

The DB2 Catalog: The Past, the Present, and the Future

John Lyle
IBM Silicon Valley Lab. (Etats Unis)

jlyle@us.ibm.com



Réunion du Guide DB2A
Jeudi 29 mai 2008
Feel Europe , Vincennes (94)
France

© 2008 IBM Corporation

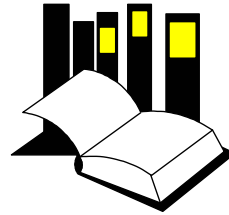
John Lyle has worked at the Silicon Valley Laboratory in the DB2 organization since 1985.

Joining DB2 development in 1995, he has worked in the RDS area focusing on the DB2 system catalog, migration and fallback and the data definitional language (DDL) areas.

He was the lead designer and implementer of the catalog changes for V5, V6, V7 and V8, the fallback coordinator for V5, V6, and V7 and has worked on numerous other Line Items in these releases. He spent some time in DB2's index manager component and is currently working in the DB2 utilities department.

OVERVIEW

- What is the DB2 Catalog?
- How do DB2 operations affect the Catalog and Directory?
- Pre-migration considerations - Data sharing and non-data sharing
- Catalog migration - what really happens? What's new in V6 – V9.
- The future - what's on the horizon?



© 2008 IBM Corporation

The purpose of this presentation is to unveil some of the mysteries of the DB2 catalog and the migration process.

The presentation will cover some basic catalog information before moving into the mysteries of a catalog migration. Will cover data sharing/non-data sharing migration considerations, catalog and migration enhancements for DB2 releases through DB2 V9, performance considerations, and finish with information about the future direction of the DB2 catalog and migration process. Will cover information about the various modes of V8 and V9.

What is the DB2 Catalog?

- The tables in the DSNDB06 database

- Record information about
 - objects
 - authorization
 - plans/packages/stored procedures
 - recovery (image copy, REORG, etc.)
 - statistics
 - string translation



- Are updated by DB2 operations and SQL INSERT, UPDATE and DELETE on some of the tables
- Are used by DB2 to obtain information about the system
- Can be queried using SQL

© 2008 IBM Corporation

The DB2 catalog consists of tables of data about everything defined to the DB2 system, including table spaces, indexes, tables, copies of table spaces and indexes, storage groups, and so forth. The DB2 catalog is contained in system database DSNDB06. When you create, alter, or drop any structure, DB2 inserts, updates, or deletes rows of the catalog that describe the structure and tell how the structure relates to other structures.

Catalog Table Spaces

- 28 table spaces through V9

- DSNDB06.SYSDBASE

- definition of objects and authorization

- DSNDB06.SYSPLAN

- plans, DBRMs, statements in the plan, plan authorization, plan dependencies

- DSNDB06.SYSCOPY

- recovery information

- others

- dsndb06.sysstats, dsndb06.sysviews, dsndb06.syspkage, dsndb06.sysdbaut, dsndb06.sysgpaut, dsndb06.sysgroup, dsndb06.sysuser, dsndb06.sysstr, dsndb06.sysddf (v5), dsndb06.sysobj (v6), *dsndb06.sysseq (v6), dsndb06.sysseq2 (v6), dsndb06.syshist (v7), dsndb06.sysgrtns (v7), dsndb06.sysjava (v7), dsndb06.sysjauxa (v7), dsndb06.sysjauxb (v7), sysalter (v8), sysebcdc (v8), sysroles(v9), syscontx(v9), sysrtsts(v9), sysxml(v9), systarg(v9), syspluxa(v9)*

© 2008 IBM Corporation

The SYSDBASE catalog table space, for example, contains 13 tables in V9. These tables contain information on the objects that make up a database: table spaces, tables, views, columns, indexes, relationships, etc. Additionally, view and table authorization information is also kept here.

In V8, the SYSLINKS catalog table was deleted from the SYSDBASE catalog table space during the V8 ENFM process. There were 14 tables in SYSDBASE before the SYSLINKS catalog table removal.

Evolution of the DB2 Catalog

The catalog continues to grow with every DB2 release.



DB2 Version	Table Spaces	Tables	Indexes	Columns	Table Check Constraints
V1	11	25	27	269	n/a
V3	11	43	44	584	n/a
V4	11	46	54	628	0
V5	12	54	62	731	46
V6	15	65	93	967	59
V7	20	82	119	1206	105
V8	22	88	132	1277	105
V9	28	102	165	1661	119

© 2008 IBM Corporation

This table indicates how the catalog has grown and how it continues to grow with each DB2 release. Perhaps most significant in this table is the growth in the number of table spaces and indexes. This is because of the need to maintain the underlying VSAM data sets for them. The more catalog table spaces and indexes you have the more objects you have to manage.

What is the DB2 Directory?

- The objects in the DSNDB01 database

- Record information about

- objects, DBDs (DSNDB01.DBD01)
- plans, SKCTs (DSNDB01.SCT02)
- packages, SKPTs (DSNDB01.SPT01)
- status of utilities currently in progress (DSNDB01.SYSUTILX)
- Log RBA ranges (DSNDB01.SYSLGRNX)

- Are updated by DB2 operations

- Are used by DB2 to access information quickly

- Cannot be queried using SQL



© 2008 IBM Corporation

The DB2 directory contains information used by DB2 during normal operation. You cannot access the directory using SQL, although much of the same information is contained in the DB2 catalog, for which you can submit queries. Directory objects are not described in the DB2 catalog.

The DB2 directory consists of 5 table spaces:

SCT02 is the skeleton cursor table space (SKCT), which contains the internal form of SQL statements contained in an application. When you bind a plan, DB2 creates a skeleton cursor table in SCT02.

SPT01 is the skeleton package table space (SKPT), which is similar to SCT02 except that SPT01 contains package information. When you bind a package, DB2 creates a skeleton package table in SPT01.

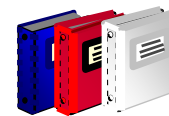
SYSLGRNX is the log range table space, used to track the opening and closing of table spaces, indexes, or partitions.

SYSUTILX is the system utilities table space that contains a row for every utility job that is running. Information in this table remains until the utility completes.

DBD01 is the database descriptor (DBD) table space, which contains internal information, called database descriptors (DBDs), about the databases existing within DB2.

How do DB2 operations affect the Catalog and Directory?

- CREATE TABLE
- ALTER INDEX
- ALTER TABLE ADD COLUMN
- ALTER TABLE ALTER COLUMN SET DATA TYPE
- LABEL ON / COMMENT ON
- GRANT
- BIND PLAN
- DROP TABLE
- FREE PLAN



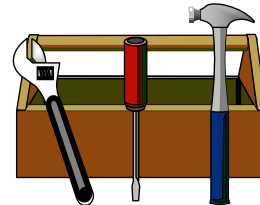
© 2008 IBM Corporation

For V5, there are 68 SQL statements available, For V6, there are 91. For V7 there are 105. For V8 there are 119. This includes multiple forms of some statements (i.e., the CREATE, GRANT, REVOKE, ALTER, DECLARE, etc. statements). Not all of these statements have an effect on the DB2 catalog and/or directory (i.e. CALL, OPEN, SET, etc.).

The CREATE TABLE statement, for example, will update the applicable SYSTABLESPACE row (i.e. the NUMTABLES column), will insert a new row into SYSTABLES, will insert one row for each table column into SYSCOLUMNS, and make additional catalog updates depending on the syntax specified.

The ALTER TABLE ALTER COLUMN SET DATA TYPE statement, new in V8, allows the length attribute of existing CHAR columns to be increased.

What tools can be used to tune the Catalog and Directory?



- REORG TABLESPACE / INDEX
- CREATE INDEXES ON CATALOG TABLES
- SQL ALTER INDEX / TABLESPACE ON CATALOG OBJECTS

© 2008 IBM Corporation

The REORG TABLESPACE utility reorganizes a table space to improve access performance and reclaim fragmented space. Data that is organized well physically can improve the performance of access paths that rely on index or data scans, and it can also help reduce the amount of DASD used by the index or table space.

REORG TABLESPACE can be executed on the table spaces in the DB2 catalog database (DSNDB06) and some table spaces in the directory database (DSNDB01). It cannot be executed on any table space in the DSNDB07 database. Reorganizes all catalog data and non-partitioning indexes.

REORG TABLESPACE

- **REORG TABLESPACE** on any Catalog or Directory table space except for DSNDB01.SYSUTILX.
- Authorization needed
 - At least, the Reorg privilege for DSNDB06
 - If SYSUSER or SYSDBAUT table spaces or indexes are unavailable, install SYSADM or install SYSOPR is required
- LOG NO required (**LOG YES for LOB table spaces - V7 added SYSJAUXA and SYSJAUXB**)
- UNLOAD ONLY not allowed
- Recoverable point required
 - **Must** run DSNTIJIC or similar job to image copy ALL of the Catalog and Directory table spaces that will be REORGed prior to REORG TABLESPACE

© 2008 IBM Corporation

Catalog and directory table spaces: Before running REORG on a catalog or directory table space, you must take an image copy. Be aware that for the DSNDB06.SYSCOPY and DSNDB01.DBD01 table spaces, REORG scans logs to verify that an image copy is available. If the scan of logs does not find an image copy, DB2 will request archive logs.

Must take an image copy after a catalog or directory table spaces reorg to remove the COPY pending state.

In V8, you can use an online reorg to reorganize your catalog and directory table spaces. Specify SHRLEVEL REFERENCE with the COPYDDN keyword and your catalog table space will not be in COPY PENDING when the reorg completes. In an offline REORG (SHRLEVEL NONE), the table space is in COPY PENDING after the REORG so another image copy must be done before the table space is available for use. So for availability, I'd recommend using online reorgs against the catalog and directory starting in V8.

REORG TABLESPACE

- Run **CHECK INDEX** prior to Reorg!
- PCTFREE** and **FREEPAGE** observed for most table spaces and all indexes. Use SQL ALTER to change values.
- When Reorganizing a Directory table space, consider starting DSNDB01 for utility access.
- REORG TABLESPACE** on SYSCOPY with LOG NO leaves the table space in copy pending.

- In V8, can use **SHRLEVEL REFERENCE** with the **COPYDDN** keyword to avoid copy pending state!



© 2008 IBM Corporation

We recommend running CHECK INDEX prior to reorganizing the DB2 catalog and Directory to make sure all the catalog indexes are in good shape.

Must take an image copy after a catalog or directory table spaces reorg to remove the COPY pending state.

Can use online REORG in V8 (SHRLEVEL REFERENCE) to avoid COPY PENDING state.

REORG TABLESPACE

Additional considerations for the following table spaces:

sysdbase, sysdbaut, sysgroup, sysplan, sysviews, dbd01

- **WORKDDN, SORTDATA, SORTDEVT, SORTNUM, SORTKEYS, COPYDDN, and RECOVERYDDN** ignored for **SHRLEVEL NONE**. **COPYDDN** and **RECOVERYDDN** can be used for **SHRLEVEL REFERENCE** reorgs.

- No **BUILD** and **SORT** phases

- Reorg performance is slower on these table spaces

- Run **DSN1CHKR, DSN1COPY CHECK** prior to **REORG**

- Calculation for unload data set size is different

- $(28 + \text{longest record length in table space}) * \text{numrows}$



© 2008 IBM Corporation

For table spaces with links there is no build or sort phase of REORG. For these table spaces, REORG TABLESPACE reloads the indexes and data during the reload phase.

The DSN1CHKR utility verifies the integrity of DB2 directory and catalog table spaces. DSN1CHKR scans the specified table space for broken links, broken hash chains, and records that are not part of any link or chain.

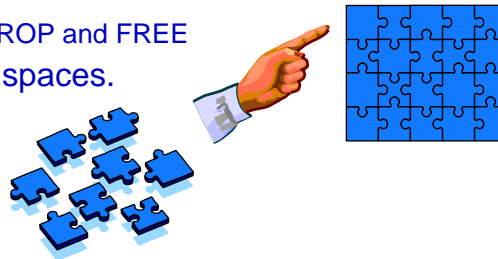
DSN1COPY CHECK Checks each page from the SYSUT1 data set for validity. The validity checking operates on one page at a time and does not include any cross-page checking. If an error is found, a message is issued describing the type of error, and a dump of the page is sent to the SYSPRINT data set. If you do not receive any messages, no errors were found. If more than one error exists in a given page, the check identifies only the first of the errors. However, the entire page is dumped.

For the unload data set size calculation, the constant 28 is for the maximum number of 'link' bytes for any of the catalog tables. Using the constant 28 ensures that the calculations will yield enough space for the unload data set.

REORG TABLESPACE

•How do I know when to run REORG?

- Look at statistics from **RUNSTATS** utility
 - For SYSDBASE, SYSPLAN and SYSVIEWS a higher value for SYSINDEXPART.NEAROFFPOSF and SYSINDEXPART.FAROFFPOSF is acceptable.
- FRAGMENTATION
 - Caused by extensive use of ALTER, GRANT, REVOKE, etc.
- WASTED SPACE
 - Caused by extensive use of DROP and FREE
- No stats for Directory table spaces.
 - SYSDBASE -> DBD01
 - SYSPLAN -> SCT02
 - SYSPKAGE -> SPT01



© 2008 IBM Corporation

When to run REORG on the catalog and directory: You should not need to run REORG TABLESPACE on the catalog and directory table spaces as often as you do on user table spaces. The statistics collected by RUNSTATS that you use to determine if a REORG is required for a user table space can also be used for the catalog table spaces. The only difference is the information in the columns NEAROFFPOSF and FAROFFPOSF in table SYSINDEXPART. These columns can tolerate a higher value before a reorganization is needed if the table space is DSNDB06.SYSDBASE, DSNDB06.SYSVIEWS, or DSNDB06.SYSPLAN

There are no statistics that can be used to determine when the directory table spaces should be REORGed. The directory table spaces should be REORGed when the associated catalog table space is reorganized as follows:

DSNDB06.SYSDBASE -> DSNDB01.DBD01
DSNDB06.SYSPLAN -> DSNDB01.SCT02
DSNDB06.SYSPKAGE -> DSNDB01.SPT01

REORG TABLESPACE

- Consider running Reorg against a table space if there is extensive use of the DB2 operations noted:

TABLE SPACE	DB2 OPERATION
SYSCOPY	UTILITIES
SYSDBASE, DBD01	CREATE, ALTER, DROP, LABEL ON, COMMENT ON, GRANT and REVOKE
SYSDBAUT	CREATE, ALTER, DROP, GRANT, REVOKE on database
SYSGPAUT	GRANT, REVOKE of buffer pools, stogroup, table spaces, collection privileges
SYSGROUP	CREATE, ALTER, DROP for stogroup statements
SYSPKAGE, SPT01	FREE, BIND, REBIND, GRANT, REVOKE for packages
SYSPLAN, SCT02	FREE, BIND, REBIND, GRANT, REVOKE for plans
SYSSTR	CREATE, ALTER of check constraints on tables
SYSSTATS	RUNSTATS, DROP, FREE
SYSUSER	GRANT, REVOKE of user privileges
SYSVIEWS	CREATE, DROP of views

REORG TABLESPACE

- Consider running Reorg against a table space if there is extensive use of the DB2 operations noted:

TABLE SPACE	DB2 OPERATION
SYSDDF (V5)	INSERT, UPDATE, DELETE of CDB tables
SYSOBJ (V6)	CREATE, ALTER, DROP, COMMENT ON on UDFs, UDTs, triggers, stored procedures
SYSSEQ (V6)	CREATE, ALTER, DROP of tables with Identity Columns
SYSSEQ2 (V6)	CREATE, ALTER, DROP of tables with Identity Columns
SYSHIST (V7)	RUNSTATS, DROP, FREE
SYSALTER(V8)	Versioning ALTERs, DROP of tables that had versioning

REORG TABLESPACE

- Consider running Reorg against a table space if there is extensive use of the DB2 operations noted:

TABLE SPACE	DB2 OPERATION
SYSROLES (V9)	Role authorization changes
SYSCTX (V9)	CREATE, ALTER, DROP of trusted contexts
SYSRTSTS (V9)	RTS activity
SYSXML (V9)	CREATE, ALTER, DROP of tables with XML columns
SYSTARG (V9)	CREATE, ALTER, DELETE for indexes with Index on expression

SQL ALTER, CREATE, DROP

- The following SQL statements and options are allowed against objects in the Catalog database:
 - ALTER INDEX (IBM defined)
 - CLOSE, FREEPAGE, GBPCACHE, PCTFREE, PIECESIZE, COPY
 - ALTER INDEX (User defined indexes)
 - All clauses except BUFFERPOOL
 - ALTER TABLE
 - DATA CAPTURE CHANGES
 - ALTER TABLESPACE
 - CLOSE, FREEPAGE, GBPCACHE, LOCKMAX, MAXROWS, PCTFREE, TRACKMOD
 - CREATE / DROP INDEX (User Defined Indexes only)
 - All clauses allowed except CLOSE YES, CLUSTER, UNIQUE, DEFER YES, BUFFERPOOL (other than BP0)
 - A maximum of 100 User Defined Indexes allowed on the Catalog through V8. **Now 200 in V9!**

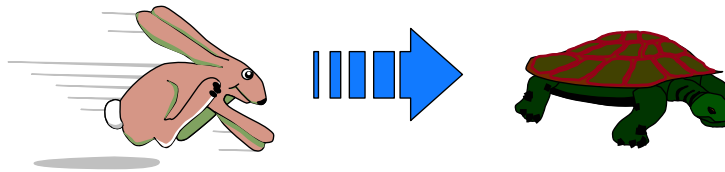
© 2008 IBM Corporation

For ALTER INDEX on IBM defined indexes, the GBPCACHE value cannot be altered for the following SYSINDEXES indexes: DSNDXX01, DSNDXX02, and DSNDXX03.

For ALTER TABLESPACE, GBPCACHE and/or MAXROWS cannot be specified for some of the catalog table spaces. Consult the SQL Reference for a complete list of these table space restrictions.

CREATE INDEX on the Catalog

- Useful for tuning applications that select from catalog tables.
- DB2 operations will **NOT** use User Defined Indexes on the Catalog.
- Every time DB2 updates a Catalog table that has a User Defined Index, it will need to update the new index also. This can affect performance of DB2 operations so be careful.



© 2008 IBM Corporation

Just be careful when creating indexes on the DB2 catalog. DB2 itself will not use your indexes. Only your own applications will use these indexes. If you put too many indexes in the wrong place then you could greatly slow down your subsystem because of the additional overhead needed to update these indexes when the catalog is changing. If you create indexes for applications that run once a year, create and delete those indexes during that process so the index overhead isn't experienced throughout the rest of the year.

How can you identify problems in the Catalog and Directory?

- SQL queries in **DSNTESQ** sample
 - looks for Catalog table spaces recovered to different points in time
 - looks for logical relationships between Catalog tables
 - Run **RECOVER TABLESPACE** to fix
- **DSN1COPY CHECK**
 - looks for problems with strict clustering
 - Run **REORG TABLESPACE** to fix
- **DSN1CHKR**
 - looks for problems with links
 - Use **REPAIR** to fix broken links
- **CHECK DATA / INDEX**
 - looks for: violations of referential and table check constraints / indexes to match data
 - Run **RECOVER** to fix data / **REBUILD INDEX** to fix indexes



© 2008 IBM Corporation

The performance of some of the DSNTESQ queries were improved via APAR PQ71775 (2/04).

Pre-migration considerations - Data sharing and non-data sharing

- From a catalog perspective, there really is no difference between a data sharing and a non-data sharing migration
- Fallback SPE (Small Program Enhancement) - **pre-V8**
 - In data sharing, DB2 enforces that each active group member has the fallback SPE on before the new version of DB2 is allowed to start. Not enforced for migrating group member.
 - In non-data sharing, DB2 does not enforce the fallback SPE to be on before allowing a migration to the new release



© 2008 IBM Corporation

For non-data sharing subsystems, I'd recommend that your DB2 subsystems be at a code level that contains the fallback SPE for the next release BEFORE migrating to the next release. This way you won't have to apply maintenance while you're trying to fallback. It'd be best to migrate from and fallback to the same code level. This will be required in DB2 V8!

Note: In V8 it is required that non-data sharing DB2s run with the fallback SPE prior to migrating to the a DB2 release. DB2 must actually be started in V7 with the V8 fallback SPE on before attempting to start a V8 subsystem. A V8 subsystem will not even start if the last time V7 was up and running it did not have the V8 fallback SPE code on.

Pre-migration considerations - Data sharing and non-data sharing

- Fallback SPE – Changes starting in V8

- In data sharing, SPE must be on all active group members
INCLUDING the migrating group member

- In non-data sharing, SPE **must** be on Vn subsystem with DB2 started before attempting to start Vn+1 subsystem. **This is enforced!**



**Make sure the fallback
SPE is on your Vn
subsystems before
attempting to migrate to
Vn+1!**

© 2008 IBM Corporation

Starting in V8, you cannot migrate a DB2 subsystem unless it was running in V7 with the fallback SPE on. The SPE had to be on the V7 subsystem the last time DB2 was started on V7 or V8 will not start. This is for both data sharing and non-data sharing.

Catalog migration - what really happens?

- The **DSNTIJTC** migration process is not the 'black hole' that everyone thinks it is

- Authorization check (INSTALL SYSADM) - DSNU760

- Verify Catalog is at correct level for migration - this check varies with each DB2 release - DSNU765, DSNU766

- New release DDL

- Additional Catalog updates and data checking

- Update Directory header page and BSDS/SCA with new release information



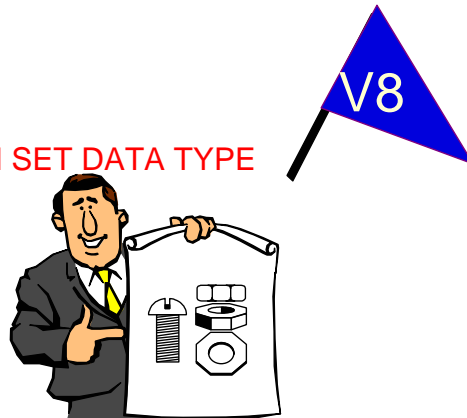
© 2008 IBM Corporation

The migration process shouldn't scare anyone anymore. The process has been greatly improved during my tenure and hopefully this has been seen in the past several release migrations. In the coming charts I'll explain exactly what's being done during the migration process and what you can do to make sure that the process is actually running.

Catalog migration - New release DDL

- DDL executed to migrate the DB2 Catalog is the same DDL that you use to build user objects:

- CREATE TABLESPACE
- CREATE TABLE
- ALTER TABLE ALTER COLUMN SET DATA TYPE
- ALTER TABLE
- CREATE INDEX



© 2008 IBM Corporation

The DDL executed during any given migration is the same as any DDL that everyone uses. There may be exceptions to this 'rule' depending on the release being migrated from and to. This is why we encrypt the migration DDL before shipping it.

Catalog migration - Additional updates made during migration

- Varies from release to release
 - Copying information from one column to another
 - Copying information from one or more columns to another after the data is massaged (i.e. DATE and TIME to TIMESTAMP, INTEGER to FLOAT, etc.)
 - Changing column values

© 2008 IBM Corporation

These additional updates that are made during the migration process are pretty much a thing of the past. Making these updates make for long migration times so we will avoid making them if at all possible.

We did loads of these during the V5 migration process.

Catalog migration - Additional updates made during migration

- Not doing much of this after V5!
 - A majority of the Catmaint processing time is spent in making these additional updates (most made via table space scans).
 - By eliminating these updates we have been able to greatly reduce migration times.
 - Saw a big difference in a V6 migration, vs. a V5 migration, and V7/V8/V9 migrations were very fast!

© 2008 IBM Corporation

These additional updates are pretty much a thing of the past (as of V6 and later releases). This additional processing is the main reason catmaint takes as long as it does. If we don't do much of these additional updates then migration times will greatly improve. I'm sure this will be evident in your V6 and V7 migrations times.

In V8, we have a return of some of this processing as we have to prevent migrations to V8 if there are any Type 1 indexes noted in the catalog.

Catalog migration - Additional data checking



- Varies from release to release

- Look for values that will violate new Catalog table check constraints. 'Fix' bad values.

- Look for values or functions that are no longer allowed

- In V6 and V7: TYPE 1 indexes, data set passwords, shared read only data are still not allowed but migration processing will **not** terminate if you have any of them.



- In V8: TYPE 1 indexes are still not allowed and migration processing **will** terminate if you have any of them.

Catalog migration job - DSNTIJTC

- V6 migration job is 3 steps and V7 migration job is 2 - 3 steps:

- Step 1 - Mandatory migration processing.
Process DDL and any additional updates. **No table space scan!!! This step is extremely fast.**
- Step 2 - Looks for unsupported objects.
- Step 3 - Stored procedure migration processing
(not used for V6 - V7 migrations).



- V8/V9 migration job is 1 step:

- Step 1 - Mandatory migration processing.
Process DDL and any additional updates. **V8 Table space scan so not as quick as step 1 in V6 - V7.**



© 2008 IBM Corporation

Step 3 will not be executed for a V6 - V7 migration. This is because the step 3 processing, stored procedure migration processing, would have been done on a V5 to V6 migration.

Catalog migration - Is Catmaint doing anything?

- Can tell what Catmaint is doing:
 - Look at SYSPRINT data set for migration status messages (DSNU777)
 - These messages are issued at beginning of each Catmaint step. These messages will vary from release to release.
 - Use the **DISPLAY UTILITY DB2** command
 - Displays number of rows processed during table space scan steps. If you know number of rows in table space you can determine percentage of processing completed for that table space. Example: **-DISPLAY UTIL(*)**



© 2008 IBM Corporation

In the 'old days', migrations could take extended periods of time and there was no way to tell what, if anything, CATMAINT was doing. In V5 I added the ability to tell exactly what's happening and the catmaint processing that's being done. This can be done in 2 ways: by using the DISPLAY Utility and by looking at the SYSPRINT data set to see the migration messages that are being issued as the CATMAINT process continues.

Catalog migration - Using DISPLAY to check Catmaint progress

- Information available varies from release to release

- SYSDBASE and SYSSTATS (V5)
- Not applicable in V6 and in V7. Can still use in Phase 2 Catmaint processing (looking for unsupported objects).
- SYSDBASE (V8)

- Version 8 example:

```
.DISPLAY UTIL(*)
DSNU105I . DSNUGDIS - USERID = SYSADM
      MEMBER = V81A
      UTILID = RELODCAT
      PROCESSING UTILITY STATEMENT 1
      UTILITY = CATMAINT
      PHASE = CATMAINT  COUNT = 5316
      STATUS = ACTIVE
DSN9022I . DSNUGCCC '-DISPLAY UTIL' NORMAL COMPLETION
```



© 2008 IBM Corporation

The DISPLAY command is a useful way to determine if catmaint is doing anything. In V6 and V7 this ability doesn't provide much usefulness since the utility counter is only incremented during table space scan processing. In V6 and V7 there is no table space scan processing in Phase 1, where the catalog is actually migrated, so the utility counter is not used. Phase 2 catmaint processing still uses this counter and DISPLAY can still be used to ensure that actual processing is proceeding. But your catalog has already been migrated at this point.

In V8, there is a table space scan in step 1 migration processing again so the DISPLAY command can be used to determine how much of the SYSDBASE table space has already been processed.

Catalog migration - Catmaint status messages

•V5 Catmaint progress messages displayed:

```
DSNU000I DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = RELODCAT
DSNU050I DSNUGUTC - CATMAINT UPDATE
DSNU750I DSNUECM0 - CATMAINT UPDATE STARTED
DSNU777I DSNUECM0 - CATMAINT UPDATE STATUS - VERIFYING CATALOG IS AT CORRECT LEVEL FOR MIGRATION.
DSNU777I DSNUECM0 - CATMAINT UPDATE STATUS - BEGINNING MIGRATION SQL PROCESSING PHASE.
DSNU777I DSNUEXUP - CATMAINT UPDATE STATUS - BEGINNING SYSGPAUT TABLE SPACE MIGRATION PROCESSING.
DSNU777I DSNUEXUP - CATMAINT UPDATE STATUS - BEGINNING SYSGROUP TABLE SPACE MIGRATION PROCESSING.
DSNU777I DSNUEXUP - CATMAINT UPDATE STATUS - BEGINNING SYSPLAN TABLE SPACE MIGRATION PROCESSING.
DSNU777I DSNUEXUP - CATMAINT UPDATE STATUS - BEGINNING SYSUSER TABLE SPACE MIGRATION PROCESSING.
DSNU777I DSNUEXUP - CATMAINT UPDATE STATUS - BEGINNING SYSSTR TABLE SPACE MIGRATION PROCESSING.
DSNU777I DSNUEXUP - CATMAINT UPDATE STATUS - BEGINNING SYSDBAUT TABLE SPACE MIGRATION PROCESSING.
DSNU777I DSNUEXUP - CATMAINT UPDATE STATUS - BEGINNING SYSPKAGE TABLE SPACE MIGRATION PROCESSING.
DSNU777I DSNUEXUP - CATMAINT UPDATE STATUS - BEGINNING SYSSTATS TABLE SPACE MIGRATION PROCESSING.
DSNU777I DSNUEXUP - CATMAINT UPDATE STATUS - BEGINNING SYSDBASE TABLE SPACE MIGRATION PROCESSING.
DSNU777I DSNUEXUP - CATMAINT UPDATE STATUS - PROCESSING SYSSTRINGS TABLE UPDATES.
DSNU777I DSNUECM0 - CATMAINT UPDATE STATUS - UPDATING DIRECTORY WITH NEW RELEASE MARKER.
DSNU752I DSNUECM0 - CATMAINT UPDATE COMPLETED
DSNU010I DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0
```

© 2008 IBM Corporation

The V5 migration process is a 1 step process.
It's all or nothing.

Catalog migration - Catmaint status messages

•V6 Catmaint progress messages displayed:

```
DSNU0001 DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = RELODCAT
DSNU0501 DSNUGUTC - CATMAINT UPDATE
DSNU7501 DSNUECM0 - CATMAINT UPDATE PHASE 1 STARTED
DSNU7771 DSNUECM0 - CATMAINT UPDATE STATUS - VERIFYING CATALOG IS AT CORRECT LEVEL FOR MIGRATION.
DSNU7771 DSNUECM0 - CATMAINT UPDATE STATUS - BEGINNING MIGRATION SQL PROCESSING PHASE.
DSNU7771 DSNUEXUP - CATMAINT UPDATE STATUS - BEGINNING ADDITIONAL CATALOG UPDATES PROCESSING.
DSNU7771 DSNUECM0 - CATMAINT UPDATE STATUS - UPDATING DIRECTORY WITH NEW RELEASE MARKER.
DSNU7521 DSNUECM0 - CATMAINT UPDATE PHASE 1 COMPLETED
DSNU0101 DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0
DSNU0001 DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = RELODST2
DSNU0501 DSNUGUTC - CATMAINT UPDATE UNLDDN STEP2
DSNU7501 DSNUECM0 - CATMAINT UPDATE PHASE 2 STARTED
DSNU7771 DSNUECM0 - CATMAINT UPDATE STATUS - VERIFYING CATALOG IS CORRECT FOR MIGRATION PHASE 2 PROCESSING
DSNU7771 DSNUECM0 - CATMAINT UPDATE STATUS - BEGINNING MIGRATION PHASE 2 PROCESSING.
DSNU7771 DSNUEXUP - CATMAINT CHECK STATUS - BEGINNING SYSDBAUT TABLE SPACE MIGRATION PROCESSING.
DSNU7771 DSNUEXUP - CATMAINT CHECK STATUS - BEGINNING SYSDBASE TABLE SPACE MIGRATION PROCESSING.
DSNU7521 DSNUECM0 - CATMAINT UPDATE PHASE 2 COMPLETED
DSNU0101 DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0
DSNU0001 DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = RELODSTP
DSNU0501 DSNUGUTC - CATMAINT UPDATE UNLDDN STORPROC
DSNU7501 DSNUECM0 - CATMAINT UPDATE PHASE 3 STARTED
DSNU7771 DSNUECM0 - CATMAINT UPDATE STATUS - VERIFYING CATALOG IS CORRECT FOR STORED PROCEDURE MIGRATION
PROCESSING.
DSNU7771 DSNUECM0 - CATMAINT UPDATE STATUS - BEGINNING STORED PROCEDURE MIGRATION PROCESSING.
DSNU7521 DSNUECM0 - CATMAINT UPDATE PHASE 3 COMPLETED
DSNU0101 DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0
```

© 2008 IBM Corporation

With PQ38035, the V6 catmaint job becomes a 3 step job. A second step is inserted to look for unsupported objects. The third step, which was previously the second step, does the stored procedure migration processing.

Catalog migration - Catmaint status messages

•V7 Catmaint progress messages displayed:

```
DSNU000I DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = RELODCAT
DSNU050I DSNUGUTC - CATMAINT UPDATE
DSNU750I DSNUECM0 - CATMAINT UPDATE PHASE 1 STARTED
DSNU777I DSNUECM0 - CATMAINT UPDATE STATUS - VERIFYING CATALOG IS AT CORRECT LEVEL FOR MIGRATION.
DSNU777I DSNUECM0 - CATMAINT UPDATE STATUS - BEGINNING MIGRATION SQL PROCESSING PHASE.
DSNU777I DSNUECM0 - CATMAINT UPDATE STATUS - BEGINNING SYSSTRINGS TABLE UPDATES.
DSNU777I DSNUECM0 - CATMAINT UPDATE STATUS - UPDATING DIRECTORY WITH NEW RELEASE MARKER.
DSNU752I DSNUECM0 - CATMAINT UPDATE PHASE 1 COMPLETED
DSNU010I DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0
DSNU000I DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = RELODST2
DSNU050I DSNUGUTC - CATMAINT UPDATE UNLDDN STEP2
DSNU750I DSNUECM0 - CATMAINT UPDATE PHASE 2 STARTED
DSNU777I DSNUECM0 - CATMAINT UPDATE STATUS - VERIFYING CATALOG IS CORRECT FOR MIGRATION STEP 2
PROCESSING
DSNU777I DSNUECM0 - CATMAINT UPDATE STATUS - BEGINNING MIGRATION STEP 2 PROCESSING.
DSNU777I DSNUEXUP - CATMAINT UPDATE STATUS - BEGINNING SYSDBAUT TABLE SPACE MIGRATION PROCESSING.
DSNU777I DSNUEXUP - CATMAINT UPDATE STATUS - BEGINNING SYSDBASE TABLE SPACE MIGRATION PROCESSING.
DSNU752I DSNUECM0 - CATMAINT UPDATE PHASE 2 COMPLETED
DSNU010I DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0
```

© 2008 IBM Corporation

The V7 migration job is a two or three step job depending on which release you're migrating from.

If migrating to V7 from V6 it will be a two step job. In this case it won't have the third step that does the stored procedure migration processing. This is because this stored procedure migration processing would have already been done during the V5 to V6 migration.

If migrating to V7 from V5 then the migration job will contain all three steps.

Catalog migration - Catmaint status messages

•V8 Catmaint progress messages displayed:

```
DSNU000I DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = RELODCAT
DSNU1044I DSNUGTIS - PROCESSING SYSIN AS EBCDIC
DSNU050I DSNUGUTC - CATMAINT UPDATE
DSNU750I DSNUECM0 - CATMAINT UPDATE PHASE 1 STARTED
DSNU777I DSNUECM0 - CATMAINT UPDATE STATUS - VERIFYING CATALOG IS AT CORRECT LEVEL FOR MIGRATION
DSNU777I DSNUECM0 - CATMAINT UPDATE STATUS - BEGINNING MIGRATION SQL PROCESSING PHASE
DSNU777I DSNUEXUP - CATMAINT CHECK STATUS - BEGINNING SYSDBASE TABLE SPACE MIGRATION PROCESSING.
DSNU777I DSNUEXUP - UPDATE BEGINNING ADDITIONAL CATALOG UPDATES PROCESSING.
DSNU777I DSNUEXUP - CATMAINT UPDATE STATUS - PROCESSING SYSSTRINGS TABLE UPDATES.
DSNU777I DSNUECM0 - CATMAINT UPDATE STATUS - UPDATING DIRECTORY WITH NEW RELEASE MARKER.
DSNU752I DSNUECM0 - CATMAINT UPDATE PHASE 1 COMPLETED
DSNU777I = DSNUECM5 - CATMAINT UPDATE STATUS - CCSID UPDATES COMPLETED.
DSNU010I DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0
```



© 2008 IBM Corporation

The V8 migration job is a one step job. Everything must complete successfully. Once completed, your subsystem/group is now in what's called V8 compatibility mode (cm). There are 3 V8 modes: compatibility mode, enabling new function mode, and new function mode.

The V8 modes are described in complete detail later on in the presentation.

Catalog migration - Catmaint status messages

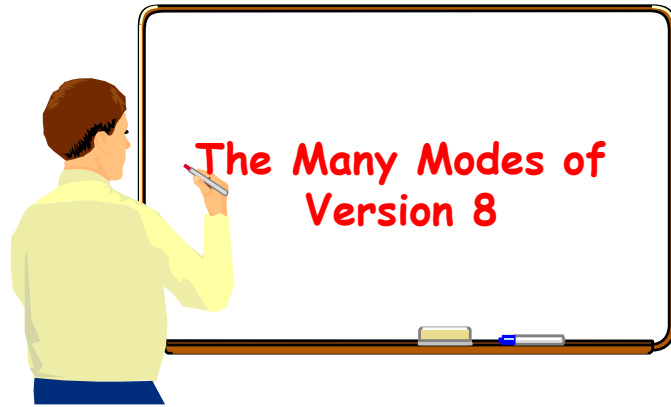
- V9 Catmaint progress messages displayed:

```
DSNU000I 135 14:44:51.80 DSNUGUTC - OUTPUT START FOR UTILITY, UTILID =  
DSNU1044I 135 14:44:52.68 DSNUGTIS - PROCESSING SYSIN AS EBCDIC  
DSNU050I 135 14:44:52.76 DSNUGUTC - CATMAINT UPDATE  
DSNU750I 135 14:44:52.81 DSNUECM0 - CATMAINT UPDATE PHASE 1 STARTED  
DSNU777I 135 14:44:52.81 DSNUECM0 - CATMAINT UPDATE STATUS - VERIFYING CATALOG IS AT  
CORRECT LEVEL FOR MIGRATION.  
DSNU777I 135 14:44:52.87 DSNUECM0 - CATMAINT UPDATE STATUS - BEGINNING MIGRATION SQL  
PROCESSING PHASE.  
DSNU777I 135 14:45:06.79 DSNUEXUP - CATMAINT UPDATE STATUS - BEGINNING ADDITIONAL CATALOG  
UPDATE PROCESSING.  
DSNU777I 135 14:45:06.79 DSNUECM0 - CATMAINT UPDATE STATUS - UPDATING DIRECTORY WITH NEW  
RELEASE MARKER.  
DSNU752I 135 14:45:06.80 DSNUECM0 - CATMAINT UPDATE PHASE 1 COMPLETED  
DSNU010I 135 14:45:06.99 DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0
```



© 2008 IBM Corporation

These are the migration progress messages that CATMAINT puts out. These can be seen during migration processing by looking at the SYSPRINT data set.



The Many Modes of V8

→ DB2 introduces 3 new modes in V8



*f*Compatibility mode (V8 **cm**)

*f*Enabling new function mode (V8 **enfm**)

*f*New function mode (V8 **nfm**)

→ Single code base for all V8 modes

© 2008 IBM Corporation

V8 introduces the notion of DB2 'modes'. There are three distinct modes in version 8: compatibility mode, enabling new function mode, and new function mode.

There is a single code base for all these V8 modes so you don't have to worry about adding any kind of maintenance between modes.

Compatibility Mode

→ Compatibility mode (cm) is essentially a traditional release migration



f Same migration process as in previous releases

f Most new release function is not available

–V8 exceptions? Online Reorg, 64 bit, ?

© 2008 IBM Corporation

In previous releases, a migration to a new release meant that new release function could be used immediately. It also meant that all 'must complete' migration processing had been completed.

In V8, this traditional migration process takes a DB2 subsystem from V7 to V8 compatibility mode. The DB2 catalog has been tailored and changed (although all the V8 catalog updates have not been made during this compatibility mode migration).

New V8 function is not available in V8 compatibility mode.

Enabling New Function Mode

→ Enabling new function mode (enfm)

*f*In V8 this is a **conversion** process

- Long name processing
- Unicode conversion of catalog data
- NOT PADDED catalog indexes (non-user)

*f*New release function is still not available

*f*Same V8 code level as in other modes

*f***Group wide event**



© 2008 IBM Corporation

During enable new function mode, additional catalog tailoring is done. For V8, this consists of long name DDL updates, the changing of catalog indexes to NOT PADDED indexes, and converting some catalog data from EBCDIC to Unicode.

New V8 function is not available in V8 enable new function mode.

New Function Mode

→ New function mode (nfm)



f Available only after successful completion of the entire V8 enfm process

f New release function is now available

f New V8 Installed subsystems are in nfm

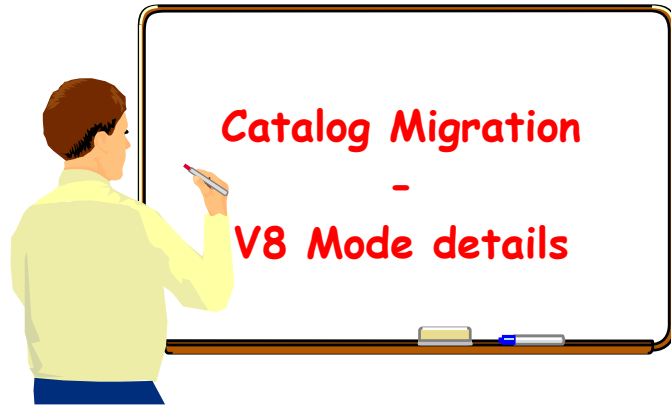
f Same V8 code level as in other modes

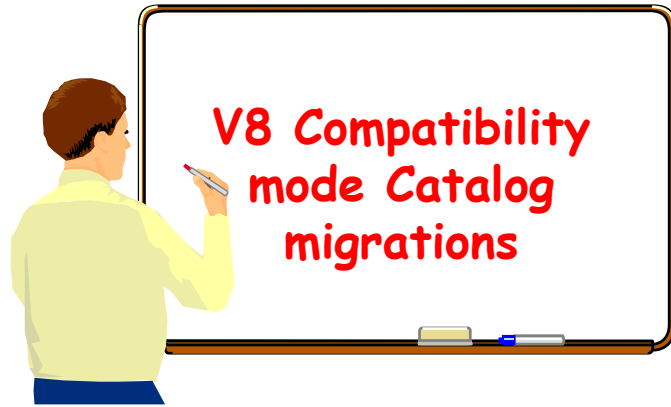
f **Group wide event**

© 2008 IBM Corporation

After the enable new function mode process completes, you can make the move to V8 new function mode (nfm) at any time. This can only be done after the successful completion of all the enable new function mode processing. Before new function mode can be entered, DB2 verifies that all the enfm processing has been completed.

New V8 function IS available in V8 new function mode.





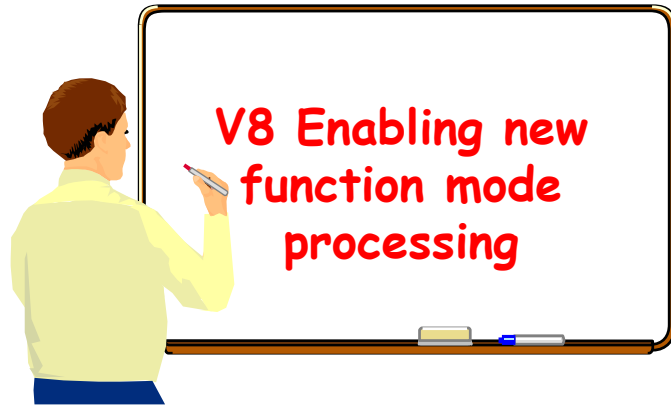
Compatibility mode Catalog migrations - what really happens?

- The **DSNTIJTC** migration process is not the 'black hole' that everyone thinks it is. Mandatory migration processing. Two steps in V8.
 - f Authorization check (INSTALL SYSADM) - DSNU760
 - f Verify catalog is at correct level for migration - this check varies with each DB2 release - DSNU765, DSNU766
 - f New release DDL
 - f Additional catalog updates and data checking
 - Look for unsupported objects - Type 1 indexes in V8
 - f Update directory header page and BSDS/SCA with new release information
 - f CCSID updates (step 2) in V8



© 2008 IBM Corporation

The migration process shouldn't scare anyone anymore. The process has been greatly improved during my tenure and hopefully this has been seen in the past several release migrations. In the coming charts I'll explain exactly what's being done during the migration process and what you can do to make sure that the process is actually running.



Enabling new function mode (enfm) - V8

→ The process by which the catalog and directory are readied for V8 new function mode (nfm) - **DSNTIJNE**

*f*Long names changes

→ Types and lengths of existing catalog columns are changed

→ **CHAR -> VARCHAR**

→ **VARCHAR(x) -> VARCHAR(y)**

→ **FOR BIT DATA**

*f*Catalog and directory data converted to Unicode

*f*Page size/buffer pool changes

*f*NOT PADDED catalog indexes



© 2008 IBM Corporation

During the enabling process, the catalog and directory are changed as necessary to ready DB2 V8 for new function mode. This entails a great deal of processing.

Enabling New Function Mode - V8

- Job named **DSNTIJNE**
- First step enters enabling new function mode (enfm)
 - f*No Fallback or Coexistence with V7
 - f*No returning to compatibility mode
 - f*Can't enter enfm if any member is at V7

- Several steps for each table space to be processed
 - f*Test for table space conversion - Long name DDL ALTERs - **ENFMnnn0**
 - f*Delete old data sets - **ENFMnnn1**
 - f*Define data sets needed for Online reorg - **ENFMnnn3**
 - f*Reorg the table space SHRLEVEL(REFERENCE) - **ENFMnnn7**
 - Record Formats - Page Sizes - Unicode - **NOT PADDED** indexes
 - Inline Image Copy
 - f*Delete old data sets - **ENFMnnn9**

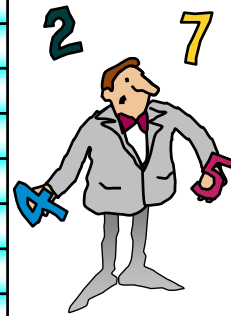
© 2008 IBM Corporation

The most important thing to note here is that you cannot go back to V8 compatibility mode from V8 enfm or V8 nfm. Once the enfm process starts there is no going back (without a complete system restore).

Enabling New Function Mode (cont.)

18 table spaces processed - order enforced!

SYSVIEWS	SYSOBJ
SYSDBASE	SYSPKAGE
SYSDBAUT	SYSPLAN
SYSDDF	SYSSEQ
SYSGPAUT	SYSSEQ2
SYSGROUP	SYSSTATS
SYSGRTNS	SYSSTR
SYSHIST	SYSUSER
SYSJAVA	SPT01



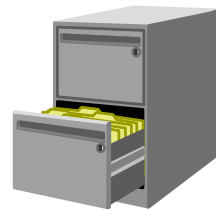
© 2008 IBM Corporation

These are the 18 table spaces that are processed during the enfm process. These are processed in the order specified. This order cannot be changed. It's an enforced order. If the DSNTIJNE job is modified to change the order then the enfm process will fail. We recommend that you do no attempt to modify anything in the DSNTIJNE job (other than space requirements).

SPT01 is the 1 directory table space that's processed during the enfm process.

Enabling New Function Mode (cont.)

- Can be stopped after long name DDL (ENFMnnn0) and reorg steps (ENFMnnn7) for each table space
*f*DSNTIJNH job
- Can be restarted without modification
 - Skips already processed table spaces
 - Resumes processing at first table space not successfully converted
- Insufficient space
 - f*Succeeding steps are skipped
 - f*-TERM UTIL issued to make table space available
 - f*Can change space parameters and restart
 - Skips already processed table spaces
 - Resumes processing at first table space not successfully converted



© 2008 IBM Corporation

The DSNTIJNH job can be used to nicely halt/stop the enabling new function mode process. Once submitted and executed, the enfm process will complete the current step that's processing and will then stop further enfm processing. This allows the current step to complete.

Enabling New Function Mode (cont.)

How many and which
table spaces converted
so far?

```
*** BEGIN DISPLAY OF GROUP(.....) GROUP LEVEL(...) MODE(E)  
GROUP ATTACH NAME(.....)
```

```
DB2          DB2 SYSTEM  IRLM  
MEMBER ID  SUBSYS  CMDPREF  STATUS  LVL NAME  SUBSYS IRLM  DB2  
.....  0 V81A  =   ACTIVE  810 ZS13PD  PR21  PRLMPR21
```

Indicates mode:
C, E, N

```
TABLE  ENABLED  
SPACE  NEW FUNCTION
```

```
SYSVIEWS  YES  
SYSDBASE  YES  
SYSDBAUT  YES  
SYSDDF    YES  
SYSGPAUT  YES  
SYSGROUP  YES  
SYSGRTNS  NO  
SYSHIST   NO  
SYSJAVA   NO  
SYSOBJ    NO  
SYSPKAGE  NO  
SYSPLAN   NO  
SYSSEQ    NO  
SYSSEQ2   NO  
SYSSTATS  NO  
SYSSTR    NO  
SYSUSER   NO  
SPT01     NO
```

```
*** END DISPLAY OF GROUP(.....)  
DSN90221 = DSN7GCMDB'DISPLAY GROUP' NORMAL COMPLETION
```

Use DISPLAY GROUP DETAIL
for both DS and non-DS!

© 2008 IBM Corporation

Use the DISPLAY GROUP DETAIL command in both data sharing and non-data sharing to determine the current version 8 mode of your subsystem or group. In data sharing, all group members will be in the same mode. Changing between modes is a group wide event.

Using DISPLAY GROUP DETAIL from any group member will therefore yield the same result.

A stylized illustration of a person with brown hair, wearing a light yellow long-sleeved shirt and blue pants, standing to the left of a whiteboard. The person is holding a red marker and appears to be writing on the board. The whiteboard has a dark brown border and a white surface. On the ledge of the whiteboard, there are two markers, one yellow and one blue. The text on the whiteboard is written in red, bold, sans-serif font.

**V8 New function
mode processing -
details**

New Function Mode

→ Job named **DSNTIJNF** enters new function mode (nfm)

→ No Fallback or Coexistence with V7

→ No returning to compatibility mode (cm)

*f*Returning would involve process outside of usual customer experience

-No change to the code base

• **Identical code in all modes - V8 cm, enfm, and nfm**

-Change to the catalog data and structure

• Recover to Point In Time prior to Enabling

• Recover Catalog and Directory to PIT

• Recover Customer data to PIT

→ Job named **DSNTIJEN** to return to enfm

*f*No change to the catalog data or structure



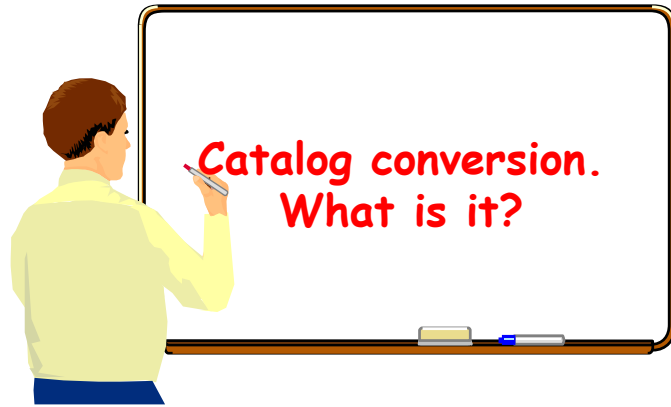
© 2008 IBM Corporation

To enter new function mode (nfm) you run the DSNTIJNF job. This job will take just an instant to run. It will only be successful if the entire enfm process had already completed successfully.

Once in nfm you cannot return to V8 cm.

You can return to the enfm mode as a means to prevent new V8 function from being used.

Doing this, via the DSNTIJEN job, will not undo any of the previously successful enfm processing. It will only prevent new function from being used. Objects that were created/alterd while in nfm using new V8 facilities are still available in enfm.



Unicode conversion of Catalog data

- During the enabling new function mode (enfm) process, some catalog data is converted from EBCDIC CCSID to Unicode CCSID

*f*18 table spaces converted to Unicode

- 17 catalog
- 1 directory

*f*Other catalog and directory table spaces still EBCDIC

- SYSCOPY, SYSEBCDC, SCT02, DBD01, SYSLGRNX, SYSUTILX

- ***User data is not converted to Unicode***

© 2008 IBM Corporation

The catalog and directory data is converted to Unicode during a SHRLEVEL REFERENCE reorg.

Unicode Catalog

- In compatibility mode (cm), all table spaces are EBCDIC
- In enabling new function mode (nfm), some are EBCDIC, some are Unicode

f-DISPLAY GROUP DETAIL

–Used for both data sharing and non-data sharing

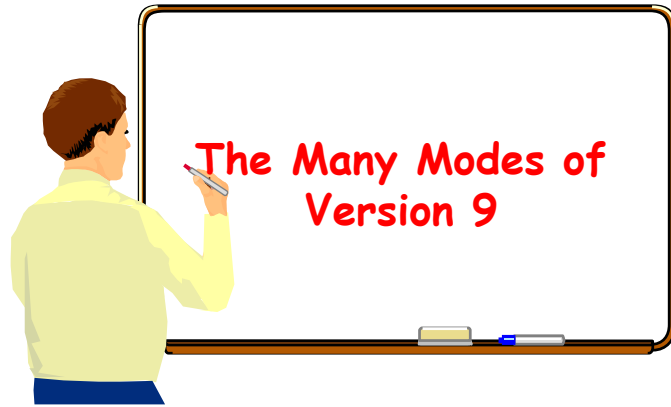
*f*SYSTABLES - ENCODING_SCHEME

*f*SYSTABLESPACE - ENCODING_SCHEME

- In new function mode (nfm), all 18 converted table spaces are Unicode, others are still EBCDIC (i.e. SYSCOPY)

© 2008 IBM Corporation

It's important to note that in nfm the entire catalog and directory is not Unicode. There are still some EBCDIC table spaces in both the catalog and directory in V8 nfm.



The Many Modes of V9

- DB2 introduces 2 new modes in V9 in addition to those added in V8

*f*Compatibility mode (V9 **cm**)

*f*Enabling new function mode (V9 **enfm**)

*f*New function mode (V9 **nfm**)

*f***Compatibility mode 'star'** (V9 **cm***)

*f***Enabling new function mode 'star'** (V9 **enfm***)

- Single code base for all V9 modes



© 2008 IBM Corporation

V8 introduces the notion of DB2 'modes'. There are three distinct modes in version 8: compatibility mode, enabling new function mode, and new function mode.

There is a single code base for all these V8 modes so you don't have to worry about adding any kind of maintenance between modes.

Compatibility Mode

→ Compatibility mode (**cm**) is essentially a traditional release migration

f Same migration process as in previous releases (ie **DSNTIJTC** job)

f Most new release function is not available

© 2008 IBM Corporation

In previous releases, a migration to a new release meant that new release function could be used immediately. It also meant that all 'must complete' migration processing had been completed.

In V8, this traditional migration process takes a DB2 subsystem from V7 to V8 compatibility mode. The DB2 catalog has been tailored and changed (although all the V8 catalog updates have not been made during this compatibility mode migration).

New V8 function is not available in V8 compatibility mode.

Enabling new function mode (enfm) - V9

- ➔ The process by which the catalog and directory are readied for V9 new function mode (nfm)
 - f*A much shorter process than in V8!
 - f*Page size/buffer pool changes (SYSOBJ)
 - f***DSNTIJEN** job (was named DSNTIJNE for V8!)
 - f*Copies the RTS from the user table spaces to new table spaces in the catalog.
 - f*Process cannot be halted as it could in V8
 - f*Most new release function is still not available



© 2008 IBM Corporation

During the enabling process, the catalog and directory are changed as necessary to ready DB2 V8 for new function mode. This entails a great deal of processing.

New Function Mode

- Job named **DSNTIJNF** enters new function mode (nfm)
- No Fallback or Coexistence with V8
- No returning to compatibility mode (cm) or enfm
 - **Identical code in all modes – V9 cm, enfm, nfm, cm*, enfm***
- Can return to compatibility mode 'star' (cm*)
- Can return to enabling new function mode 'star' (enfm*)



© 2008 IBM Corporation

To enter new function mode (nfm) you run the DSNTIJNF job. This job will take just an instant to run. It will only be successful if the entire enfm process had already completed successfully.

Once in nfm you cannot return to V8 cm.

You can return to the enfm mode as a means to prevent new V8 function from being used. Doing this, via the DSNTIJEN job, will not undo any of the previously successful enfm processing. It will only prevent new function from being used. Objects that were created/alterd while in nfm using new V8 facilities are still available in enfm.

Enabling new function mode 'star' (enfm*)

- ➔ Enabling new function mode 'star'
 - f* Indicates that the subsystem/group had previously been to V9 nfm
 - f* **DSNTIJES** job
 - f* No Fallback or Coexistence with V8
 - f* Can return to V9 cm* using **DSNTIJCS** job
 - f* Catalog changes are not undone when reverting to the 'star' modes – enfm* and cm*
 - f* New V9 function no longer available
 - f* Same code base as all other V9 modes



© 2008 IBM Corporation

During the enabling process, the catalog and directory are changed as necessary to ready DB2 V8 for new function mode. This entails a great deal of processing.

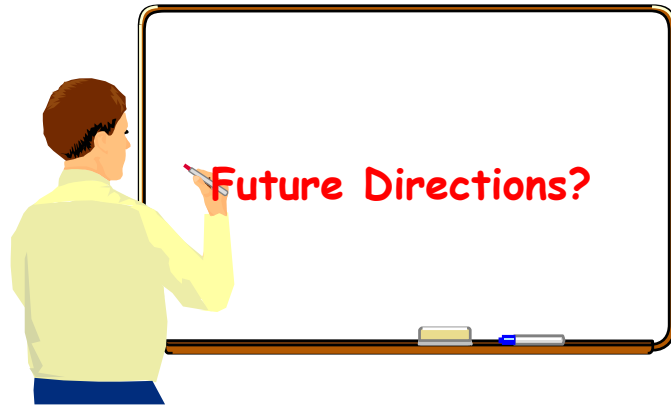
Compatibility mode 'star' (cm*)

- Compatibility mode 'star' (cm*)
 - f* Indicates that the subsystem/group had previously been to V9 nfm, V9 enfm, or V9 enfm*
 - f* **DSNTIJCS** job
 - f* No Fallback or Coexistence with V8
 - f* New V9 function no longer available
 - f* Same code base as all other V9 modes



© 2008 IBM Corporation

During the enabling process, the catalog and directory are changed as necessary to ready DB2 V8 for new function mode. This entails a great deal of processing.





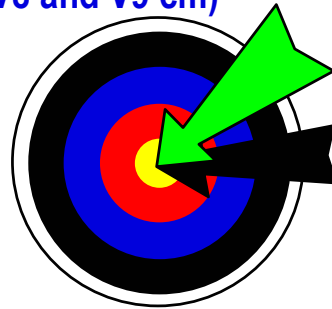
Migration Summary

→ Migration to V8 and V9 requires normal CATMAINT (DSNTIJTC) process

f Enters compatibility mode (V8 and V9 cm)

→ No new function (almost)

→ Precompile NEWFUN=NO



Migration summary

→ **Can only migrate to V8 from V7 and to V9 from V8**

*f*No Skip-Release Migration

→ **Data Sharing Coexistence only between V7 and V8 compatibility mode and v8 and V9 compatibility mode**



→ **Enabling New Function Mode is a Group-Wide event**

*f*Will not start with a previous DB2 release running anywhere in the group

*f*Previous release will not start after Enabling process has begun

–Regardless of whether or not Enabling process is running

–Now in Enabling New Function Mode (ENFM)

© 2008 IBM Corporation

The V8 ENFM process is a group wide event.

You do not have to apply any special 'mode' maintenance between V8 modes.

Skip release migration from V6 to V8 is NOT supported. You can only migrate to V8 from V7.

Migration Summary

- Changes to the Catalog are not optional
 - f* Everyone must run through the enabling new function mode (enfm) process
 - f* Will only be able to migrate to v-next from a Version's new function mode

- Enabling new function mode process is a single job (DSNTIJNE)
 - f* Several steps depending on the release
 - f* No Fallback to previous release
 - f* In V8, no returning to compatibility mode (cm)
 - f* In V9, can return to compatibility mode 'star' (cm*)

© 2008 IBM Corporation

You'll only be able to get to v-next from V8 nfm so you cannot skip the enfm process as a means to avoid the Unicode catalog conversion.

Migration Summary

- Upon enabling new function mode process completion, run job (DSNTIJNF) to enter new function mode
 - f*All new version function available
 - f*No Fallback to previous release
 - f*No returning to compatibility mode (cm)
 - f*Run DSNTIJNG job to rebuild DSNHDECP
 - Specify NEWFUN=YES in DSNHDECM

- f*You **CAN** return to enabling new function mode (enfm) in V8 with DSNTIJEN job and to enfm* in V9 with DSNTIJES job
 - Useful to prevent new function from being used. Catalog changes are not undone.
 - Run DSNTIJNG job to rebuild DSNHDECP
 - ☞ Specify NEWFUN=NO in DSNHDECM

© 2008 IBM Corporation

The DSNTIJNF job is used to change the mode from V8 enfm to V8 nfm. In V8 nfm, all new V8 function is available for use. This is really all it does. There's no additional processing that's done by the DSNTIJNF job.

Can return to V8 enfm from V8 nfm by running the DSNTIJNE job. No enfm processing is undone. Only the mode changes. This is a useful way to prevent new V8 function from being used.

Questions?



© 2008 IBM Corporation

Questions???

The DB2 Catalog: The Past, the Present, and the Future

John Lyle
IBM Silicon Valley Lab. (Etats Unis)

jlyle@us.ibm.com

