



SWG BetaWorks

DB2 9 for z/OS

Technical Education Series

“Availability”

BetaWorks

Important Disclaimer

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY.

WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE.

IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION.

NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, OR SHALL HAVE THE EFFECT OF:

- CREATING ANY WARRANTY OR REPRESENTATION FROM IBM (OR ITS AFFILIATES OR ITS OR THEIR SUPPLIERS AND/OR LICENSORS); OR
- ALTERING THE TERMS AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENT GOVERNING THE USE OF IBM SOFTWARE.

Agenda

- **Online Create and Rebuild Index**
- **Volume based utility enhancements**
- **Serviceability Enhancements**
- **RENAME column**
- **Refresh Early Code**
- **Automatic creation of objects**
- **SMS**



SWG BetaWorks

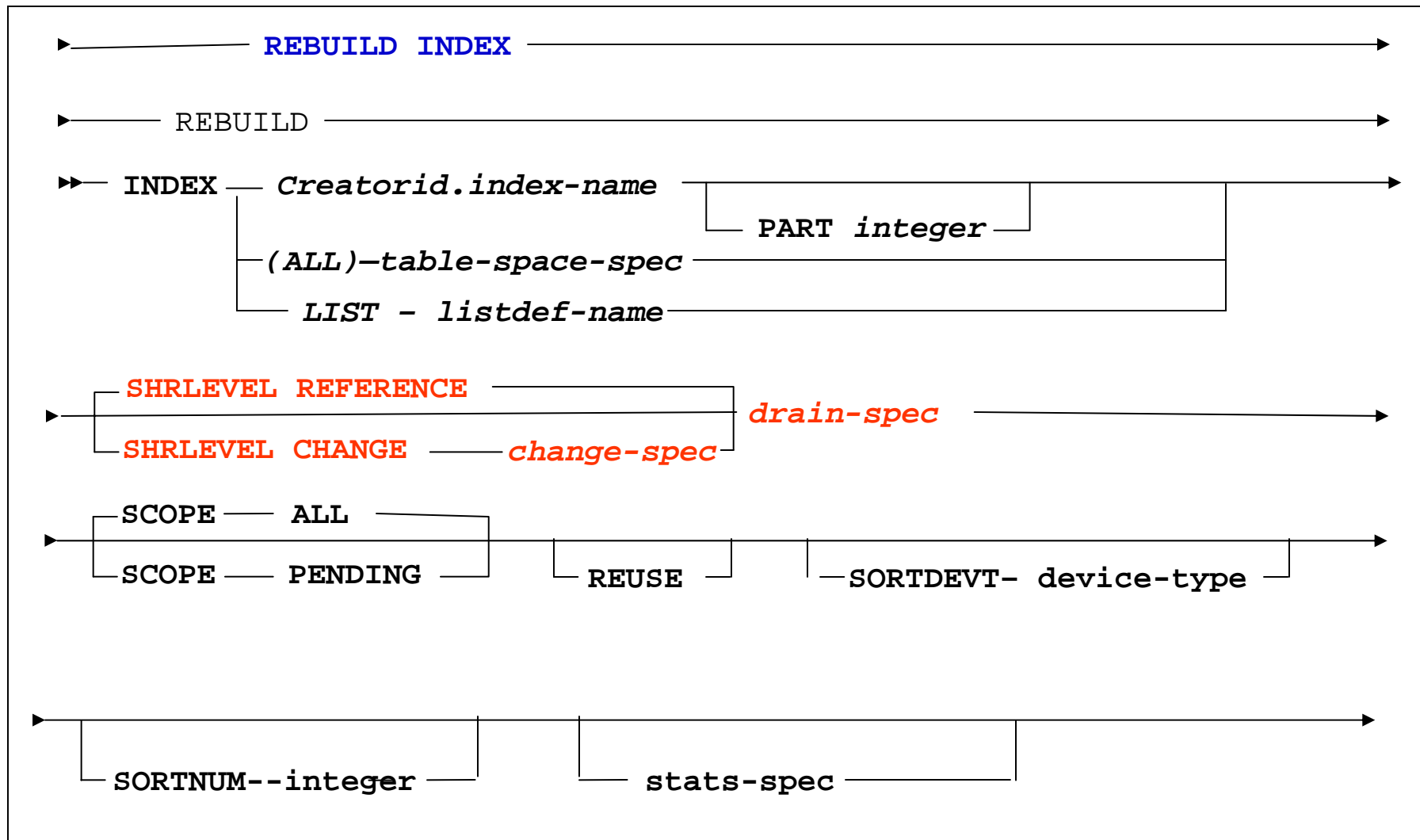
Online Create and Rebuild Index

Improving availability and reducing outages

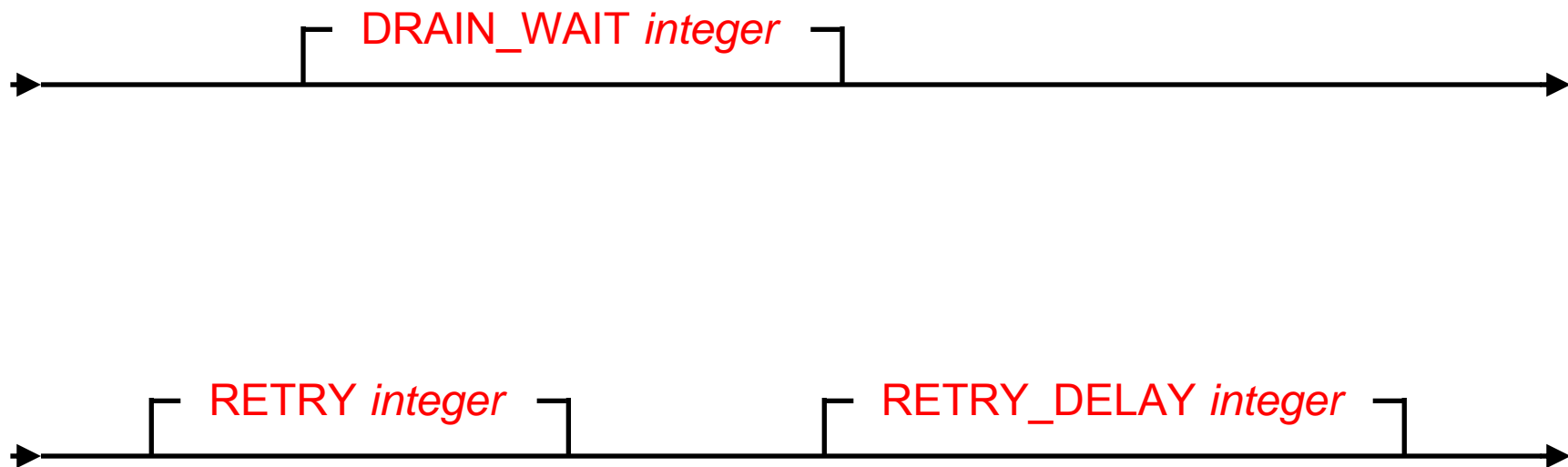
- Creating an index on existing large table has limitations (DSNDB07).
- **CREATE INDEX with DEFER YES**
 - When the index is created with DEFER YES, it should not be available for use by queries.
 - However, customer uses DEFER YES to speed up create index
- **Run the REBUILD INDEX utility**
 - As this utility drains the write claim class, no other thread can update the table until the utility ends.
 - However, the rebuild still runs for a long period of time...
 - During which queries get **unavailable resource** on the new, not yet rebuilt, index.

Restricted availability of applications for a very long time.

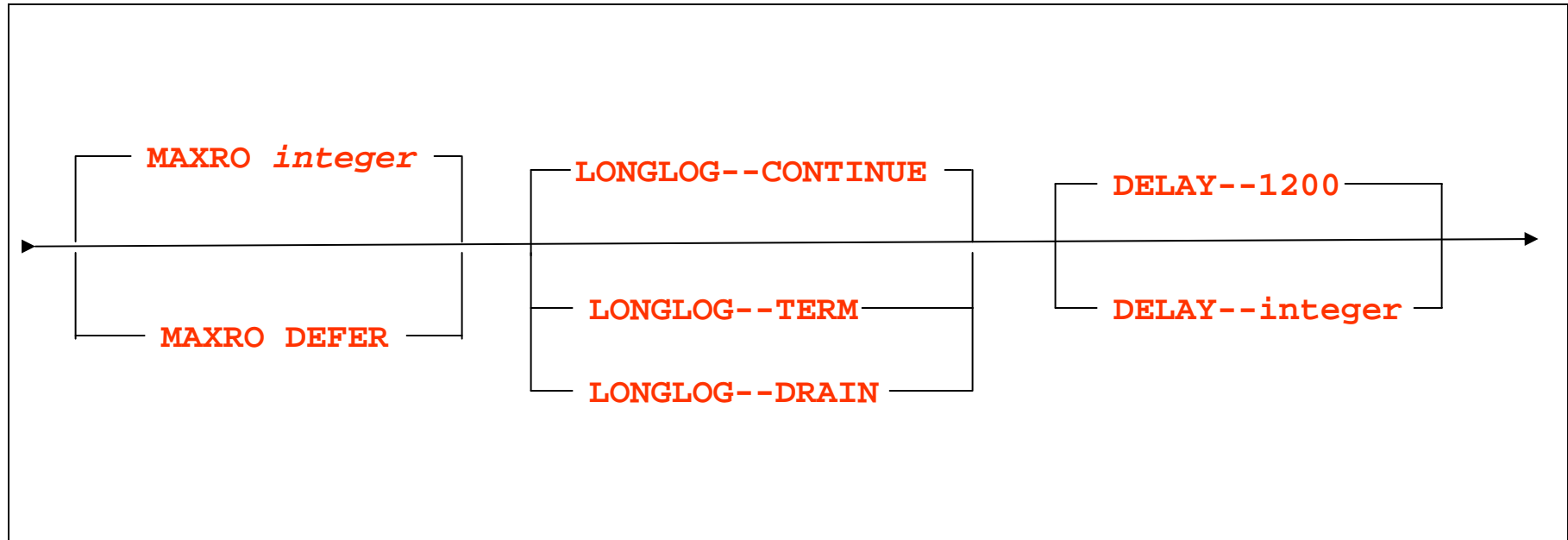
REBUILD INDEX Syntax



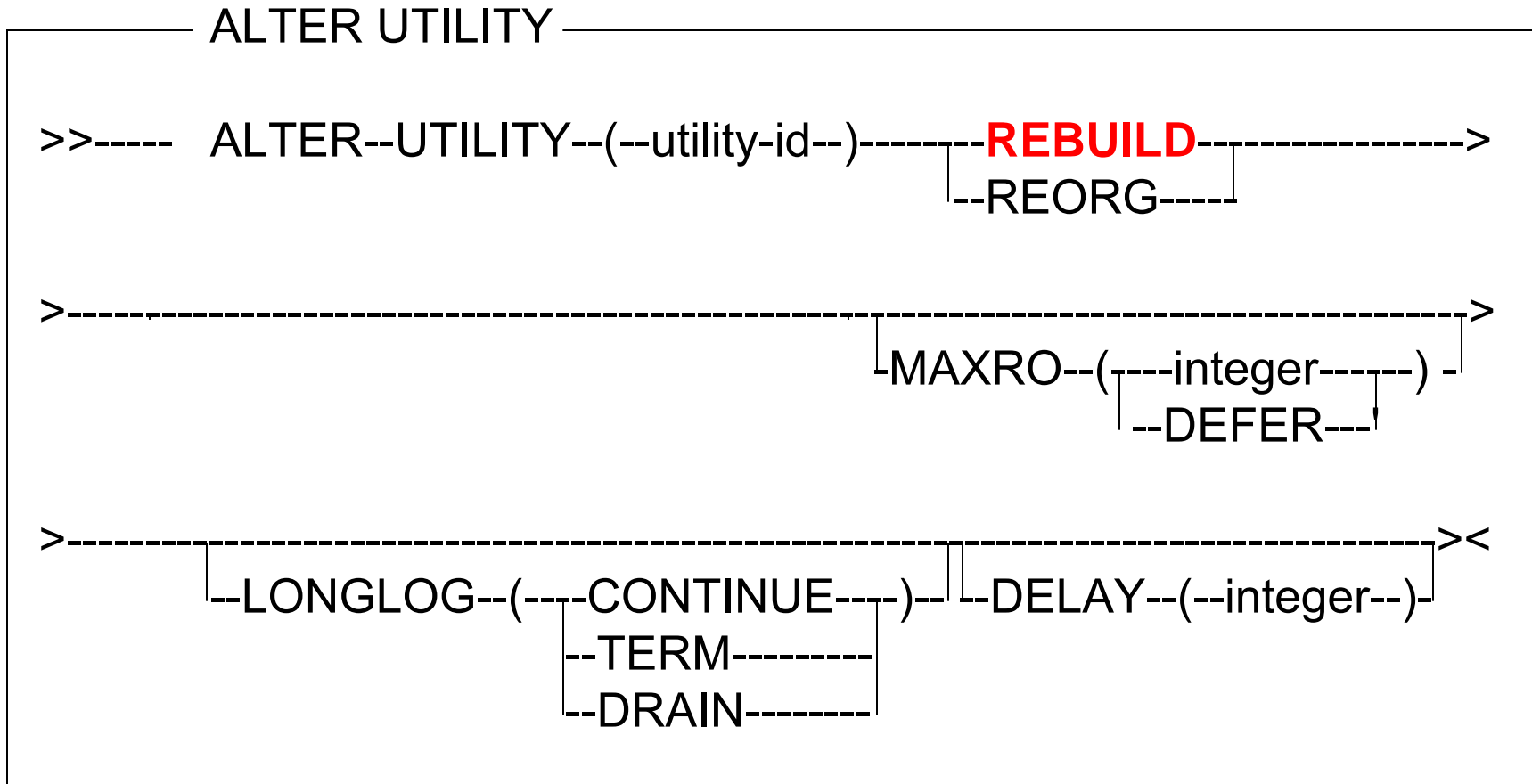
REBUILD INDEX Syntax – DRAIN specification



REBUILD INDEX SYNTAX – Change Specification



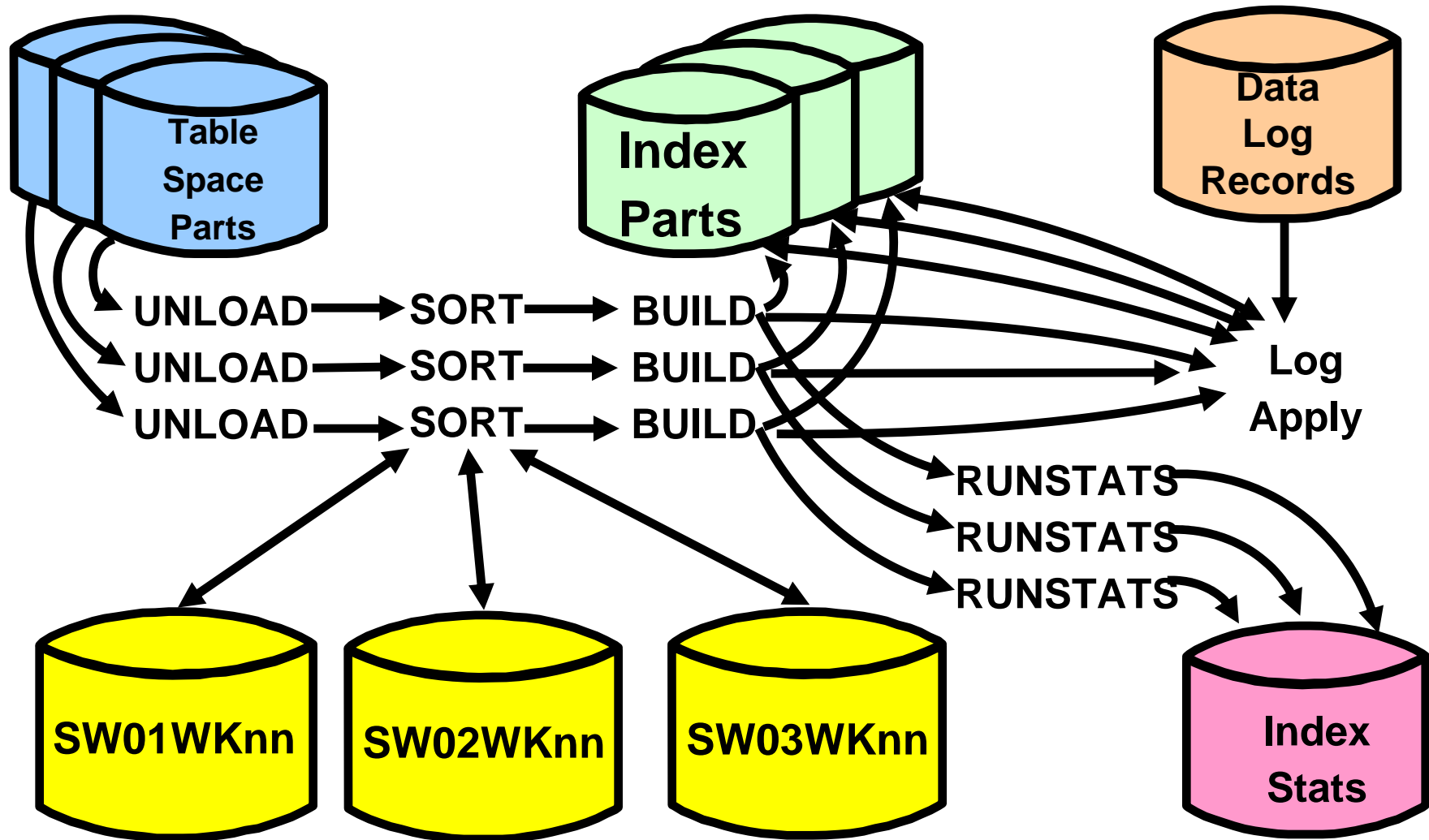
ALTER UTILITY



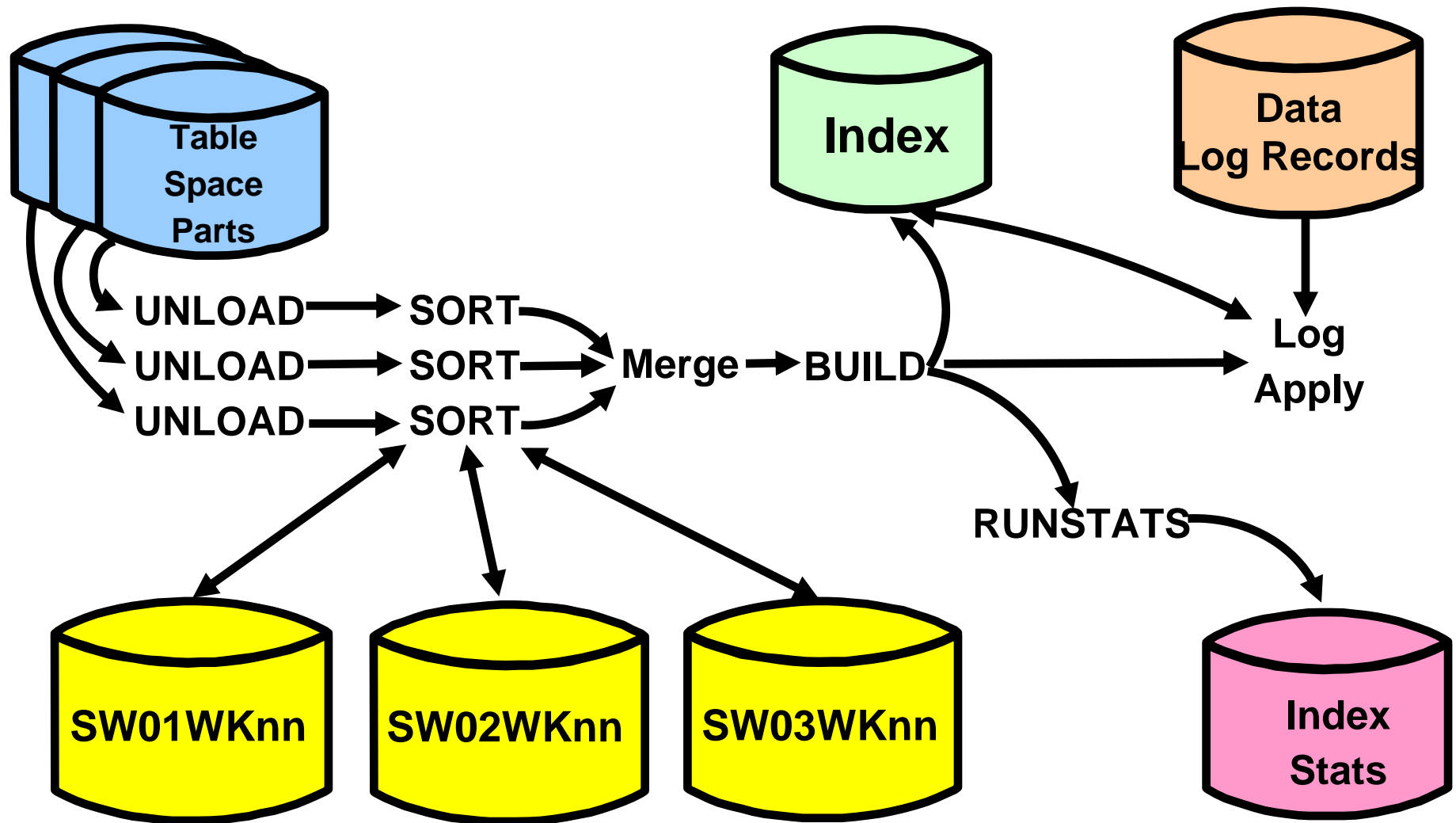
Online REBUILD INDEX - Processing

- **Claim data and drain index(es)**
 - Turn on full logging for data
 - Set index(es) in RBDP and dedrain
- **Unload data isolation UR and extract keys**
- **Sort keys and build index as usual**
- **Bring index to currency by:**
 - Reading the data (Insert / Update / Delete) log records
 - Extracting index (Insert / Delete) changes
 - Sorting index changes by obid / key / part / lrsn
 - Apply changes via DSNKCON / DSNKDEL
- **Drain data writers to finish last of log**
- **Dedrain data**
- **Reset RBDP for index(es)**

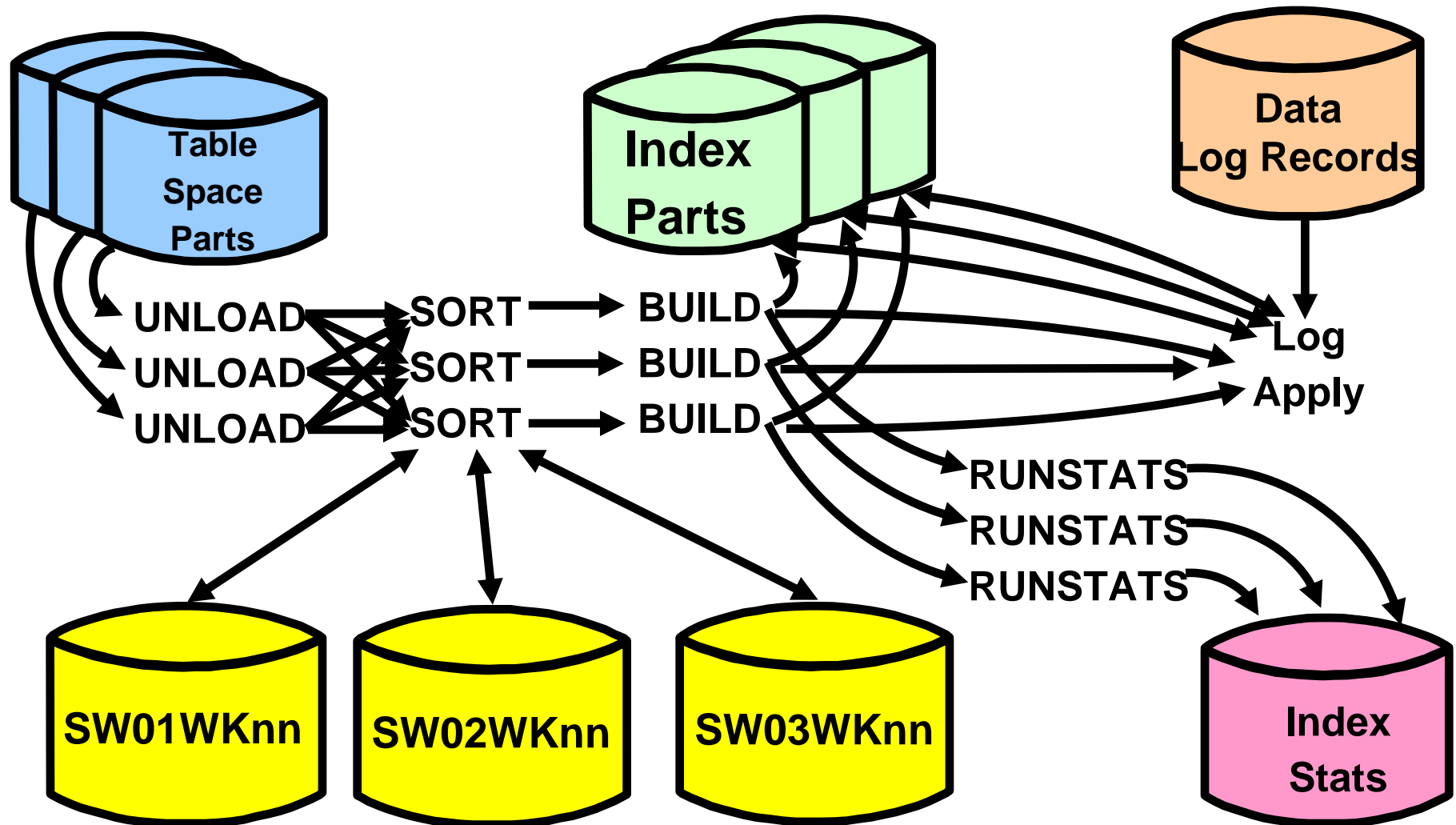
Rebuild Partitioning Index



REBUILD Single NPI on Partitioned



REBUILD ALL on Partitioned





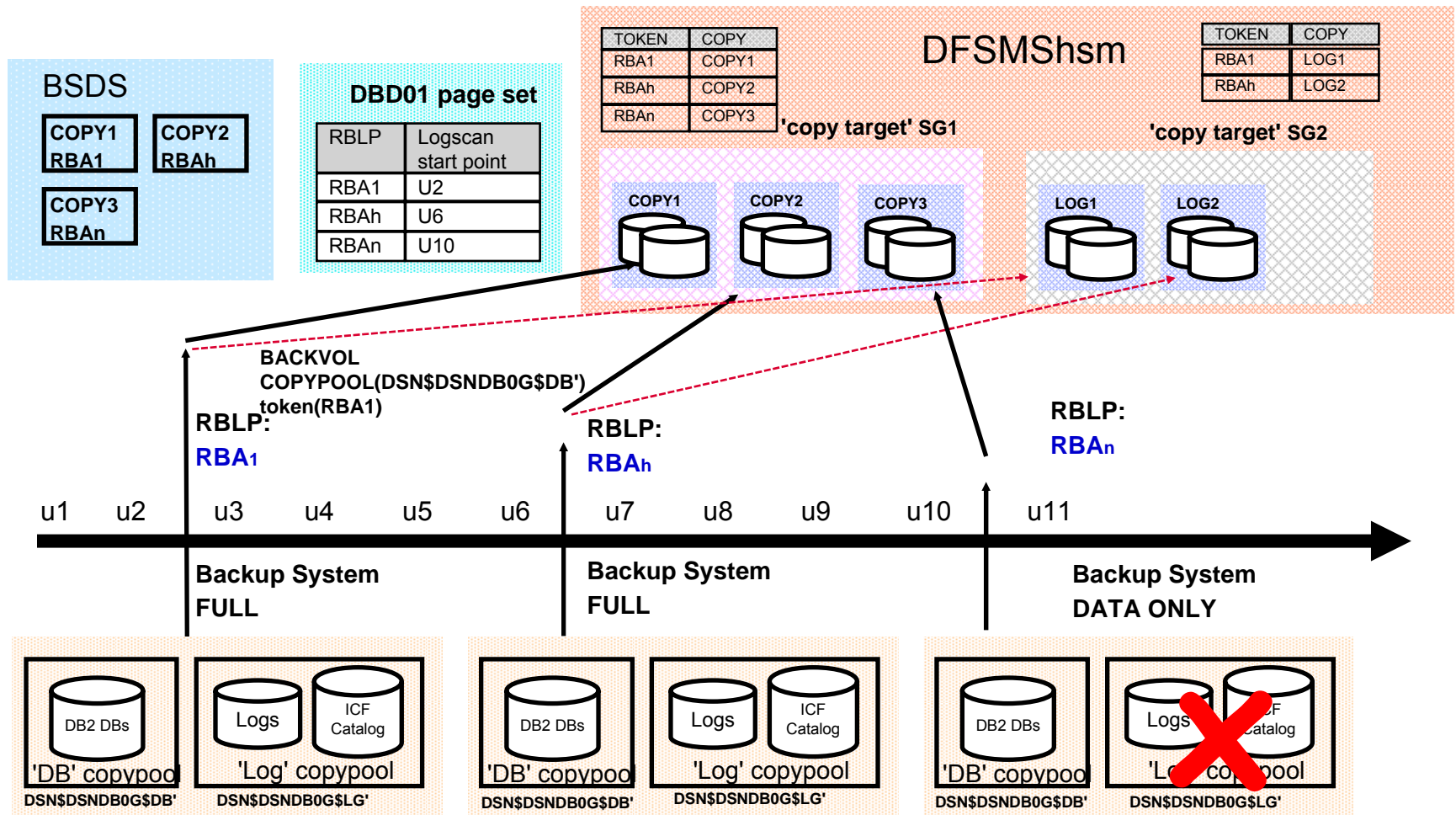
SWG BetaWorks

Volume based utility enhancements

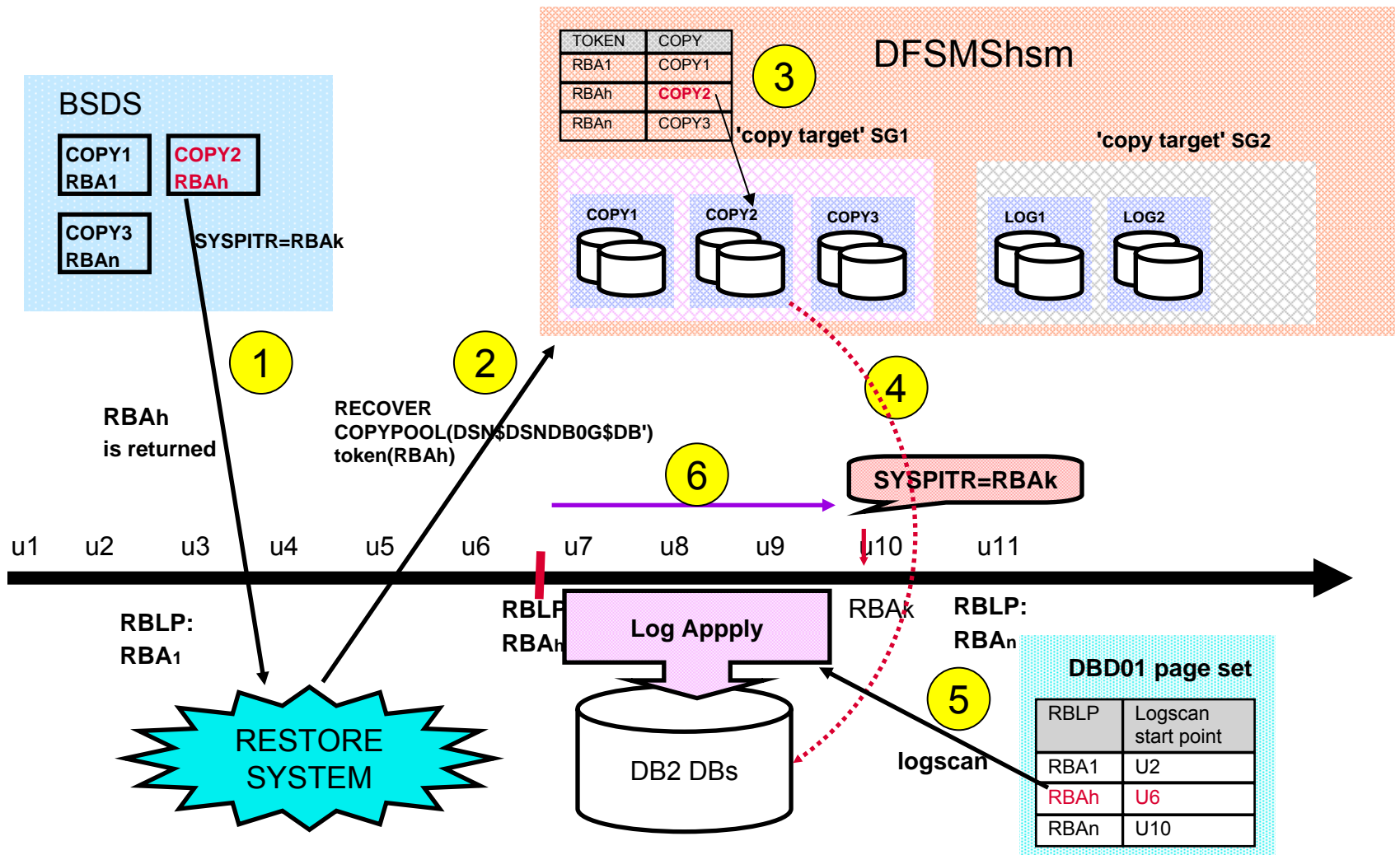
Foundation provided by DB2 V8

- DB2 for z/OS V8 introduced two new system-level Utilities:
 - **BACKUP SYSTEM**
 - **RESTORE SYSTEM**
- Support for system-level PIT recovery of large and complex DB2 systems that are utilized by a single application.
- Primarily Enterprise Resource Planning (ERP) environments
 - SAP NetWeaver, Siebel, or PeopleSoft that use DB2 UDB for z/OS as the database server.
- Exploited features of z/OS V1.5 DFSMSHsm services for backing up and restoring groups of DASD volumes as copy pools (Database & Logs)
 - Minimal disruption using the FlashCopy fast replication technology provided by the IBM TotalStorage Enterprise disk systems.

BACKUP SYSTEM



RESTORE SYSTEM



Requirement - Automated volume backups in DB2

- Today's DASD subsystems provides the means of very fast volume-based backups.
- What does not exist is DB2 for z/OS integration into the associated z/OS tape management products: (DFP, HSM, DSS).
- Backup/recovery in the SAP environments is the most frequently raised issue by our SAP/390 customers.
 - Typically large systems
 - SAP applications demand continuous availability
 - Customers require fast, non-disruptive and automated procedures for performing these database administration operations.
 - Particularly with regard to recovery granularity.



Enhancements in DB2 9

- **BACKUP SYSTEM** and **RESTORE SYSTEM** utilities will take advantage of a new **DFSMSHsm** feature provided by z/OS 1.8
 - **DFSMS Fast Replication Tape Support Stage 2**
 - Manage the dump/restoration of fast replication copies of system-level backups (SLB) to and from tapes

- The **RECOVER** utility will also be enhanced to invoke **DFSMSHsm**
 - **DFSMS Fast Replication Data Set Recovery Stage 3**
 - Can use a SLB as a recovery base for an object or a list of objects
 - SLB on disk or one dumped to tape

Customer benefits – Reduced tape management

- Removes need to develop processes to manage dump and restoration of fast replication copies of DB and LOG copy pools to and from tape

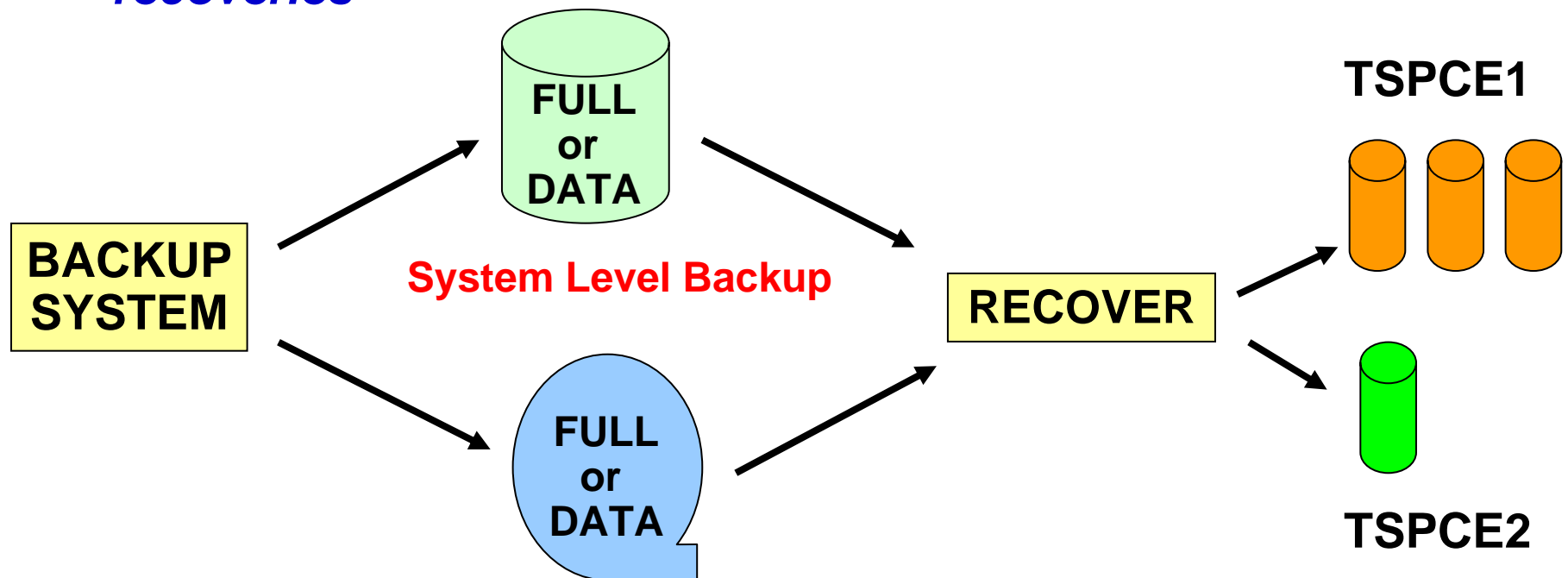


- Consider using the tape backup options for:
 - Managing disk space
 - Retaining fast replication copies for long term use
 - An additional source of recovery from media failure

Customer benefits – Recovery enhancement

- RECOVER utility enhancements permit using a backup taken at the system-level as the recovery base for a subset of objects in the system

Connecting system-level backups with object level recoveries



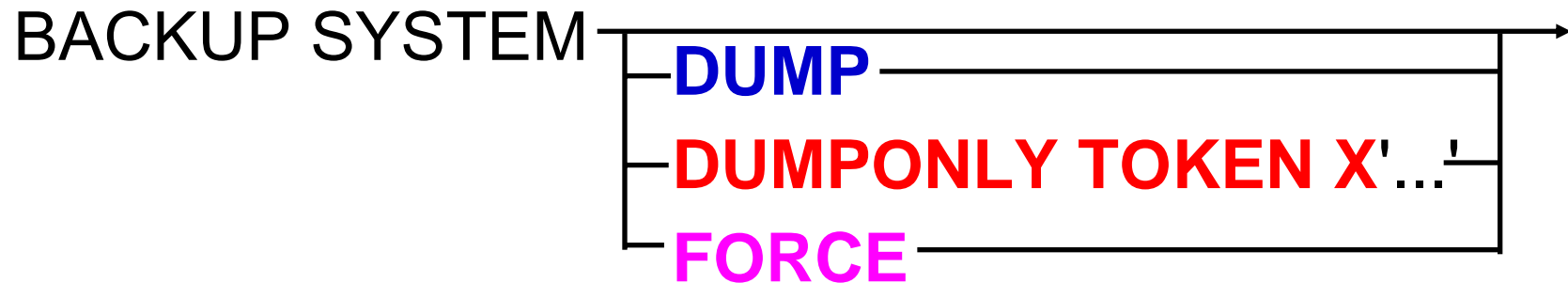
BACKUP SYSTEM utility enhancement

- User can now specify whether dump processing of the disk copies to tape should be initiated when the fast replication of the database and/or log copy pools are logically complete.

- Up to five DFSMSHsm dump classes may be specified.

- Optionally, invoke the BACKUP SYSTEM Utility twice:
 - Once to initiate the fast replication copies of the copy pools
 - Then a second time to initiate the dump processing to tape.

BACKUP SYSTEM syntax



- **DUMP** - initiate dump processing to tape when the fast replication of the copy pool(s) is logically complete
- **DUMPONLY TOKEN** - "restart" the dump processing for a SLB if the dump has failed
- **FORCE** - allow a SLB whose required dumps have not completed to be overwritten with a new SLB
 - ▶ **DUMPCCLASS** sub option also available on both DUMP and DUMPONLY

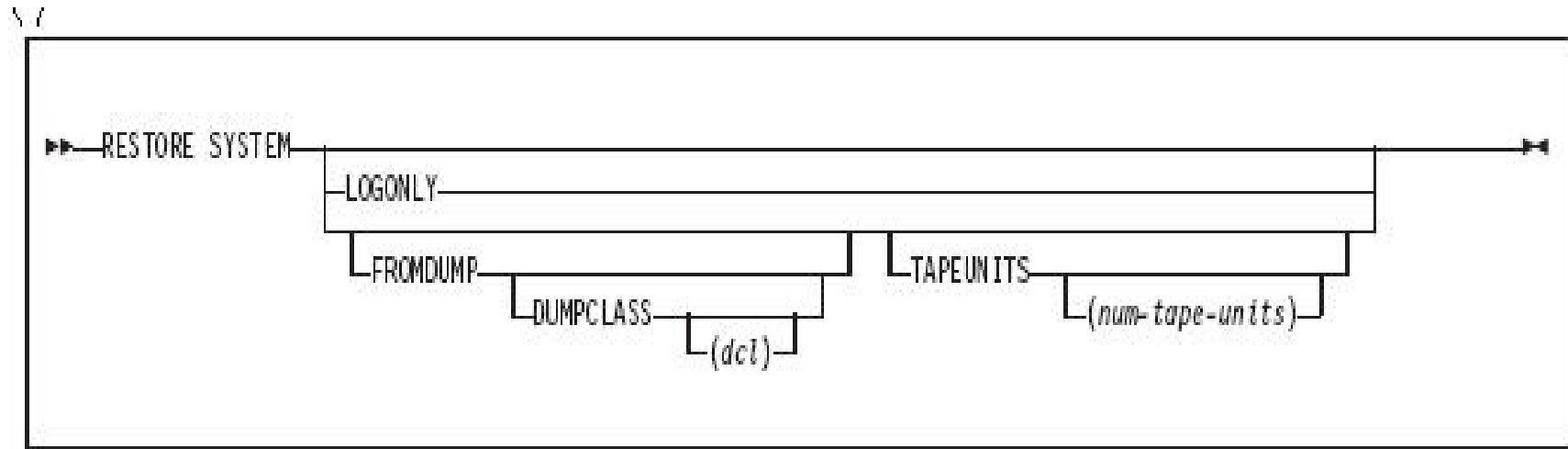
Managing System level Backups on Tape

- If the TOKEN option is omitted, then the most recent-level backup is dumped.
- The DUMPCCLASS option can be specified for both DUMP and DUMPONLY to process particular DFSMSHsm dump classes.
- Ensure that system-level backups dumped to tape do not reside on the same tape volumes as image copies or concurrent copies taken by the DB2 COPY Utility.
 - Problems may occur if the RECOVER Utility requires access to both the system-level backups and the copies at the same time.

RESTORE SYSTEM

- Enhanced to restore the database copy pool from a SLB that has been dumped to tape
- Restore of the DASD volumes from tape is done
 - via DFSMSDss
 - in parallel, the Utility determines the optimal degree of parallelism
- If SLB resides on DASD and on tape, then the DASD version is chosen as the recovery base
- Install ZPARM options
 - FROMDUMP - user can specify that they do not want to use the DASD version
 - DUMPCLASS (dc) - user can specify a specify HSM dump class to restore from

RESTORE SYSTEM syntax



- **FROMDUMP** – Restore only requires dumps on tape of the DB copy pool
- **TAPEUNITS** – Specifies the tape drives required during restore of DB copy pool

DSNTIP6 – A new utilities install panel

The entries on this panel affect the execution of DB2 Utilities.

```

DSNTIP6          INSTALL DB2 - DB2 UTILITIES PARAMETERS
===>
Enter system-level backup options for RESTORE SYSTEM and RECOVER below:

| 1 SYSTEM-LEVEL BACKUPS ===> NO           As a recovery base: YES or NO
| 2 RESTORE/RECOVER       ===> NO           From dump (NO or YES)
| 3 DUMP CLASS NAME       ===>             RESTORE/RECOVER from dump
| 4 MAXIMUM TAPE UNITS    ===> NOLIMIT     For RESTORE SYSTEM: NOLIMIT or 1-255

Enter other DB2 Utilities options below:
5 TEMPORARY DS UNIT NAME===> SYSDA        Device for temporary utility data sets
6 UTILITY CACHE OPTION   ===> NO          3990 storage for DB2 utility IO
7 STATISTICS HISTORY     ===> NONE        Default for collection of stats history
8 STATISTICS ROLLUP      ===> NO          Allow statistics aggregation. NO or YES
9 UTILITY TIMEOUT        ===> 6           Utility wait time multiplier

PRESS:  ENTER to continue  RETURN to exit  HELP for more information

```

New DSNZPARM options (1)

- **SYSTEM_LEVEL_BACKUPS YES | NO**
 - Whether the RECOVER utility should use SLBs generated by the BACKUP SYSTEM Utility as a recovery base

- **RESTORE_RECOVER_FROMDUMP NO | YES**
 - Use the SLB from disk (NO), or from the dump on tape (YES)
 - Can be overridden:
 - FROMDUMP keyword on the RESTORE SYSTEM Utility
 - RECOVER Utility statement when it is executed.

New DSNZPARM options (2)

- **UTILS_DUMP_CLASS_NAME** **blank (default) or a DFSMShsm dump class name**
 - DUMPCLASS that will be used by:
 - RESTORE SYSTEM utility to restore from a SLB that was dumped to tape.
 - RECOVER utility to restore objects from an SLB that was dumped to tape.
 - Applicable only when you specify YES in the RESTORE/RECOVER field above.
 - Can be overridden with the DUMPCLASS keyword on the utilities at execution time.
- **RESTORE_TAPEUNITS** **NOLIMIT (default) or 1 – 255**
 - The maximum number of tape units that the RESTORE SYSTEM Utility can allocate when restoring a SLB that has been dumped to tape.
 - NOLIMIT means that the RESTORE SYSTEM Utility will allocate as many tape units as necessary to restore the system level backup.
 - Can be overridden with the TAPEUNITS keyword on the RESTORE SYSTEM Utility at execution time.

RECOVER Utility - SYSTEM_LEVEL_BACKUP YES

- Most recent recovery base (prior to the recovery point) is chosen:
 - could be image copy, concurrent copy, log yes event, or SLB

- If FROMDUMP is specified or SLB is on tape:
 - HSM is invoked with parallelism to restore the datasets

- If FROMDUMP is not specified:
 - If SLB is a recovery base for one or more objects and it exists on DASD, then HSM is invoked to restore the datasets.
 - HSM caps the degree of parallelism depending on system parms
 - If the restore of datasets from a SLB on DASD fails, then the recovery of the object will not proceed *

* note: RECOVER does not automatically go to the dump copy of the SLB or to a previous SLB, as the problem may be correctable

MODIFY RECOVERY Utility

- Delete SYSLGRNX records according to AGE or DATE criteria even if there are no SYSCOPY records to delete
- Insert a new SYSCOPY record ICTYPE=M, STYPE=R with START_RBA= highest RBA/LRSN of the SYSCOPY or SYSLGRNX records deleted

Miscellaneous items

- **HSMPRINT dataset**
 - Temporary dataset for output from the HSM **LIST COPYPOOL DUMPVOLS** command
 - Used by BACKUP SYSTEM, RESTORE SYSTEM, & RECOVER Utilities when SLB dump information is needed

- **Gathering SLB / Recovery information**
 - Execute **DSNJU004** (PRINT LOG MAP Utility) on each member to get SLB info
 - HSM **LIST COPYPOOL DUMPVOLS** command for information on DASD versions, dumps, and dump classes
 - **REPORT RECOVERY** Utility at the object level



SWG BetaWorks

Serviceability Enhancements

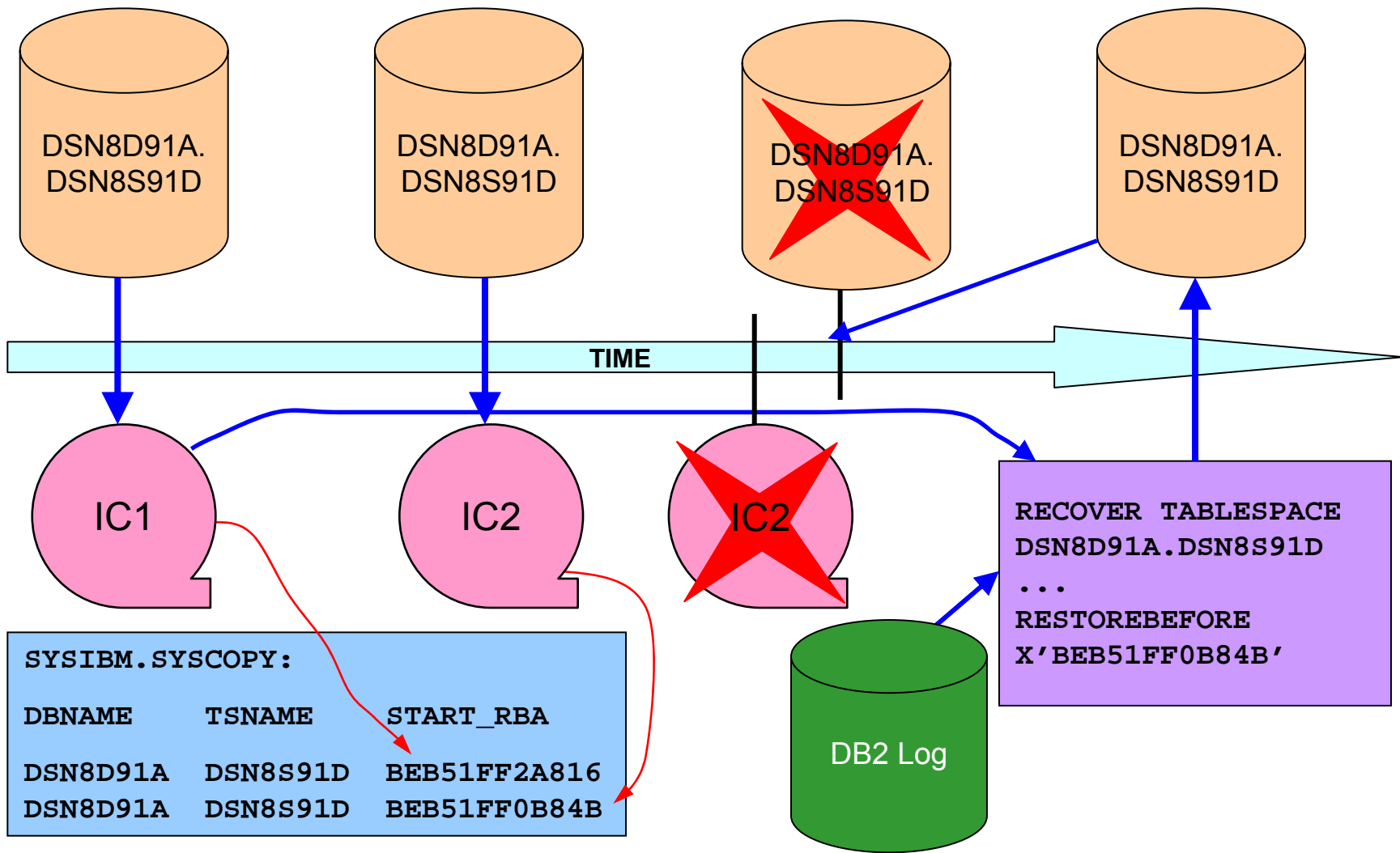
RECOVER RESTOREBEFORE

- The DB2 RECOVER utility now allows you to recover an object, but avoid using a specific image copy as the recovery base.

```
◀—— RECOVER ... .. RESTOREBEFORE X'byte-string' ——▶▶
```

- For example, where an image copy data set has been lost, because the corresponding row is still present in SYSIBM.SYSCOPY, RECOVER always attempts to allocate the data set.
- This is an alternative to restoring to an older but known good image copy using DSN1COPY and running RECOVER with LOGONLY.
- New RECOVER keyword RESTOREBEFORE(RBA or LRSN) allows you to specify the RBA or LRSN of a missing or unwanted image copy.
- The RECOVER utility searches for a full image copy or concurrent copy with an RBA or LRSN value earlier than that specified by RESTOREBEFORE to use as a recovery base.
- It applies any appropriate incremental image copies and the log records to restore the object to current.
- This enhancement is available in compatibility mode.

RECOVER ... RESTOREBEFORE



New Deadlock/Timeout Messages

- When a utility deadlock or timeout condition occurs, additional messages are issued in the utility job output.

```
DSNU120I csect-name DEADLOCK INFORMATION: INTERVAL COUNT =n,  
NUMBER OF WAITERS = m. LOCK NAME = lcknm TYPE = type  
FUNC/STATE/DURATION = f st dur  
HOLDER/WAITER = h / w PLAN-ID = planid CORR-ID = corrid MEMBER NM = mbrnm
```

```
DSNU121I csect-name TIMEOUT INFORMATION: NUMBER OF HOLDERS/WAITERS =m,  
TIMEOUT FACTOR = t. LOCK NAME = lcknm TYPE = type  
FUNC/STATE/DURATION = f st dur  
HOLDER/WAITER = h / w PLAN-ID = planid CORR-ID = corrid MEMBER NM = mbrnm
```

- The lock type, function, state and duration values are as documented for the IFCID172 deadlock trace record.
- For timeout, the lock resource function will only be filled in for the requestor (waiter) that received the timeout.
- The utility will abnormally terminate with an ABEND04E and a utility reason code.
- This enhancement is available in compatibility mode.

Julian Date in Utility SYSPRINT Messages

- Utility messages sent to SYSPRINT now include the Julian date and timestamp following the message number, displaying the local time in the form of: DDD HH:MM:SS.TT – for example:

```
DSNU515I 106 13:20:07.11 -DSM1 DSNUCBAL - THE IMAGE COPY DATA SET.....
```

- In a data sharing environment, the data sharing member name follows the timestamp.
- Utility messages that are issued by utility subtasks may not necessarily come out in order.
- This enhancement will be available in compatibility mode.

DSN1PRNT FORMAT and Broken Pages

- **DSN1PRNT** with the **FORMAT** option now continues with the formatting of broken pages in the printed job output.
 - In releases of DB2 prior to DB2 9, when a broken page was encountered by DSN1PRNT with **FORMAT**, a message '**PAGE IN ERROR - NOT FORMATTABLE**' is issued and the broken page is dumped unformatted.
 - With DB2 9 for z/OS, DSN1PRNT with **FORMAT** now issues a message '**PAGE IN ERROR - FORMATTED CONTENTS MAY BE INCONSISTENT**' and then attempts to print the formatted page.
 - In addition, the broken page is also dumped (unformatted) following the formatted attempt.
 - This enhancement is available in compatibility mode.

RECOVER LOGAPPLY Phase Status Information

- -DISPLAY now provides progress status for the RECOVER utility during the LOGAPPLY phase in new message DSNU116I.
- It displays the START RBA / LRSN, END RBA / LRSN and the last COMMIT point LRSN / RBA in the LOGAPPLY phase together with the elapsed time from the start time of the LOGAPPLY phase.
- This provides an estimate for the amount of the log processing that has been completed by RECOVER.

```
DSNU116I DSNUCALA - RECOVER LOGAPPLY PHASE DETAILS: STARTING TIME =  
hh:mm:ss START RBA = xxxx START LRSN = yyyy END RBA = wwww END LRSN = zzzz  
LAST COMMITTED RBA = rrrr LRSN = cccc ELAPSED TIME = hh:mm:ss
```

- This enhancement is available in compatibility mode.

COPY and Broken Pages

- When COPY is run with CHECKPAGE, it performs validity checking for each page, one at a time, in the table space or index space.
- If it finds an error, COPY issues message DSNU518I which identifies the broken page and the type of error.
- If there is more than one error in a page, only the first error is identified.
- COPY also sets new SYSCOPY column TTYPE to 'B' (for broken page).
- COPY continues checking the remaining pages in the table space or index space after it finds an error.
- If any errors are detected, the table space or index space is NOT put in COPY pending.
- Instead, a record with an ICTYPE of 'T' is written to SYSCOPY, to prevent subsequent incremental image copies from being taken.
- CHECKPAGE is the default for COPY.
- This enhancement is available in compatibility mode.

“Unable to Obtain P-Lock” Diagnostics

- DB2 modifies reason code 00C20255 (unable to obtain a P-Lock) and adds new codes to improve diagnosis of this event.
- RC00C20255 new description: “DB2 is unable to obtain a physical lock (P-lock) because another member holding the lock has not responded to the P-lock request.”
 - Old description: “DB2 is unable to obtain a physical lock (P-lock).”
- New RC00C2026A: “DB2 is unable to obtain a physical lock (P-lock) because another member holding the lock has been in DB2 shutdown for an extended amount of time.”
- New RC00C2026B: “DB2 is unable to obtain a physical lock (P-lock) because another member holding the lock has had logging suspended for an extended amount of time, due to either the active logs being full, or to an outstanding SET LOG SUSPEND command.”
- New RC00C026C: “DB2 is unable to obtain a physical lock (P-lock) because another member holding the lock encountered an abend in its P-lock exit.”
- This enhancement is only available in New Function Mode.

-DISPLAY THREAD (*) TYPE(SYSTEM) Report

```

DSNV401I  @ISC9 DISPLAY THREAD REPORT FOLLOWS -
DSNV497I  @ISC9 SYSTEM THREADS - 833
DB2 ACTIVE
NAME      ST A   REQ ID           AUTHID   PLAN      ASID  TOKEN
ISC9      N  *     0 002.VMON     03 SYSOPR           00D6    0
V507-INACTIVE MONITOR, INTERVALS=6821, STG=N/A, BOOSTS=N/A
ISC9      N  *     0 028.ERRMON01 SYSOPR           00D6    0
ISC9      N  *     0 028.RESYNT01 SYSOPR           00D6    0
ISC9      N  *     0 027.OPNACB01 SYSOPR           00D6    0
V490-SUSPENDED 06115-11:15:51.51 DSNLQCTL +00000990 14.16
ISC9      N  *     0 027.DDFINT03 SYSOPR           00D6    0
V490-SUSPENDED 06115-11:15:51.51 DSNLQCTL +00000990 14.16
ISC9      N  *     0 027.DDFINT04 SYSOPR           00D6    0
V490-SUSPENDED 06115-11:15:51.51 DSNLQCTL +00000990 14.16
ISC9      N  *     0 027.SLSTNR02 SYSOPR           00D6    0
ISC9      N  *     0 027.RLSTNR02 SYSOPR           00D6    0
ISC9      N  *     0 027.GQRQST02 SYSOPR           00D6    0
V490-SUSPENDED 06115-11:15:51.52 DSNLQCTL +00000990 14.16
ISC9      N  *     0 028.RESYNC01 SYSOPR           00D6    0
V490-SUSPENDED 06115-11:15:51.53 DSNLQCTL +00000990 14.16
ISC9      N  *     0 020.PEXCTL00 SYSOPR           00D4    0
V490-SUSPENDED 06115-10:55:58.00 DSNTLCTL +000004F0 17.19

```

Improved Startup and Shutdown Messages

- New DB2 startup and shutdown progress messages are provided for the MSTR and DBM1 address spaces as well as the existing ones for the DIST address space and their resource managers; these are issued to the system console.
- **DSNY024I *facility* INITIALIZATION IS COMPLETE**
- **DSNY025I *facility* SHUTDOWN IS COMPLETE**
- **DSNY026I *facility* SHUTDOWN IN PROGRESS**
 - For the above three, *facility* can be MSTR, DBM1 or DIST
- **DSNY027I *resource manager* PROCESSING PHASE *executing process***
 - *resource manager* includes: Agent Services, Recovery Manager, Buffer Manager, Data Manager, LOB Manager, Data Space Manager, and Service Controller.
 - *executing process* phases include: Subsystem Startup, Facility Startup, Release Work, Begin Stop Work, End Stop Work, Facility Shutdown and Subsystem shutdown.



SWG BetaWorks

Rename Column

RENAME COLUMN

```
ALTER TABLE table-name
```

```
RENAME COLUMN source-column-name TO target-column-name
```

- NFM enhancement available under certain circumstances
- Only 1 rename column per ALTER TABLE.
- Avoids dropping and recreating table and associated objects
- All dependent plans and packages invalidated
- Automatic rebind triggered
- Ensure you have the original source code before doing this.
- Need to be aware of external references (third party tools, JCL, recovery jobs, change management tools, DDL, etc).

RENAME COLUMN -- some restrictions

- Can not rename column if its referenced in a VIEW
- Not allowed when a TRIGGER is defined on the table
- Also invalid if table is referenced in an MQT
- Not allowed if the source column has a CHECK CONSTRAINT or FIELDPROC defined on it
- Can not rename columns used in index expressions.
- Can not rename a catalog column.
- See DB2 9 (for z/OS) for z/OS publications for a full list



SWG BetaWorks

REFRESH DB2 EARLY code

-REFRESH DB2,EARLY

- Refreshing the EARLY code without IPL
- The new DB2 command **-REFRESH DB2,EARLY** reloads the EARLY code modules, and rebuilds the EARLY control block.
 - Previous copies of the EARLY modules are deleted from the system the next time that DB2 is started.
- Command is only valid when DB2 is inactive
 - DSNY003I message will be issued if the REFRESH command is issued when DB2 is active.
- No DB2 authority required, but must be executed from a z/OS Console
- Data Sharing – please note command ONLY has Member Scope



SWG BetaWorks

Automatic Creation of Objects

Overview

- **Deprecate simple table space**
 - Can't create them any more
 - Will continue to support the existing simple table spaces
 - Same syntax will create segmented or partition by growth
- **Added**
 - Implicitly created database
 - Implicitly created table spaces
 - Implicitly add indexes for primary key, unique constraints, rowid
 - Implicitly support for LOB table space, auxiliary table and index
- **Why ?**
 - Simplify porting of other DBMS databases / application to z/OS
 - Encourage good database design

Implicit database

- **Pre-DB2 9 CREATE TABLE with no database name, DB2 will use DSNDB04 (assuming user has authority).**
- **In DB2 9 NFM, DB2 creates a new database:**
 - Database named DSN00001 to DSN10000, creator will be SYSIBM
 - DB2 may choose to commit the implicit database creation as a separate task. If so, implicit database creation will not be rolled back if CREATE TABLE fails.
 - When DB2 reaches 10,000 implicitly created databases, DB2 will wrap around and reuse the previously create databases.
 - Implicitly created databases can have multiple table spaces.
 - Requires CREATETAB on DSNDB04 to create a table in an implicit database.
 - Users can not (SQLCODE -2035, reason code 1) explicitly create a table space or table in an implicitly created database.
 - Users can alter any attributes of an implicitly create database.

Implicit database defaults

NAME	DSN00001 – DSN10000
BUFFERPOOL	BP0, BP8K0, BP16K0, BP32K0 (Default values, changeable through ZPARM update).
INDEXBP	IDXBPOOL setting in ZPARM
STOGROUP	SYSDEFLT
Value in column IMPLICIT of SYSIBM.SYSDATABASE	Y
ENCODING_SCHEME	Default: DSNHDECP ENScheme setting
SBCS_CCSID	Default: DSNHDECP SCCSID setting
DBCS_CCSID	Default: DSNHDECP GCCSID setting
MIXED_CCSID	Default: DSNHDECP MCCSIID setting

Implicit Table Space

- In CM will create a SEGSIZE 4, LOCKSIZE ROW
- In NFM:
 - ZPARM IMPTSSEG = SEGMENTED - SEGSIZE 4, LOCKSIZE ROW.
 - ZPARM IMPTSSEG = PARTITION (partition by growth – see Universal TS) -DSSIZE 4G, MAXPARTITIONS 256, LOCKSIZE ROW
 - ZPARM IMPSTDEF = YES or NO - DEFINE YES or NO for the dataset
 - ZPARM IMPTSCMP = YES or NO - COMPRESS YES or NO
 - CCSIDs as per DECP module.

Note: DSSIZE and SEGSIZE can not be ALTERed after CREATE, requires a DROP/CREATE.

- If the table is dropped the implicit table space will be dropped but **not** the database.
- ZPARM MGEXTSZ (sliding scale for secondary allocation) will now default to YES.

Implicit indexes, LOB support and ROWID index

- If the containing table space is **implicitly created**, DB2 will implicitly create the following system-required objects:
 - Primary key index (to enforce the primary key)
 - Unique constraint enforcing index (to enforce the unique key)
 - LOB table space, auxiliary table and auxiliary index
 - ROWID index (where the ROWID column is defined as GENERATED BY DEFAULT)
- Regardless of anything else including SET CURRENT RULES special register
- Furthermore, DB2 will not allow user to drop any of the above implicitly created objects if the containing table space is **implicitly created** (SQLCODE -669 and -2035 rc 2).



SWG BetaWorks

DB2 SMS Constructs

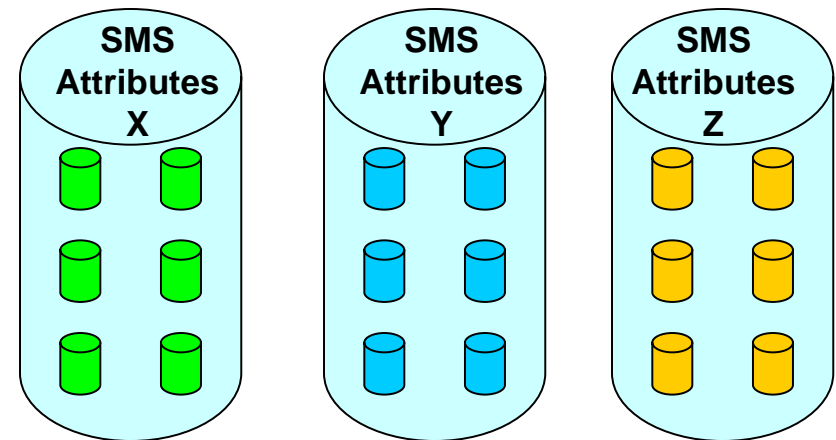
DB2 STOGROUP use of SMS storage classes



The issues today

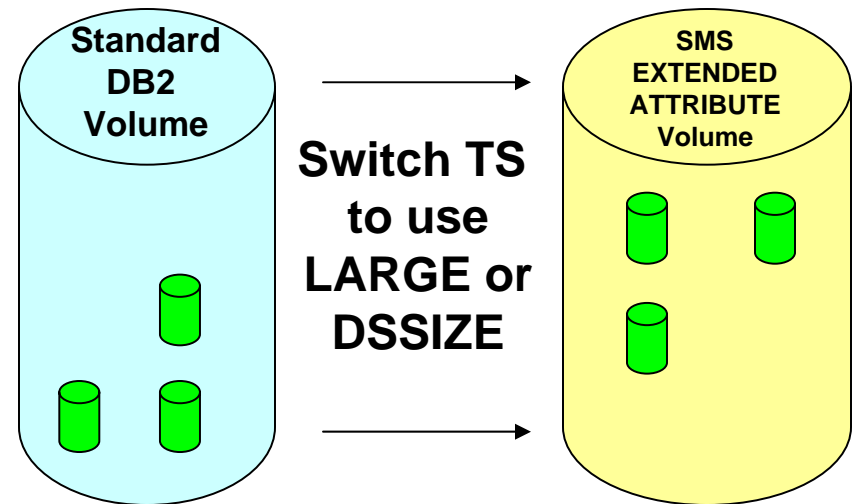
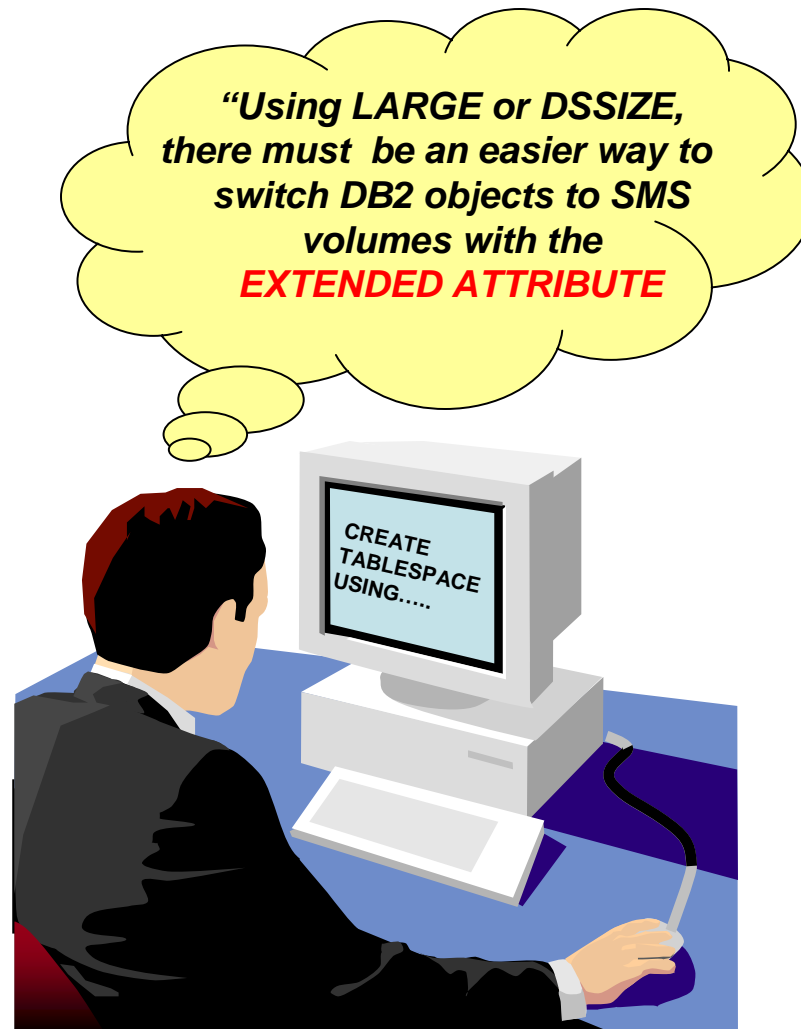
- Parameters can ONLY be used on explicit IDCAMS defines.
- STOGROUP uses IDCAMS under the covers, but my only option is to use ACS routines.
- ACS filters on data set names and I have many data sets!
 - ▶ Too much time spent updating ACS routines.
 - ▶ They are large and error prone.
- I need flexibility and minimised effort.
 - ▶ Freeing up personnel

DB2 STOGROUP use of SMS classes – allocation



- Passing SMS Classes on the CREATE STOGROUP would mean that I could place my DB2 objects on volumes with the appropriate SMS attributes
- Far less work for all

DB2 STOGROUP use of SMS classes – EA volumes



- Customers need the ability to pass the SMS CLASS's when converting objects to ones with the **LARGE / DSSIZE** attribute
- DB2 using more and more EA volumes.
- Space administrator manual intervention reduced by passing SMS parameters in the STOGROUP statement.

DDL CREATE STOGROUP changes

- **CREATE STOGROUP** is enhanced to include SMS data class, management class, and storage class as optional parameters.

CREATE STOGROUP stogroup-name

< **VOLUMES** (volume-id,...)>

Now **OPTIONAL**

VCAT catalog-name

< **DATACLAS** dc-name >

< **MGMTCLAS** mc-name >

< **STORCLAS** sc-name >

} **NEW Parameters**

DDL ALTER STOGROUP changes

- ALTER STOGROUP is enhanced to include SMS data class, management class, and storage class as optional parameters.

ALTER STOGROUP stogroup-name

ADD VOLUMES (volume-id,...)

Now OPTIONAL

REMOVE VOLUMES (volume-id,...)

< DATACLAS dc-name >

< MGMTCLAS mc-name >

< STORCLAS sc-name >

} NEW Parameters

SYSIBM.SYSSTOGROUP changes

Column Name	Data Type	Description	Use
CREORTYPE	CHAR(1) NOT NULL WITH DEFAULT	Type of Creator: Blank - Authorisation ID L - Role	G
DATACLAS	VARCHAR(24) NOT NULL WITH DEFAULT	Name of the SMS data class. Blank if data class not used.	G
MGMTCLAS	VARCHAR(24) NOT NULL WITH DEFAULT	Name of the SMS management class. Blank if management class not used.	G
STORCLAS	VARCHAR(24) NOT NULL WITH DEFAULT	Name of the SMS management class. Blank if management class not used.	G
RELCREATED	CHAR(1) NOT NULL WITH DEFAULT	Release of DB2 that was used to create the object, blank for objects created before DB2 9. For all other values, see "Release dependency indicators".	G

“Thank You for listening”

If you have any questions on this DB2 9 for z/OS session, then please send them to the BetaWorks team at:

Ian_Cook@uk.ibm.com
FLETCHPL@uk.ibm.com

