John Deere participated in the IBM Early Support Program (ESP) for DB2 9 for z/OS. This presentation will introduce you to some of the new features available in DB2 9 that we tested at John Deere. These features include the DB2 9 migration path, the new Universal Tablespaces, Clone Tables, Partitioning by Growth, Index Compression, Not Logged Tables, Roles and Trusted Context and several utility enhancements including new functionality for BACKUP SYSTEM and RESTORE SYSTEM. You will gain a general understanding of these features along with some items of interest that we found while testing. Just for fun, I'll compare some of the DB2 9 features with the equipment we make at John Deere.

Agenda

- Migration
- Universal Tablespace
- Clone Tables
- Partition by Growth
- Index Compression
- Not Logged Tables
- Roles and Trusted Context

1

This presentation will cover the DB2 9 features that were tested at John Deere during the DB2 9 ESP. This presentation will not describe every detail of the features covered but will focus on a general overview along with specific insights that we gained from the ESP participation.

Agenda

- Utilities
  - COPY
  - RECOVER
  - REBUILD INDEX SHRLEVEL CHANGE
  - CATMAINT
  - BACKUP SYSTEM
  - RESTORE SYSTEM

This presentation will cover the DB2 9 features that were tested at John Deere during the DB2 9 ESP. This presentation will not describe every detail of the features covered but will focus on a general overview along with the specific insights that we gained from the ESP participation.

Disclaimer

The material in this presentation is based on our experience at John Deere. This may or may not be indicative of what you will experience in your shop.

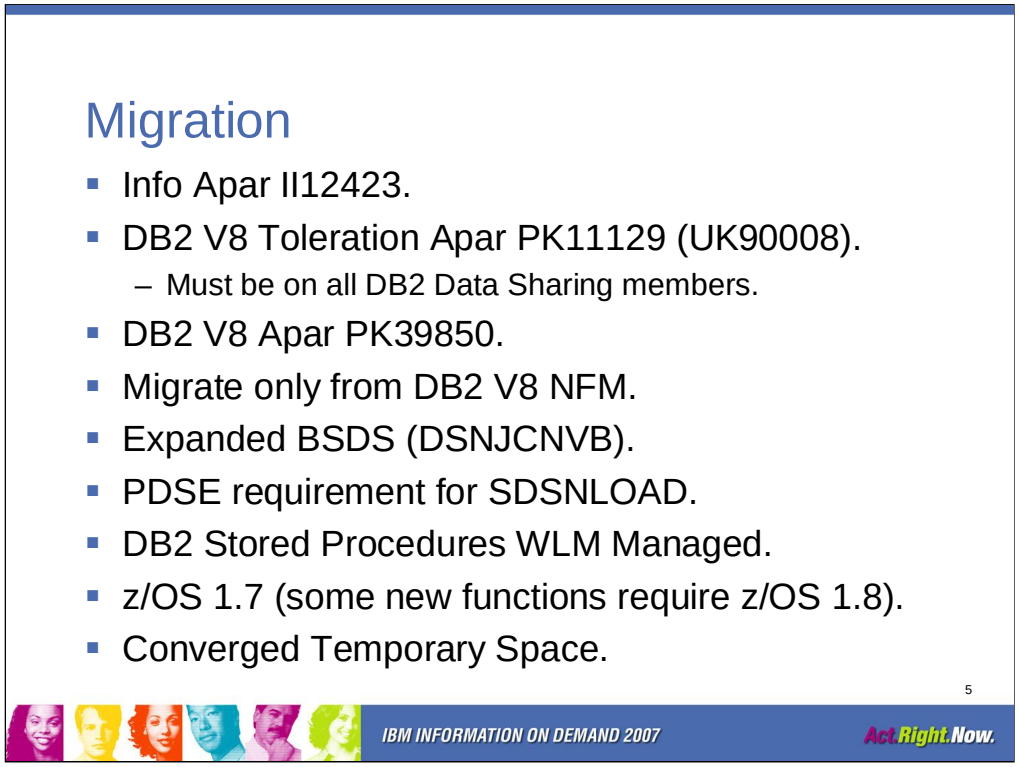
John Deere Equipment and DB2 9 Features





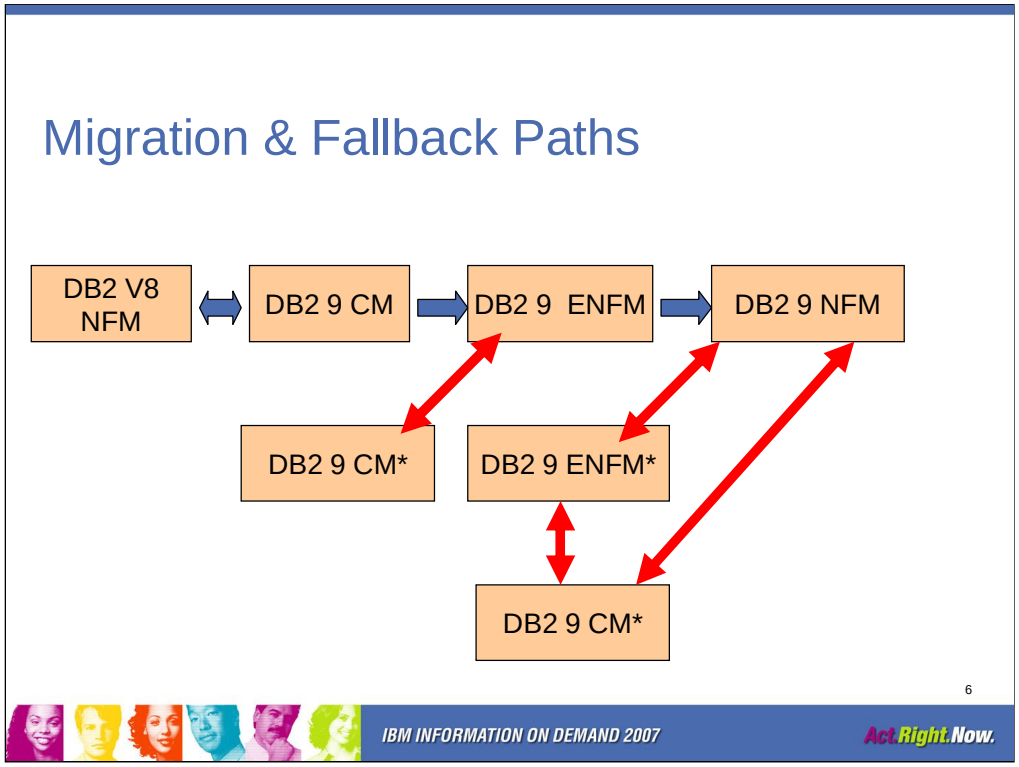

We'll use equipment from my company to summarize the key features that we have learned about DB2 9.

See why I compare DB2 9 to an end loader, a motor grader, a combine, an excavator, a tractor and the Gator.



z/OS 1.8 is required for the volume based utility enhancements and for database roles and trusted contexts.

The temporary database used by declared global temporary tables and static scrollable cursors goes away and these functions use the workfile database, DSNDB07. You may want to increase the size of DSNDB07. Do not drop the temporary database until you are sure that you won't fallback to DB2 Version 8.



Compatibility Mode (CM) is the mode that DB2 is in when DB2 9 is started for the first time from DB2 V8 New Function Mode (NFM). Limited new function of DB2 9 is available in CM. This is the only mode that permits fallback to DB2 Version 8.

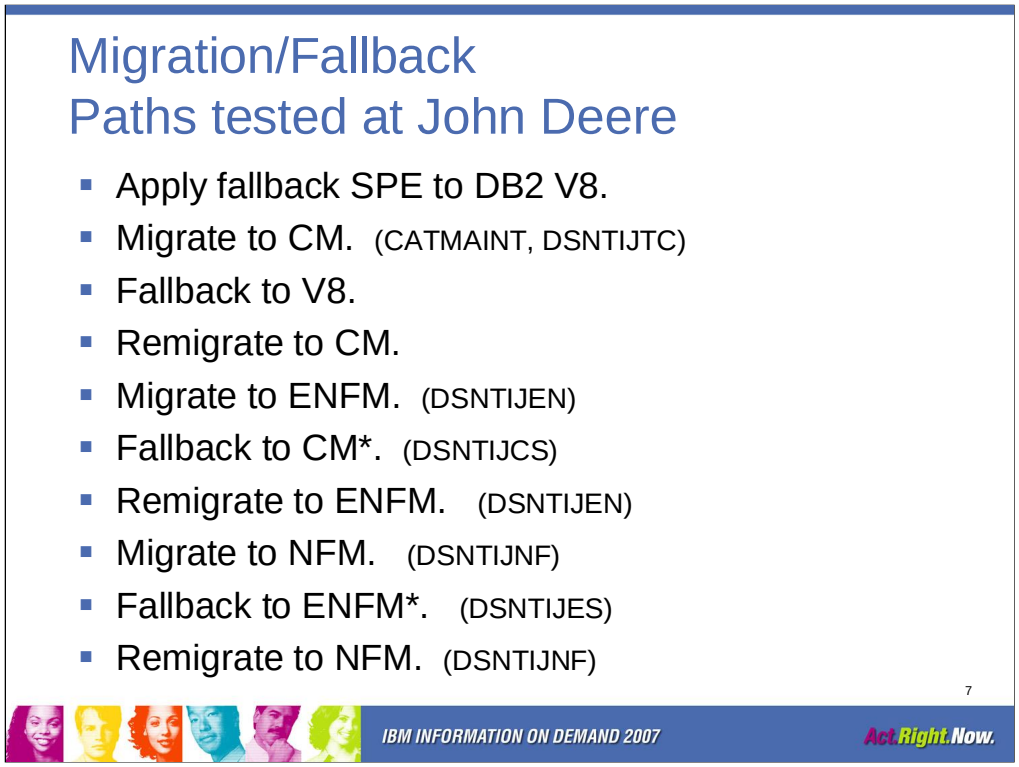
Enabling New Function Mode (ENFM) is the mode that is entered when CATENFM START is executed which is the first step of DSNTIJEN. ENFM can fall back to Compatibility Mode which will be identified as CM*.

New Function Mode (NFM) is the mode that DB2 enters once CATENFM COMPLETE is run. This is the only step of job DSNTIJNF. New functions can be used at this time. Fallback can occur to Enabling New Function Mode which will be identified as ENFM* or to Compatibility Mode which will be identified as CM*.

Enabling New Function Mode Star (ENFM*) is the same as ENFM but indicates that at one time DB2 was in NFM.

Compatibility Mode Star (CM*) is the same as CM but indicates that at one time DB2 was in either ENFM or NFM. You can not fallback to DB2 Version 8 from CM*.

In message DSN7100I the actual DB2 9 migration modes will be identified as C, C*, E, E*, or N.



DSNTIJTC is used to migrate to DB2 9.

DSNTIJEN is used to move to ENFM.

If the real time statistics database, DSNRTSDB, does not exist then steps ENFM1103 and ENFM1103 of job DSNTIJEN will fail. These steps load data from the database to the catalog.

DSNTIJCS is used to fallback to CM* from ENFM, ENFM* or NFM.

DSNTJINF is used to move to NFM.

Migration – Fallback SPE Missing

DSNR045E -DA96 DSNRRPRC DB2 SUBSYSTEM IS STARTING AND IT WAS NOT STARTED IN A PREVIOUS RELEASE WITH THE FALLBACK SPE APPLIED. FALLBACK SPE APAR: PK11129

NEW RELEASE LEVEL: 0000D450

KNOWN LEVEL(S): 0000D3010000D3250000D3500000000000000000

IEF450I DA96MSTR DA96MSTR - ABEND=S04F U0000 REASON=00000000

TIME=09.09.36

This message will be issued when starting DB2 if the DB2 Fallback SPE has not been applied.

The Migration Modes

```
-DIS GROUP DETAIL

DSN7100I  -DA96 DSN7GCMD
*** BEGIN DISPLAY OF GROUP(........) GROUP LEVEL(...) MODE(C*)
                 PROTOCOL LEVEL(3)  GROUP ATTACH NAME(....)
------------------------------------------------------------------
DB2                                    DB2 SYSTEM    IRLM
MEMBER   ID  SUBSYS CMDPREF   STATUS   LVL NAME      SUBSYS IRLMPROC
-------- --- ----   --------  -------- --- --------  ----   --------
........   0 DA96   -DA96     ACTIVE   910 CPU7      IA96   DA96IRLM
------------------------------------------------------------------
*** END DISPLAY OF GROUP(........)
```

MODE is C, C*, E, E* or N

9

The current DB2 9 mode is identified by message DSN7100I.

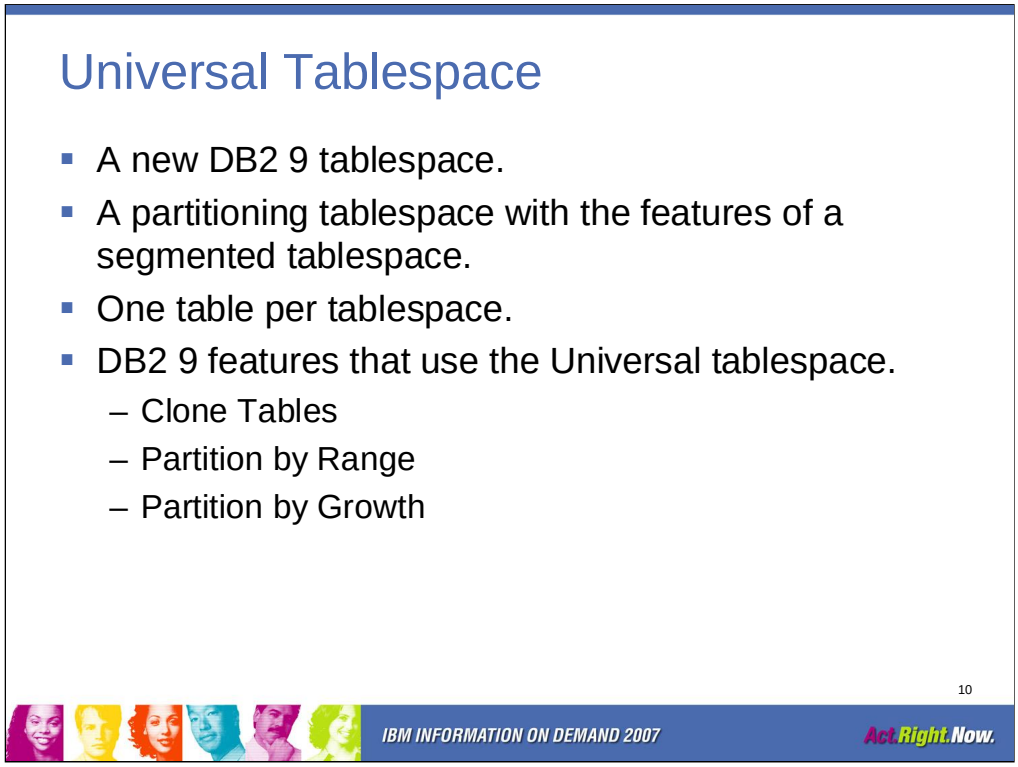
C – Compatibility Mode (CM)

C* - Compatibility Mode Star (CM*)

E- Enabling New Function Mode

E* - Enabling New Function Mode Star (ENFM*)

N – New Function Mode



Simple tablespaces – multiple tables per tablespace, different tables share pages. Simple tablespaces are deprecated in DB2 9. You can use existing ones but can not create new ones.

Segmented – multiple tables per tablespace, different tables don't share pages. Includes a space map page.

Partitioned – one table per tablespace. Needs a partitioning column.

Universal Tablespace – A partitioned segmented tablespace. Contains a space map.

Benefits of Universal Tablespace (UTS) include improved table scan performance, improved mass delete performance and improved ALTER TABLE ROTATE performance.

Clone Tables

- Fast replacement of one table with another.
- IBM's solution to the request for online load replace.
- Application is unaware of the clone table, no application changes are necessary.
- Can not clone a table that has RI.
- Can not make schema changes to a table that has a clone.
- No clone tables on MQTs.
- Can not clone the DB2 Catalog tables.
- Must be a Universal Tablespace, DB2 managed and the only table in the tablespace.

If the tablespace is not a Universal Tablespace receive SQL error:

DSNT408I SQLCODE = -148, ERROR: THE SOURCE TABLE DVDB2SP.USER_GROUP_CONNECT CANNOT BE ALTERED. REASON 05

DSNT418I SQLSTATE = 42809 SQLSTATE RETURN CODE

DSNT415I SQLERRP = DSNXICLN SQL PROCEDURE DETECTING ERROR

DSNT416I SQLERRD = 17 0 0 -1 0 0 SQL DIAGNOSTIC INFORMATION

You can not make schema changes:

ALTER TABLE DVDB2SP.USER_GROUP_CONNEC2
ADD TEST_COL CHAR(1);

DSNT408I SQLCODE = -148, ERROR: THE SOURCE TABLE DVDB2SP.USER_GROUP_CONNEC2 CANNOT BE ALTERED. REASON 11

DSNT418I SQLSTATE = 42809 SQLSTATE RETURN CODE

Additional columns added to DB2 Catalog tables:

SYSTABLESPACE

- Instance
- Clone

SYSCOPY

- Instance

Clone Tables

- Clone table is identical to the base table.
  - Created in the same tablespace.
  - Created with the same indexes.
- Clone table created as empty.
  - Table must be populated.
- Grant access on the clone table.
  - Security separate from base table.
- Utilities process against the clone table.
  - Can not run RUNSTATS on a clone table.
  - Consider running RUNSTATS after EXCHANGE DATA if there is significant differences in the data between the two tables.

12

The clone table is identical to the base table in every way including columns, data types and check constraints. It is even created with the same indexes and before triggers. The clone table can be created with a different schema (owner) than the base table.

Clone Tables

- ALTER TABLE DVDB2SP.USER_GROUP_CONNEC2
  ADD CLONE USER_GROUP_CONNECT_CLN;
- Table is created in the same table space as the base table.
- Base table:
  - DA296.DSNDBC.VRADB002.VRATS901.**I0001**.A001
    DA296.DSNDBD.VRADB002.VRATS901.**I0001**.A001
- Clone table:
  - DA296.DSNDBC.VRADB002.VRATS901.**I0002**.A001
    DA296.DSNDBD.VRADB002.VRATS901.**I0002**.A001

13

Same database, same tablespace, difference only in the instance number.

'ALTER TABLE base-table-name DROP CLONE' will drop the clone table.
'DROP TABLE' will drop both the base table and the clone.

Clone Tables

–DIS DB(VRADB002) SPACE(*)

NAME	TYPE	PART	STATUS	
VRATS901	**TSB1**	0001	RW	(Base Table - I0001.A001 dataset)
VRATS901	**TSC2**	0001	RW	(Clone Table - I0002.A001 dataset)
VRAI901	**IXB1**	0001	RW	(Base index - I0001.A001 dataset)
VRAI901	**IXC2**	0001	RW	(Clone index - I0002.A001 dataset)

14

In this example, the base table and it's associated base index have a dataset suffix of I0001.A001. The clone table and it's associated index have a dataset suffix of I0002.A001.

Clone Tables

```
EXCHANGE DATA BETWEEN TABLE
DVDB2SP.USER_GROUP_CONNEC2 AND
DVDB2SP.USER_GROUP_CONNECT_CLN;

 –DIS DB(VRADB002) SPACE(*)

NAME      TYPE  PART  STATUS
--------  ----  -----  ------
VRATS901  TSB2  0001   RW     (Base table, I0002.A001 dataset)
VRATS901  TSC1  0001   RW     (Clone table, I0001.A001 dataset)
VRAI901   IXB2  0001   RW     (Base index, I0002.A001 dataset)
VRAI901   IXC1  0001   RW     (Clone index, I0001.A001 dataset)
```

No data is moved.

15

The EXCHANGE DATA command will switch the underlying datasets associated with the base table and the clone table. Only the data instances will change, no data will be copied. This is transparent to the application and does not require any application changes.

On EXCHANGE DATA, the clone assumes the base objects' statistics. This allows bound static SQL to function without a rebind.

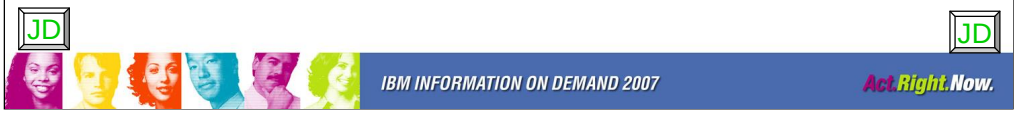
EXCHANGE DATA will put an entry in SYSCOPY with ICTYPE='A' and STYPE='E'.

EXCHANGE DATA requires one of the following:

1) Ownership of both tables.
2) Insert and delete privileges for both tables.
3) DBADM authority on the database.
4) SYSADM authority.

Clone Tables - John Deere application

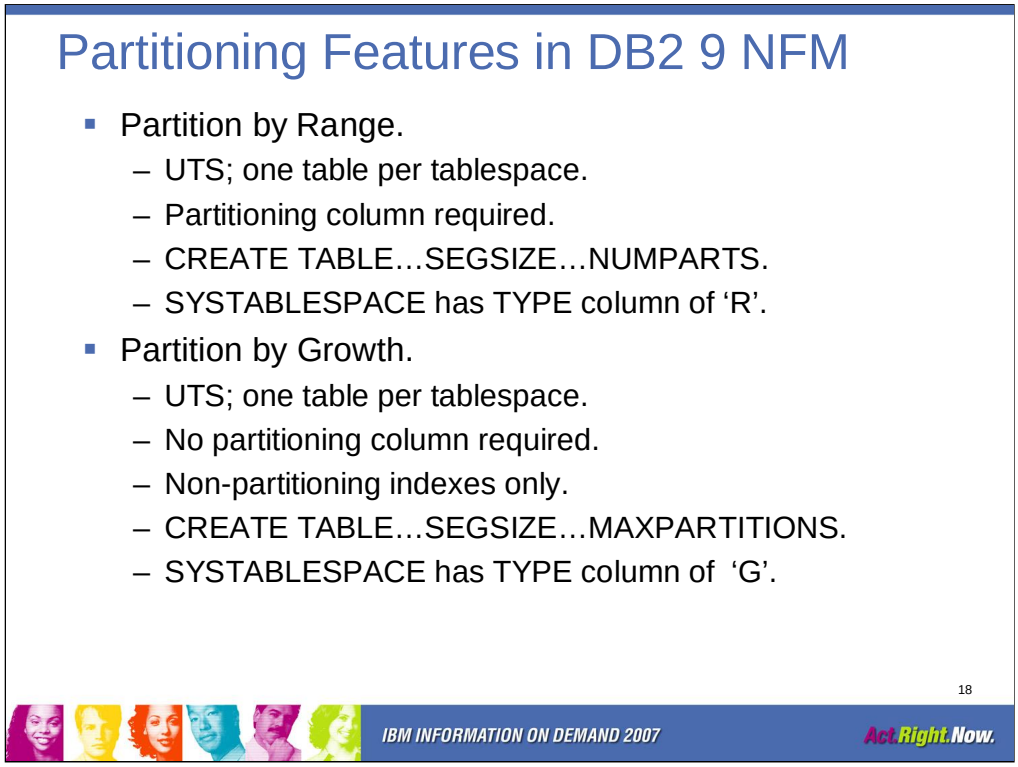
- Nightly refresh of a security application table.
- DB2 LOAD REPLACE originally used.
  - Table unavailable for 15-30 minutes.
- Currently emulating an online load.
  - Copy of the table using REPAIR utilities and IDCAMS renames of underlying datasets.
  - Table unavailalbe for 1 minute.
- Remaining issues:
  - Tablespace must be taken offline.
  - May not stop immediately (STOPP).
- Plan on implementing Clone Tables for this application.



We have a security application table that is refreshed every evening. Originally this was done using the DB2 LOAD utility with the REPLACE option. This resulted in the table being unavailable for 15-30 minutes. We switched to emulating an online load using a copy of the table and executing a series of REPAIR utilities and IDCAMS renames of underlying datasets. This cut the outage time down to one minute. (We learned about this process at an IDUG conference.) Even with this methodology we still had some issues. The tablespace still had to be taken offline. The tablespace may not be stopped immediately (STOPP) which impacts the application. By implementing Clone Tables for this application we will be able to refresh the data nightly without impacting the application.

Clone Tables are like an End Loader

- You can load new data into them without affecting your production table.



Partition by Range still requires a partitioning column but the Universal Tablespace provides advantages of the Partitioned tablespace.

Partition by Range advantages:

Better space management, fewer reorgs.

Improved mass delete.

Improved ALTER TABLE ROTATE performance.

Tablespace Key Words

SEGSIZE	MAXPARTITIONS	NUMPARTS	TABLESPACE TYPE
SEGSIZE	MAXPARTITIONS		PARTITION BY GROWTH (UTS)
	MAXPARTITIONS		PARTITION BY GROWTH (UTS) Segsize 4K
SEGSIZE		NUMPARTS	PARTITION BY RANGE (UTS)
SEGSIZE			SEGMENTED
		NUMPARTS	PARTITIONED
(blank)	(blank)	(blank)	PARTITION BY GROWTH (UTS) – previously Simple TS

19

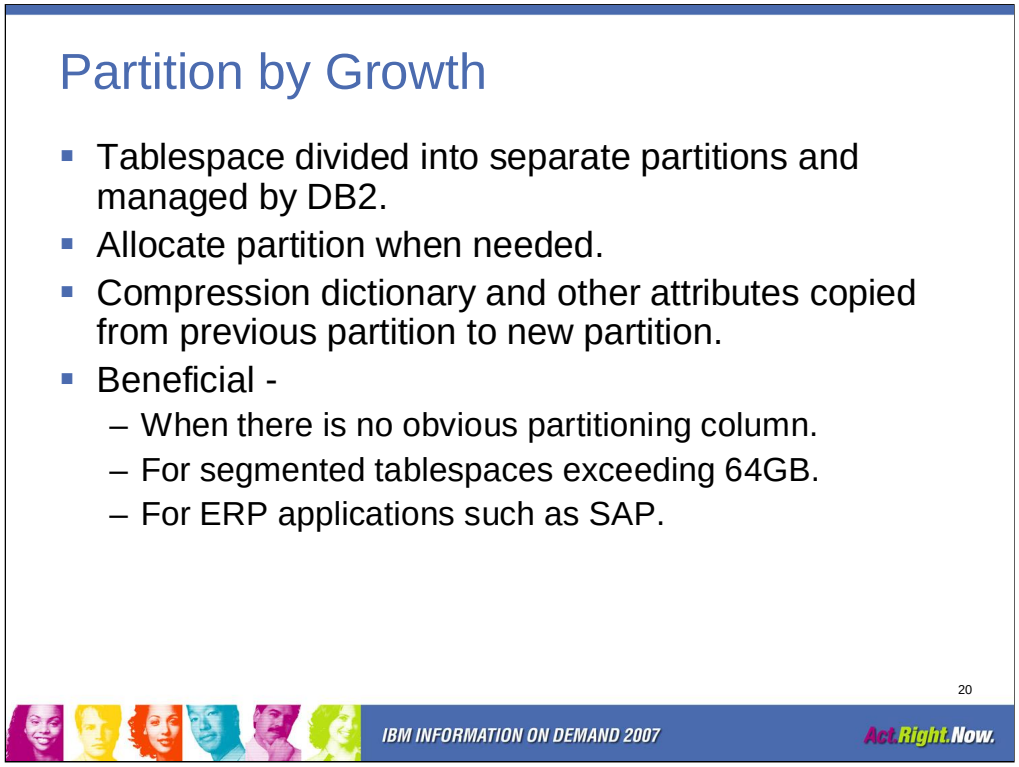
Specifying keywords SEGSIZE and MAXPARTITIONS will create a Universal Tablespace as Partition by Growth. Specifying MAXPARTITIONS without SEGSIZE will create a UTS as Partition by Growth with a default segsize of 4K. No partitioning index is defined.

Specifying keywords SEGSIZE and NUMPARTS will create a Universal Tablespace as Partition by Range. A partitioning index is required.

Specifying only the SEGSIZE keyword creates a segmented tablespace.

Specifying only the NUMPARTS keyword creates a partitioned tablespace.

If none of the keywords are specified the default table space is a Partition by Growth Universal Tablespace.



Partition by Growth partitions according to space needs and not column values.

A Partition by Growth tablespace can be explicitly created with

CREATE TABLESPACE … MAXPARTITIONS n

Or implicitly created with

CREATE TABLE PARTITION BY SIZE EVERY nG

Partition by Growth

- A partition by growth tablespace can grow up to 128 TBs. Maximum size is determined by MAXPARTITIONS, DSSIZE and PAGE SIZE.
- MAXPARTITIONS starts with one partition and then adds other partitions as needed up to the value specified for MAXPARTITIONS.
- REORG does not delete existing partitions even if they are no longer needed.
- Incompatible with ADD PARTITION and ROTATE PARTITION.

21

You can not specify NUMPARTS with MAXPARTITIONS.

Changes to DSSIZE and SEGSIZE require a DROP to change, there is no ALTER option.

The number of partitions will change dynamically.

John Deere Partition by Growth Result

- Load 66 million rows.
- 9 partitions created (4GB/each).
- Compression ratio with load 37-40%.
- Compression ratio after reorg 51%.

22

This is an example of results we saw at John Deere during the DB2 9 ESP for a tablespace defined as Partition by Growth. These results may or may not be indicative of what you will see at your shop based on your data.

John Deere Partition by Growth Result

PART	TSNAME	COMPRESS	PAGESAVE	ROWS
1	**GLPCAKN**	**Y**	**51**	**8165164 ***
2	GLPCAKN	Y	40	7904706
3	GLPCAKN	Y	39	7514008
4	GLPCAKN	Y	38	7488600
5	GLPCAKN	Y	38	7476641
6	GLPCAKN	Y	39	7703738
7	GLPCAKN	Y	38	7317544
8	GLPCAKN	Y	37	7158952
9	GLPCAKN	Y	37	5264234

* Reorg ran on this partition after load.

One of the major benefits of a partition by growth tablespace is that you do not define a partitioning key.

Running reorg after loading the partitions may result in additional compression.

Partition by Growth is like a Motor Grader

- DB2 will spread your data across the partitions.

Index Compression

- Index must be in 8K, 16K, or 32K buffer pool.
  - Indexes in DB2 V8 are 4K BP.
  - SQL code -676 if incorrect page size.
- ALTER INDEX COMPRESS YES.
  - Index placed in RBDP (rebuild pending).
- New column in SYSIBM.SYSINDEXES.
  - COMPRESS … Y/N.
- DSN1COMP utility can be run against the index .
- No compression dictionary for index compression.

25

'ALTER INDEX SAPR3.CKIT10 COMPRESS YES' returns SQLcode -676 if index page size is not 8K, 16K or 32K.

SQLCODE: -676, ERROR: THE PAGE SIZE OF THE BUFFERPOOL FOR THE INDEX

IS NOT VALID FOR THE SPECIFIED INDEX

Index Compression

- Convert 4K index to 8K, 16K, or 32K.
- Turn compression on.
- Process:
  - STOP INDEX
  - ALTER INDEX …BP8K
  - ALTER INDEX…COMPRESS YES
  - REBUILD INDEX

(Sample jcl is in the notes section of the presentation.)

26

Altering the Index to a different buffer pool puts the index in Rebuild Pending (RBDP) status. Altering the index to turn on compression also puts the Index in RBDP. However both alter statements can be executed in the same job as shown below:

```
//DB2CMD   EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSTSIN  DD *
  DSN SYSTEM(X96A)
  -STOP DB(A000XAAL) SP(TECSH0)     ← BP change requires the IX Space to be stopped
  RUN PROGRAM(DSNTEP2) PLAN(DSNTEP2) LIB('DB2X96.X96A.RUNLIB.LOAD')
  -STA DB(A000XAAL) SP(TECSH0)              ← Restart IX Space
  END
//SYSIN    DD *
  ALTER INDEX SAPR3."TECS~0" BUFFERPOOL BP16K1;   ← Move to 16K BP
  ALTER INDEX SAPR3."TECS~0" COMPRESS YES;          ← Turn compression on
//*
```

Index Compression - DSN1COMP

8 K Page Buffer Size yields a
- 51 % Reduction in Index Leaf Page Space
  The Resulting Index would have approximately
- 49 % of the original index's Leaf Page Space
  No Bufferpool Space would be unused

16 K Page Buffer Size yields a
- 71 % Reduction in Index Leaf Page Space
  The Resulting Index would have approximately
- 29 % of the original index's Leaf Page Space
- 13 % of Bufferpool Space would be unused to ensure keys fit into compressed buffers

Use DSN1COMP before you decide on an index page size. There may be a trade off between disk space savings and wasted buffer pool buffers.

John Deere Index Compression Result

- Before compression:
  - …CKITG0.I0001.A001 2 GB
  - …CKITG0.I0001.A002 2 GB
  - …CKITG0.I0001.A003 2 GB
  - …CKITG0.I0001.A004 0.56 GB
- 8K BP with compression:
  - …CKITG0.I0001.A001 2 GB
  - …CKITG0.I0001.A002 1.42 GB
- 16K BP with compression:
  - …CKITG0.I0001.A001 1.98 GB

(Index was not padded.)

28

This is an example of results we saw at John Deere during the DB2 9 ESP. These results may or may not be indicative of what you will see at your shop based on your data.

Before compression four instances of the tablespace dataset were needed (A001 through A004).

With 8K buffer pool and index compression only two instances were needed and with 16K buffer pool and index compression all the data fit into a single dataset.

Index Compression Performance

Access	Number of Rows	CPU Before Compression	CPU After Compression
Select Count, Index Only (Unique index)	88 million	1min 19.52sec	1min 16.09sec
Select Count, Index Only (Non-unique IX)	88 million	45.01 sec	45.11 sec
Random access, Index only (1 index)	76 million	79min 57.38 sec	81min 15.61 sec
Random Inserts (5 indexes)	3011	21.06 sec	20.08 sec
Sequential Inserts (5 indexes)	3011	16.40 sec	17.35 sec

Index is 8K page. CPU time is Class 2.

This is an example from our testing of index compression. In this example there was minimal impact to CPU time with index compression.

This is an example of results we saw at John Deere during the DB2 9 ESP. These results may or may not be indicative of what you will see at your shop based on your data. Index page size used 8K. The CPU time is Class 2.

Index Compression is like a Combine

- You will be able to harvest dasd savings.

Not Logged Tablespace

- DB2 9 New Function Mode only.
- Default is logged.
- DBADM authority required.
- Undo/Redo log records are not maintained.
- Indexes inherit the logging attribute from the base tablespace.
- You need to think this one through. Don't use it just because you can.

Not logged has a very small gain in cpu time. Elapsed time improvements are small other than in cases where the log has become a bottleneck, such as in cases where 20 concurrent batch insert jobs are inserting millions of rows each. With other options, such as fast disks, log striping and DB2 9 log improvements, you can log very quickly. The small gain may cost more in recoverability than it gains.

Not Logged Tablespace

- Possible uses:
  - D/W for loading summarized tables (MQTs).
  - Loading data into a table and making changes to the data with SQL before using the table.
  - Changing a lot of data in a controlled process such as for year-end processing.
- If anything goes wrong during the ‘not logged period’ then you need to recover from the previous image copy or reload the data.
- Do not sacrifice data recoverability for performance. In most cases impact on performance is not noticeable for not logged tablespaces.

For a single task inserting it is very difficult to see significant differences in performance. If you are running 10 or 20 concurrent tasks then the log could be a bottleneck. Scalability for logging is improved in DB2 9.

Not Logged Tablespace

- CREATE TABLESPACE…NOT LOGGED
- ALTER TABLESPACE…NOT LOGGED
- SYSIBM.SYSCOPY
  - ICTYPE = A
  - STYPE = O, logging attribute altered to NOT LOGGED
  - STYPE = L, logging attribute altered to LOGGED
- SYSIBM.SYSTABLESPACE
  - LOG = YES, logging attribute is Logged
  - LOG = NO, logging attribute is Not Logged
- Consider auditing SYSTABLESPACE.

33

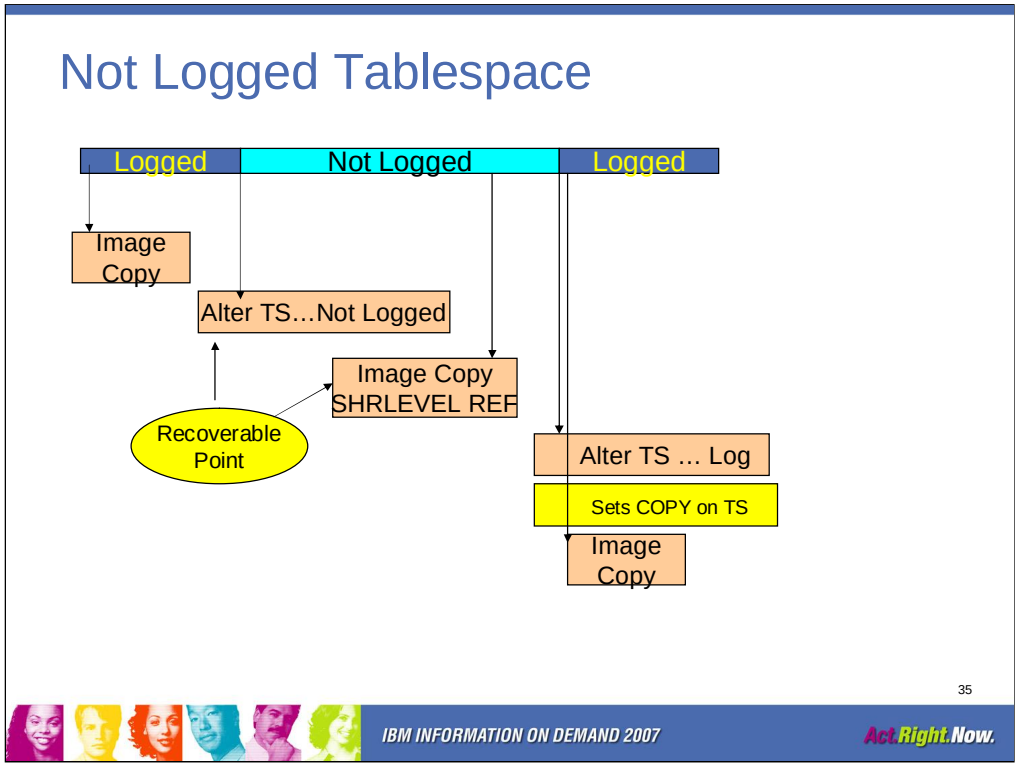
DB2ADM authority can turn logging off.

Consider auditing SYSTABLESPACE for not logged tables using the following SQL:

```
SELECT NAME, DBNAME, LOG FROM SYSIBM.SYSTABLESPACE
WHERE LOG = 'N';
```

Not Logged Tablespace

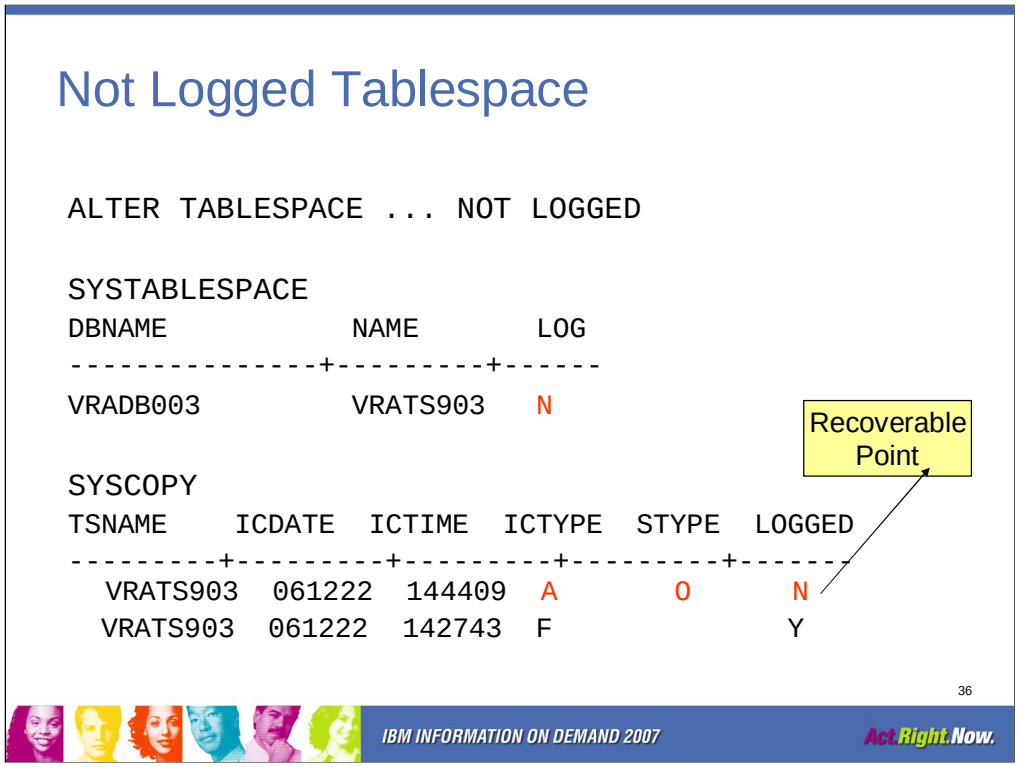
- Recoverable Points for a Not Logged Tablespace.
  - To the point of ALTER TABLESPACE NOT LOGGED.
  - Image Copy SHRLEVEL REFERENCE.
- LPL/RECP Status:
  - Deadlock
  - Timeout
  - Any abend
- Automatic LPL recovery is not initiated for not logged tablespace.
- -START DATABASE has no effect on the LPL status of a not logged tablespace.



After turning logging back on the tablespace will be in COPY pending status.

Avoid ROLLBACK in applications as this will put the not logged tablespace in LPL status and RECP (recover pending).

Recoverable points for a not logged tablespace are either the point at which ALTER TABLESPACE NOT LOGGED occurred or an Image Copy SHRLEVEL REFERENCE.



ALTER TABLESPACE NOT LOGGED places an entry into SYSCOPY and switches the LOG column of SYSTABLESPACE.

When logging is turned back on for a tablespace, SYSCOPY will be updated with ICTYPE=A, STYPE=L, LOGGED=Y.

Not Logged Tablespace

The first update to the tablespace after setting the 'Not Logged' attribute places the tablespace in ICOPY status.

```
VRATS903 TS    RW,ICOPY
```

Rollback or timeout results in the tablespace being placed in RECP status.

```
VRATS903 TS   RW,RECP,LPL,ICOPY
```

Subsequent access to the table will result in SQLCODE -904 and the table will need to be recovered to a 'recoverable point'.

Not Logged Tablespaces are like an Excavator

- You can dig yourself a really big hole.

Database Roles

- Groups together one or more privileges that can be assigned to users.
- A virtual authid.
- User assigned only to one role.
- Roles can own objects.
- Establish through a trusted context.

39

The following DB2 catalog tables contain information about roles and trusted contexts:

SYSROLES, SYSCONTEXT, SYSOBJROLEDEP, SYSCTXTTRUSTATTRS, SYSCONTEXTAUTHIDS

Trusted Context

- Establishes a trusted relationship between DB2 and an external entity.
- A database entity defined based on system authorization ID and a connection's trusted attributes.
- Database roles and trusted contexts enhance the DB2 for z/OS security model.

The following DB2 catalog tables contain information about roles and trusted contexts:

SYSROLES, SYSCONTEXT, SYSOBJROLEDEP, SYSCTXTTRUSTATTRS, SYSCONTEXTAUTHIDS

The relationship between a connection and a trusted context is established when a connection to the server is first created.

Roles and Trusted Context

- The issue at John Deere is to replace a shared userid that has SYSADM authority with individual userids that have SYSADM roles.
- Create the database role:

```
CREATE ROLE SAPADM_ROLE;
GRANT SYSADM to ROLE SAPADM_ROLE;
```

41

In this example a new role called SAPADM_ROLE is being defined. The SAPADM_ROLE is being assigned SYSADM authority on this DB2 subsystem.

Audit needs to be more detailed now for individual accountability. Roles help.

Roles and Trusted Context

- Create the trusted context:

```
CREATE TRUSTED CONTEXT BP51498
BASED UPON CONNECTION USING
SYSTEM AUTHID BP51498
ATTRIBUTES (JOBNAME 'X96SELCT',
   JOBNAME 'BP51498')
ENABLE
DEFAULT ROLE SAPADM_ROLE
WITH ROLE AS OBJECT OWNER;
```


A trusted context can only be created if the role exists otherwise the create command fails with a SQL error -204.

You can use the ALTER command to add attributes to a trusted context.

In this example a trusted context is being defined for the BP51498 primary authid. The connection attributes are for a batch job called X96SELCT and for a tso userid BP51498 that will be used for SPUFI. The default role for BP51498 is SAPADM_ROLE and the role can be the object owner for DB2 objects.

Roles and Trusted Context

- If jobname or tso user id is not an attribute on the trusted context, you will get SQL error -551.
- Original design did not allow for wild carding of jobnames in the trusted context.
- SAPADM_ROLE will be the 'grantor' when granting authorities.
  - Audit trace Class 2 will show detailed information for grant and revoke.
- SAPADM_ROLE will own the created objects.

There are many other possible uses for database roles and trusted contexts, such as for an application server. This was the only scenario that John Deere tested as part of the DB2 9 ESP.

Roles and Trusted Context

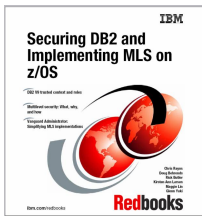
- CREATE TABLE DB2ADM.USER_TABLE;
  Using trusted context: BP51498 …
  DEFAULT ROLE SAPADM_ROLE
  WITH ROLE AS OBJECT OWNER

- New OWNER column in SYSTABLES and SYSINDEXES
- Owner = SAPADM_ROLE
- Creator = DB2ADM
- Createdby = BP51498

DB2 9 has introduced a new column, OWNER into the catalog tables (SYSIBM.SYSTABLES, SYSIBM.SYSINDEXES, etc.). This column along with column CREATEDBY are for documentation only. The CREATOR column is the column that is used as the high level qualifier when an unqualified object is accessed.

Roles and Trusted Context

- Role cannot be dropped if a trusted context depends on it. (SQL -478)
- Role can not be dropped if DB2 objects exist that are owned by the role. (SQL -478)
- Revoking a user's access to a role does not cause cascade delete.
- Redbook SG24-6480, 'Securing DB2 and Implementing MLS on z/OS'



Trusted Contexts are like a Tractor

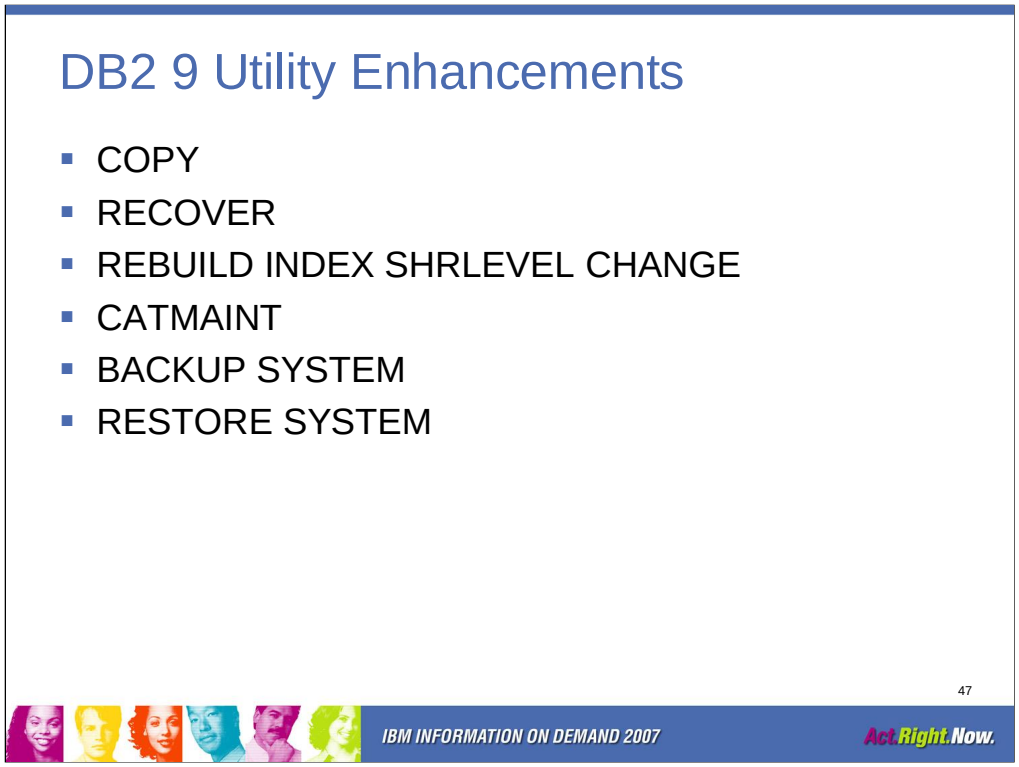
- Versatile, you are able to pull a lot of different equipment with them.




46

Able to pull a lot of different equipment and tools.

Roles and Trusted Contexts have much more depth and capabilities than just the one example that I have demonstrated here.



CPU reductions are very substantial, with some customers experiencing 20% to 30% savings. More online utilities are coming! Online rebuild build an index in place after setting RBDP. Dynamic SQL will avoid the index until available. This works well for non-unique indexes. Online reorg solves the outage during the BUILD2 phase when reorging a partition at a time. Online Check Data and Check Lob are the same as existing utilities, only with SHRLEVEL CHANGE. Check has the SHRLEVEL CHANGE technique extended to SHRLEVEL REFERENCE.

XML support in utilities is very extensive. Cloned tables are somewhat like Online Load Replace. There is the live table, and then there is the shadow or clone. One can load the clone, and then when ready for the clone to become live, a “switch” can be trigger with DDL.

Volume-based utilities allow recovery of individual objects after a volume based backup. It also extends system based backups and restores to and from tape. Modify Recovery is extended to allow n number of backups vs an age. Histogram stats uses quantiles to give the optimizer more statistics.

COPY using Template Switching

- Provides the ability to direct DB2 image copy datasets to either disk or tape based on size.
- At John Deere, it will replace an internal usermod that puts ‘large’ image copies to tape instead of disk.

COPY using Template Switching

```
TEMPLATE DB2CPYDK  UNIT DISK
  DSN ('DXX.D945.@&SSID.ALL.&DB..&TS.(+1)')
  DISP(NEW,CATLG,CATLG)
  GDGLIMIT(1)
  LIMIT(20 CYL,DB2CPYTP)

TEMPLATE DB2CPYTP UNIT TAPE
  DSN ('DXX.D945.@&SSID.ALL.&DB..&TS.(+1)')
  DISP(NEW,CATLG,CATLG)
  GDGLIMIT(1).
```

49

In template DB2CPYDK, the LIMIT(20 CYL,DB2CPYTP) parameter instructs the DB2 COPY utility to image copy a tablespace with a size up to and including 20 cylinders to disk. A table space with a size over 20 cylinders will use template DB2CPYTP, which will put the image copy dataset to tape.

CALCULATION:

Using DA296 tablespace VRADB002.VRATS901 as tablespace to backup. The corresponding tablespace dataset occupies 49 cylinders per a listcat of the dataset:

VOLUME

VOLSER----------SH9D50 PHYREC-SIZE---------4096

ALLOCATION

SPACE-TYPE------CYLINDER HI-A-RBA--------36126720

SPACE-PRI----------49 HI-U-RBA---------2457600

SPACE-SEC----------5

Per IBM doc:

Cyl: HFPunit / 184

MB: HFPunit / 256

GB: HFPunit / 262144

Convert the H-U-RBA into 4k pages: 2457600 / 4096 = 600.

600 / 184 = 3.2 Cyl - rounded down to 3 CYL

600 / 256 = 2.3 MB – rounded down to 2 MB

Note *** all calculations are truncated not rounded. So, for example a value of 3.8 cylinders would truncate to 3 cylinders.

COPY using Template Switching

```
//DA96UTIL.SYSTEMPL DD
   DSN=DXX.D945.DB2.TEMPLATE.CNTL(D945V9),
    DISP=SHR
//DA96UTIL.SYSIN DD *
COPY TABLESPACE DSNDB06.SYSDBASE
   COPYDDN(DB2CPYDK)
   SHRLEVEL REFERENCE
```

JCL will use the DB2CPYDK template specified in COPYDDN to copy the tablespace to disk. Based on the template, if the image copy exceeds the limit in DB2CPYDK then the image copy will use the DB2CPYTP template and be put to tape.

Template Switching is like a Gator

- A utility vehicle.

RECOVER…RESTOREBEFORE

- Used to bypass an image copy as the recovery base. (Available in CM.)

SYSCOPY:

```
DBNAME    TSNAME    ICDATE   START_RBA
---------+---------+---------+---------+
VRADB003  VRATS903  070104   00008090E9C5
VRADB003  VRATS903  070104   0000808E4AF7
```

RECOVER TABLESPACE VRADB003.VRATS903
RESTOREBEFORE X'00008090E9C5'

52

Allows you to recover an object but avoid using a specific image copy as the base.

This is an alternative to restoring to an older image copy using DSN1COPY and then running RECOVER LOGONLY.

Example SYSOUT:

```
THE IMAGE COPY DATA SET
DXX.D945.@DA96ALL.VRADB003.VRATS903.G0006V00
                    RECOVER UTILITY LOG APPLY RANGE IS
RBA 0000808F21FA LRSN 0000808F21FA TO
              RBA 0000808F4F63 LRSN 0000808F4F63
RECOVER UTILITY LOG APPLY RANGE IS RBA 000080918878 LRSN
000080918878 TO
              RBA 000080BFE62E LRSN 000080BFE62E
```

RECOVER…TORBA/TOLOGPOINT with data consistency

- RECOVER utility with TORBA or TOLOGPOINT is to a consistent point.
  - Uncommitted transactions are rolled back.
  - Don't need to use the QUIESCE utility.

SYSCOPY:

DBNAME	TSNAME	ICDATE	ICTIME	ICTYPE	STYPE	SHRLEVEL
VRADB003	**VRATS903**	**070105**	**141616**	**P**	**C**	
VRADB003	VRATS903	070105	140909	F		C
VRADB003	VRATS903	070105	134340	P		

53

RECOVER TORBA or TOLOGPOINT in DB2 9 will be to a consistent point.

RECOVER TOCOPY, TOLASTCOPY and TOLASTFULLCOPY using SHRLEVEL CHANGE will continue to work as they did in DB2 Version 8.

REBUILD INDEX SHRLEVEL CHANGE

- Online index rebuild.
- Allows DML concurrency on a table with REBUILD utility executing against indexes.
- Index being rebuilt is a non-unique index.
  - Unique index can result in -904's.
- Useful in a SAP environment for adding indexes.

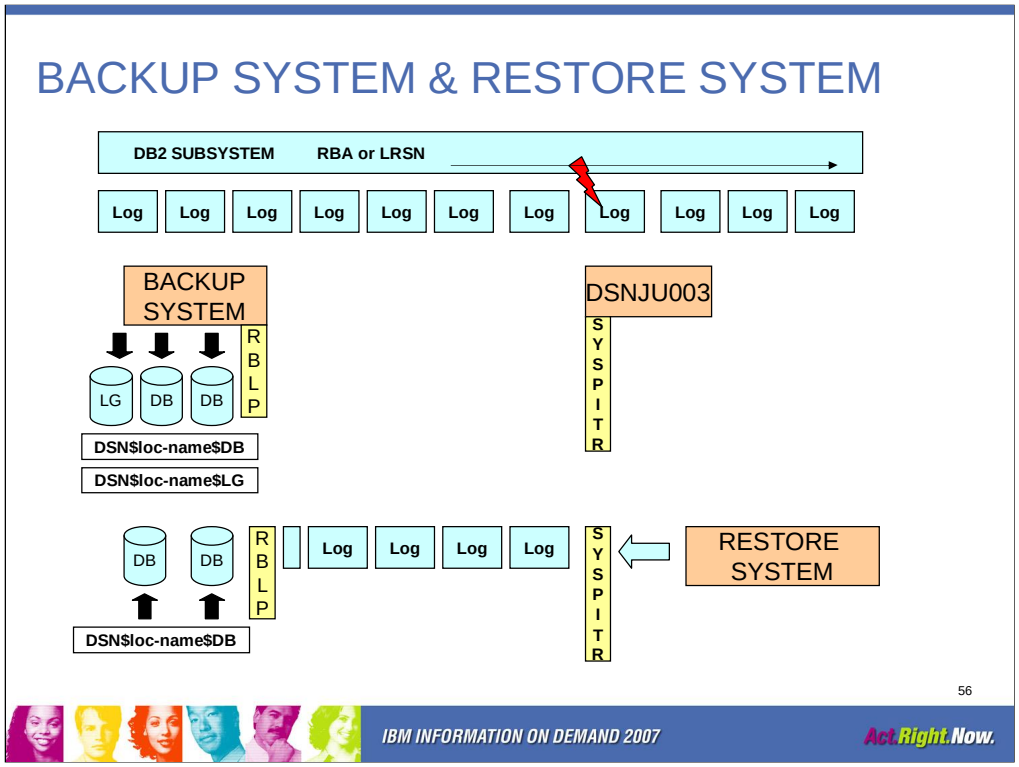
Restriction:
SHRLEVEL CHANGE is not well suited for unique indexes and concurrent DML because the index is place in RBDP while being built. Inserts and updates of the index will fail with a resource unavailable (-904) because uniqueness checking cannot be done while the index is in RBDP.
SHRLEVEL CHANGE is not allowed for not logged tables, XML indexes, or spatial indexes.

CATMAINT – UPDATE VCAT

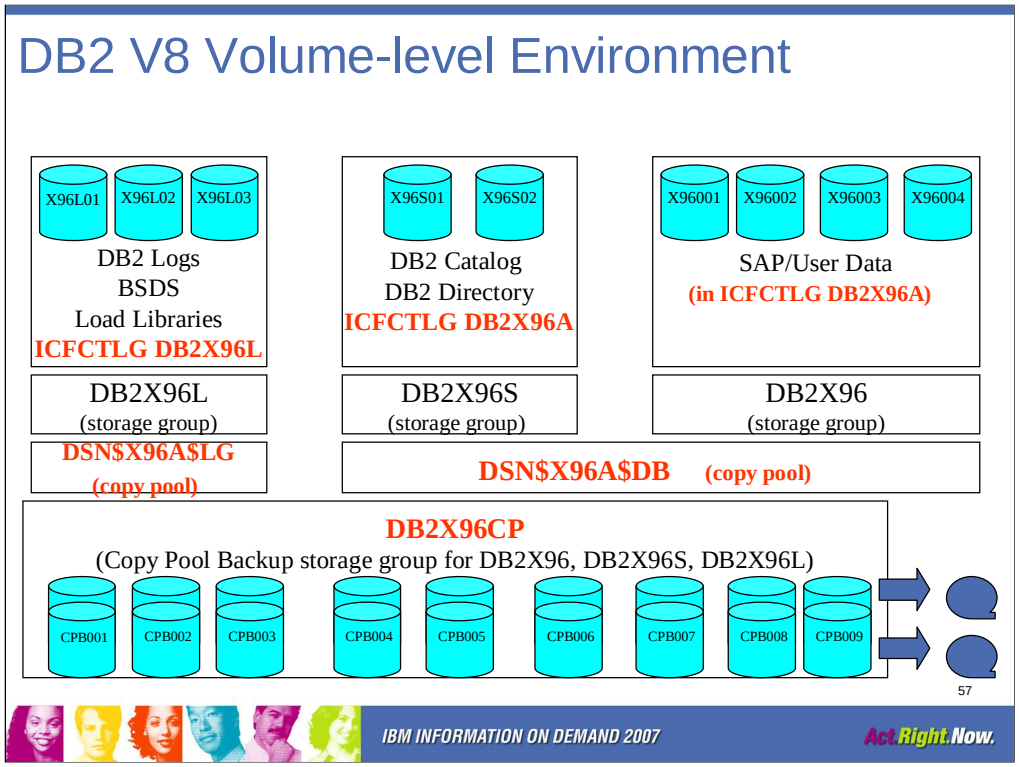
- VCAT name can be switched to a different name using a new option of the CATMAINT utility.
- CATMAINT UPDATE VCAT SWITCH(DB2X96,DB2X92)
- VCAT names for all table spaces, indexes and stogroups were switched in 9 minutes.
  - SAP system with 60,000 objects.
- Useful for cloning SAP systems.
  - We will replace an internally developed process that took over two hours to execute.

CATMAINT should be run under Install Sysadm user and DB2 should be in maintenance mode.

CATMAINT can also be used for updating SCHEMA and changing OWNER to a database role.



The BACKUP SYSTEM utility will dump the database and the log copypools and establish the RBLP (Recovery Based Log Point). DSNJU003 is used to specify the log truncation point, SYSPITR, in order to perform the system-level point-in-time recovery. The RESTORE SYSTEM utility will restore the database copypool only and then apply the DB2 logs from the RBLP up to the SYSPITR.

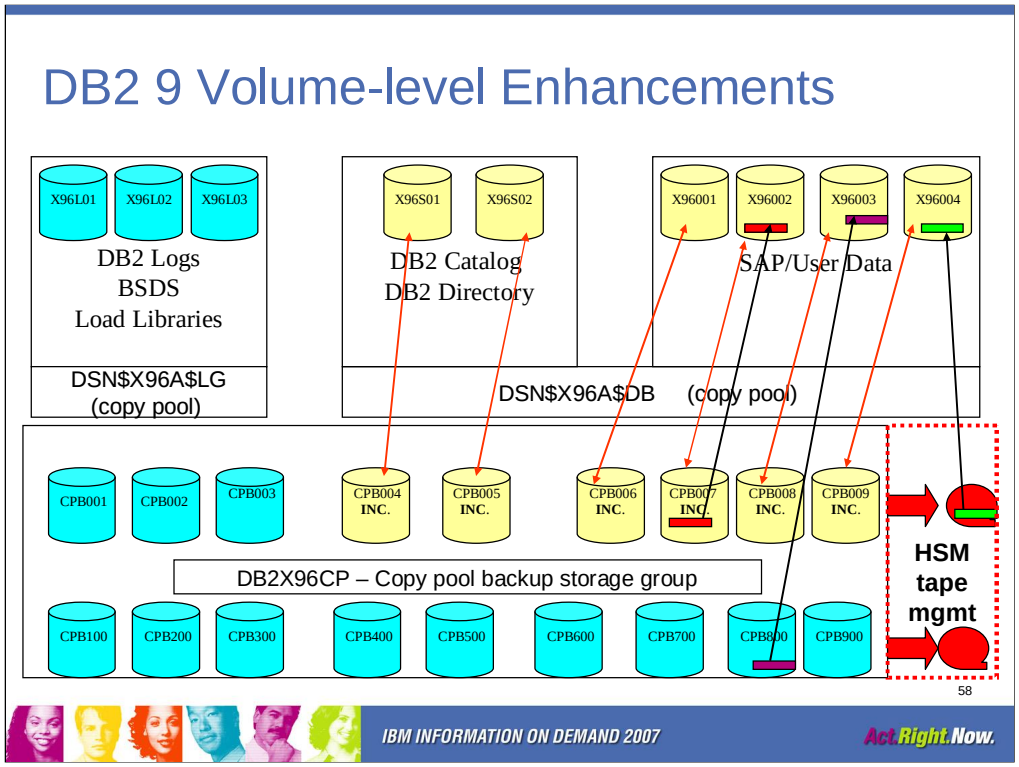


DB2 V8, z/OS 1.5, FlashCopy enabled dasd and DFSMS management of DB2 Catalog, Directory and data are required for using BACKUP SYSTEM utility.

This slide pictorially depicts the John Deere test environment that was used in examples through out this presentation. The DB2 subsystem is known as X96A. Physical volumes and their associated datasets have been put into three separate SMS managed storage groups. The SMS managed storage groups are DB2X96, DB2X96L and DB2X96S. Two separate ICFCTLGs have been created. ICFCTLG DB2X96L for datasets in the DB2X96L SMS storage group and ICFCTLG DB2X96A for datasets in both the DB2X96S and DB2X96 SMS storage groups. Two new SMS constructs are identified. DSN$X96A$LG associated with SMS storage group DB2X96L and DSN$X96A$DB associated with SMS storage groups DB2X96S and DB2X96. DB2 will use these two new constructs to invoke FRS backup and restore. DB2X96CP is a new SMS Copy Pool Backup storage group that contains the target volumes that will be used for backing up the source volumes.

For a DB2 data sharing environment, the additional DB2 members would have their logs and BSDS defined in the DB2X96L storage group.

In DB2 Version 8, offloading the volume dumps in the copy pool backup storage group to tape is a user managed process.



DB2 9 and z/OS 1.8 will enhance the volume-level backup and recovery process to include DFSMShsm tape management of the FlashCopy volume offloads, dataset level recovery from the volume-level backup and support of incremental FlashCopy. The incremental FlashCopy support is available through DFSMShsm z/OS apar OA17314. Incremental FlashCopy can be turned on outside of DB2. Current plans are for DB2 9, after GA, to provide keywords for establishing and terminating the incremental FlashCopy relationship.

The volume-level backup can not be used for dataset level recovery if the dataset has been moved from the original source volume after the FlashCopy. Online Reorg does take an inline image copy. The inline image copy will be used as the recovery base instead of the volume-level backup for those objects that have been relocated through an online reorg.

DFSMShsm Dataset Level Recovery

- DB2 RECOVER utility enhanced to use volume-level backups as the recovery base for individual objects in the system.
- Recovery can be from the FlashCopy disk volume or from the tape offload.
- New Zparm panel to set this option (DSNTIP6).

DFSMShsm Tape Support
BACKUP SYSTEM

- DB2 9 and z/OS 1.8 enhancement.
- Manage the dump/restoration of fast replication copies to and from tape.
- DUMPCLASS
  - New parm on CopyPool definition.
  - Defines attributes for the dump copy.
    - Name, frequency, expiration, etc.
- New keywords for BACKUP SYSTEM utility.
- Once the dump copy has started, DFSMShsm prevents the target volumes from being overlaid by a new FlashCopy until all volumes have been dumped.

60

Up to five DUMPCLASSes can be specified on the CopyPool definition.

BACKUP SYSTEM DUMP – initiate dump processing to tape when the fast replication of the copy pool is logically complete.

BACKUP SYSTEM DUMPONLY TOKEN X' …' - restarts the tape dump processing for a system level backup that has failed.

BACKUP SYSTEM FORCE – allows a system level backup whose required tape dumps have not completed to be overwritten with a new system level backup.

DFSMShsm Tape Support
RESTORE SYSTEM

- DB2 9 and z/OS 1.8 enhancement.
- Allows restore of the copypool from a system level backup that has been dumped to tape.
- Restores the database copy pool from tapes in parallel.
- Chooses restore from dasd over tape if database copypool backup is still on FlashCopy target volumes.
- Options set in new Zparm panel DSNTIP6.

Incremental FlashCopy Support

- DB2 9 and z/OS 1.8 enhancement.
- Initial copy copies every track.
- Subsequent copies only copy changed tracks.
- Creates a persistent relationship.
- Only one persistent relationship per source volume.
  - If a Copy Pool has more than one version defined, only one of the versions can be incremental.
- The dump to tape is the entire dasd volume, not just the changed tracks.
- New keywords for BACKUP SYSTEM utility.
  - ESTABLISH FCINCREMENTAL
  - END FCINCREMENTAL
  - Apars PK41001 and PK42014.

62

The new BACKUP SYSTEM keywords, ESTABLISH FCINCREMENTAL and END FCINCREMENTAL, are not part of the DB2 9 GA code. IBM is anticipating delivering this functionality with an apar after DB2 9 GA.

Incremental FlashCopy Support

- CPUZ 14:50:13 COPYPOOL=DSN$DSNX96$LG ,VER=001,VTOCENQ=Y,MADE
- ON 2006/08/22 AT 13:46:29,FRS=RECOVERABLE,DPS=NONE
- CPUZ 14:50:13 TKN(C)=C'INCR FIRST'
- CPUZ 14:50:13 TKN(H)=X'C9D5C3D940C6C9D9E2E3'
- CPUZ 14:50:13 TOTAL NUM OF VOLUMES=00004,INCREMENTAL=Y
- CPUZ 14:50:13 SGNAME=DB2X96L SOURCE=X96L01 TARGET=FC6408
- CPUZ 14:50:13 SGNAME=DB2X96L SOURCE=X96L02 TARGET=FC6409
- CPUZ 14:50:13 SGNAME=DB2X96L SOURCE=X96L03 TARGET=FC641A
- CPUZ 14:50:13 SGNAME=DB2X96L SOURCE=X96L04 TARGET=FC641B

Example of output showing that an incremental FlashCopy relationship exists for Version 1 of the copypool.

Incremental FlashCopy Support

- CPUZ 14:50:13 COPYPOOL=DSN$DSNX96$LG ,VER=002,VTOCENQ=Y,MADE
- ON 2006/08/22 AT 13:50:14,FRS=RECOVERABLE,DPS=NONE
- CPUZ 14:50:13 TKN(C)=C'INCR SECOND'
- CPUZ 14:50:13 TKN(H)=X'C9D5C3D940E2C5C3D6D5C4'
- CPUZ 14:50:13 TOTAL NUM OF VOLUMES=00004,INCREMENTAL=N
- CPUZ 14:50:13 SGNAME=DB2X96L SOURCE=X96L01 TARGET=FC6420
- CPUZ 14:50:13 SGNAME=DB2X96L SOURCE=X96L02 TARGET=FC6421
- CPUZ 14:50:13 SGNAME=DB2X96L SOURCE=X96L03 TARGET=FC6422
- CPUZ 14:50:13 SGNAME=DB2X96L SOURCE=X96L04 TARGET=FC6423

This example shows that Version 2 is not in an incremental FlashCopy relationship. Only one version of a copypool can be in a persistent incremental FlashCopy relationship.

DFSMShsm Apars

- OA18233 – Changing number of versions.
- OA18329 – Not all versions maintained.
- OA19713 – Adding volumes to storage group.
- OA19634 – Abend S0C9.
- OA19740 – FRRECOV and FRDELETE.
- OA20243 – FRRECOV DSNAME.
- OA20555 – FRRECOV TOVOLUME.
- OA20619 – FRBACKUP WITHDRAW.

Current apars as of May 2007 for DFSMShsm support of the DB2 volume-level utilities.

I hope these pictures help you remember the key features.

✓Clone Tables are like an End Loader. You can dump new data into them without affecting your production table.

✓Partition by Growth is like a Motor Grader. DB2 will spread your data across the partitions.

✓Index Compression is like a Combine. You will be able to harvest disk savings.

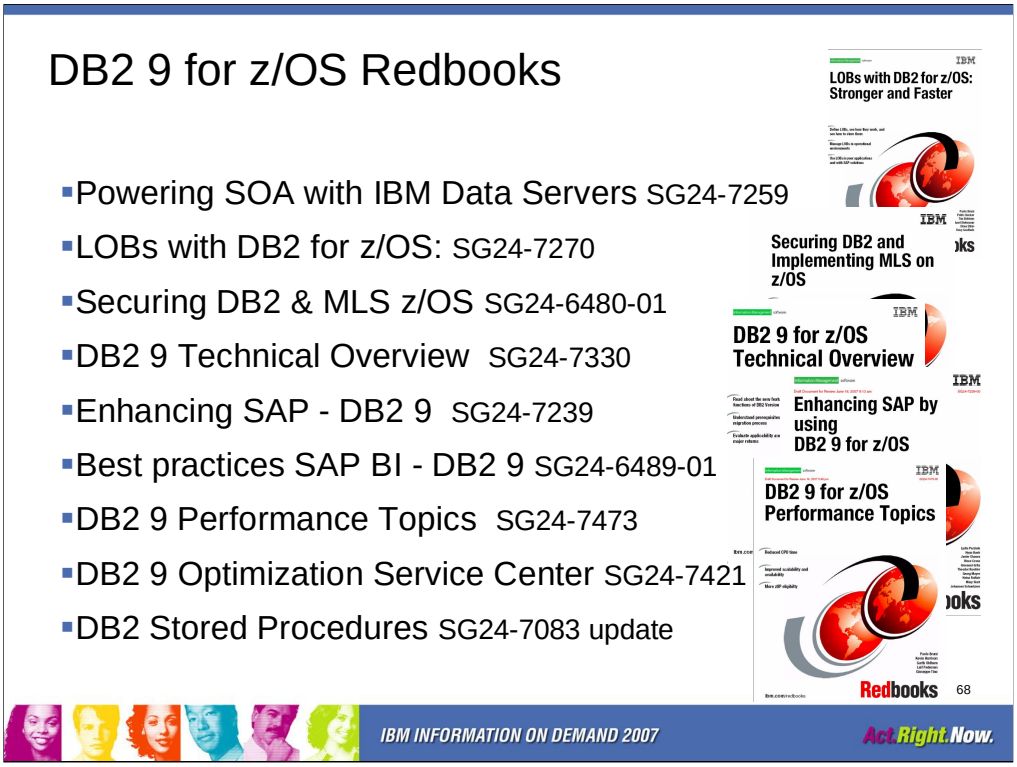
✓Not Logged Tablespaces are like an Excavator. You can dig yourself a really big hole.

✓Trusted Contexts are like a Tractor. Versatile, you are able to pull a lot of different equipment with them.

✓Template Switching is like a Gator, a utility vehicle.

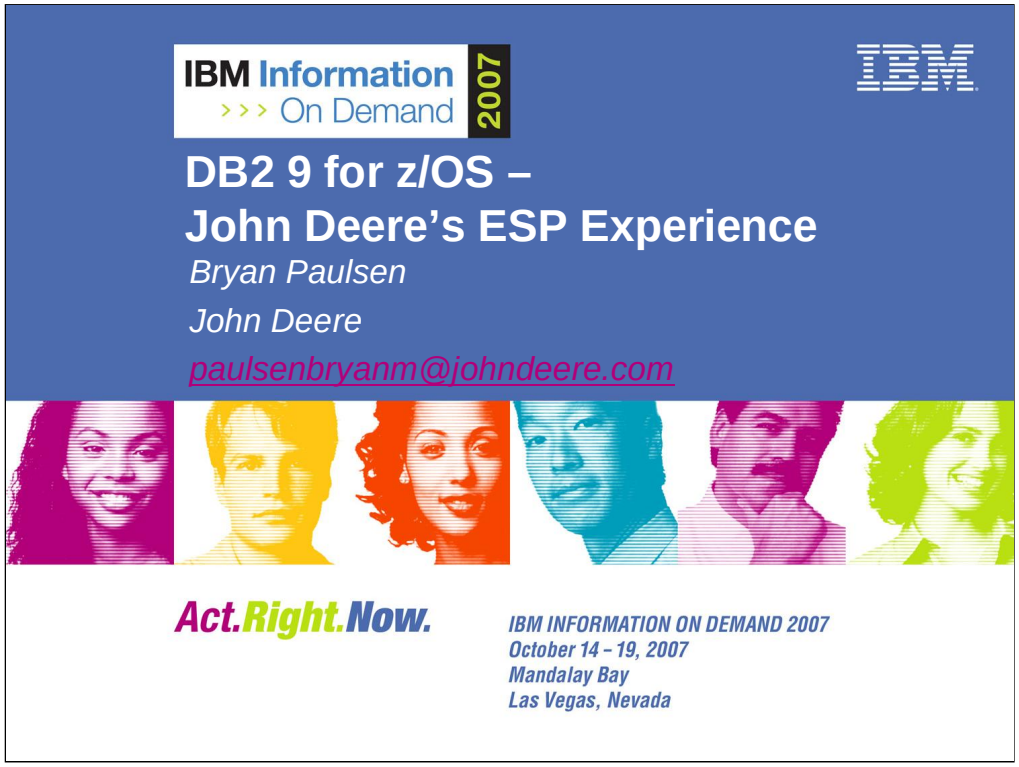
Summary

Not in DB2 9 for z/OS



See the DB2 library for detailed information.
http://www.ibm.com/software/data/db2/zos/library.html
Six redbooks with substantial DB2 9 content are on the web, with a draft being reviewed, one being written and and one more in the works for later this year.

- Powering SOA with IBM Data Servers, SG24-7259
http://www.redbooks.ibm.com/abstracts/SG247259.html
- LOBs with DB2 for z/OS: Stronger & Faster SG24-7270
http://www.redbooks.ibm.com/abstracts/SG247270.html
- Securing DB2 & MLS z/OS, SG24-6480-01
http://www.redbooks.ibm.com/abstracts/sg246480.html
- V9 Technical Overview, SG24-7330
http://www.redbooks.ibm.com/abstracts/SG247330.html
- Enhancing SAP - DB2 9, SG24-7239,
http://www.redbooks.ibm.com/abstracts/SG247239.html
- Best practices SAP BI - DB2 9, SG24-6489-01,
http://www.redbooks.ibm.com/redpieces/abstracts/sg246489.html
- DB2 V9 Performance Topics, SG24-7473, in review
http://www.redbooks.ibm.com/abstracts/SG247473.html
- DB2 V9 Optimization Service Center, SG24-7421, being written
- DB2 for z/OS Stored Procedures: CALL & Beyond SG24-7083-01 update later this year



Bryan Paulsen is a technology architect at John Deere in Moline, Illinois. He specializes in DB2 for z/OS in the DB2 Classic/Legacy and the DB2 SAP environments. His experiences include project manager for the DB2 9 and DB2 Version 8 Early Support Programs, DB2 disaster recovery technical lead, project manager for migrating SAP/R3 to DB2 for z/OS and DB2 systems programming. He has spoken at IDUG North America, IDUG Europe, SHARE, IBM's Information on Demand conference, IBM's z/SAP Technical Briefings and contributed to the IDUG Solutions Journal. He received the Best User Speaker award for the IDUG 2004 and 2007 North America conferences and is a member of the IDUG Speakers Hall of Fame. Bryan currently serves on the IDUG North America Conference Planning Committee.