	1,400M
25	REORG	55.1%	50.8%	46.5%	29.7%		
40	LOAD	74.7%	39.8%	28.1%	56.8%		
40	REBUILD	75.3%	54.8%	52.2%	58.2%		
40	REORG	54.9%	50.9%	31.0%	31.3%		
	Averages	65.8%	45.5%	32.9%	47.1%		

Average DB2 Sort % Improvement by Utility				
DB2 Utility	Sort CPU Time	Step CPU Time	Step Elapsed Time	Average % zIIP Offload
LOAD	73.0%	37.2%	23.6%	53.7%
REBUILD	75.0%	53.7%	47.3%	58.7%
REORG	53.3%	49.2%	34.2%	28.8%



IBM DB2 Solution Packs for z/OS

▪ Why Solution Packs? One price, one PID, easier install, more integration

– Can receive credit for products already purchased

– **DB2 Administration Solution Pack:**

- DB2 Administration Tool, DB2 Object Comparison Tool, DB2 Table Editor, Optim Configuration Manager

– **DB2 Utilities Management Solution Pack:**

- DB2 Automation Tool, DB2 Utilities Enhancement Tool, DB2 Sort, DB2 High Performance Unload

– **DB2 Performance Solution Pack:**

- OMPE for DB2 PE, Optim Query Workload Tuner, DB2 SQL Performance Analyzer, Query Monitor

– **DB2 Fast Copy Solution Pack:**

- DB2 Recovery Expert, DB2 Cloning Tool



Summary

- **Unparalleled investment in utilities**
- **IBM Utilities Suite is essential for exploitation of DB2 function including major availability enhancements in DB2 10**
 - Support of core DB2 function from day 1 of GA
- **Expect continued delivery of enhancements on release boundary, and in maintenance stream when prudent**
- **Continued focus on:**
 - Elimination of application impact from utilities
 - Elapsed time & CPU consumption reduction
 - Resource consumption reduction
 - Reduction in complexity & automation improvements
 - Synergy with IM Tools



Thank
YOU