



IBM Software Expo 2006. Madrid 23 de Mayo

## ***Migrating to DB2 for z/OS Version 8***

### **The many modes of V8**



***Lola Salcedo***

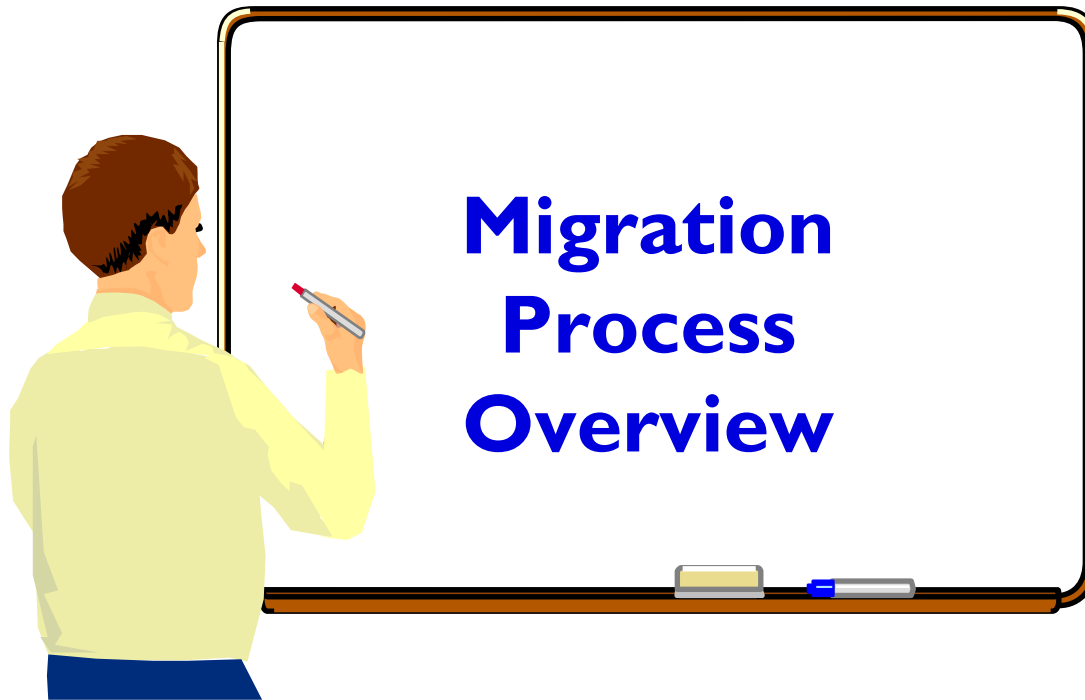
*Lola\_salcedo@es.ibm.com*



# Agenda

- **Migration Process Overview**
- **The Many Modes of Version 8**
- **Catalog Migration - V8 Style**
- **Catalog Conversion - What is it?**
- **Migration summary - tying it all together**





# Migration Terminology

- ➔ The process of upgrading from a release of DB2 to a more current release.
  - ▶ Can only migrate from DB2 V7 to V8. No skip release migration.
  
- ➔ Multi-step process
  
- ➔ Involves changing the DB2 catalog
  
- ➔ Migration process varies from release to release



## Prerequisites Summary



- zSeries Architecture (z800, z890, z900, z990)
- 64 bit exploitive
- z/OS 1.3 or above
  - ▶ Data Sharing improvements require z/OS 1.4 (with CFLevel 12)
  - ▶ System Level Backup / Recovery & Multilevel Security require z/OS 1.5
- Migrate from DB2 for z/OS and OS/390 V7
- Unicode Conversion Services enabled
- DSNHDECP customized for EBCIDIC, ASCII, and Unicode
- No Type1 Indexes
- Other prerequisites by feature
- Other assessments required
  - ▶ CCSIDs in V7
  - ▶ Language assessments



## Migration Process Overview

- More formalized than prior migration
- Migration occurs in three phases
- Only from DB2 for z/OS & OS/390 V7
  - ▶ Acquire and run DSNTIJP8 through the V7 maintenance stream and run against your V7 system early to discover items needing correction / change.
  - ▶ This is the same as DSNTIJPM delivered with V8.
- Reestablish the V7 IVP
- Assess ISV Requirements (like SAP)
- Training
  - ▶ Technical Conferences
  - ▶ IDUG
  - ▶ DB2 for z/OS V8 Migration Process Workshop

# Customary Migration Procedure

## ➔ Customary Migration Procedure

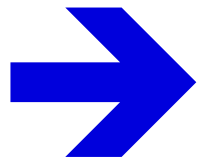
- ▶ Test Install and Migration on Test System
- ▶ Migrate Production to the new release
- ▶ Use no new function
- ▶ Verify compatibility of old function on new release
- ▶ **When satisfied with new release, begin to use new function**

## ➔ Problems for Fallback

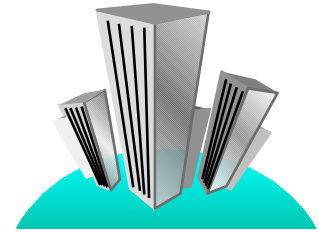
- ▶ No control over use of new function when verifying compatibility of old function on new release
- ▶ New function use usually not discovered until Fallback
- ▶ Use of new function sometimes precludes Fallback
- ▶ Fallback requires application of Fallback SPE
  - Usually overlooked during migration



# 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	987	59
V7	20	82	119	1206	105
V8	22	83	133	1265	105



# 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

- ▶ Catalog changes are the same
- ▶ Pre-migration activities almost identical



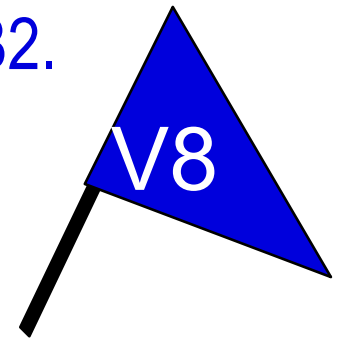
# Pre-migration considerations - Data sharing and non-data sharing

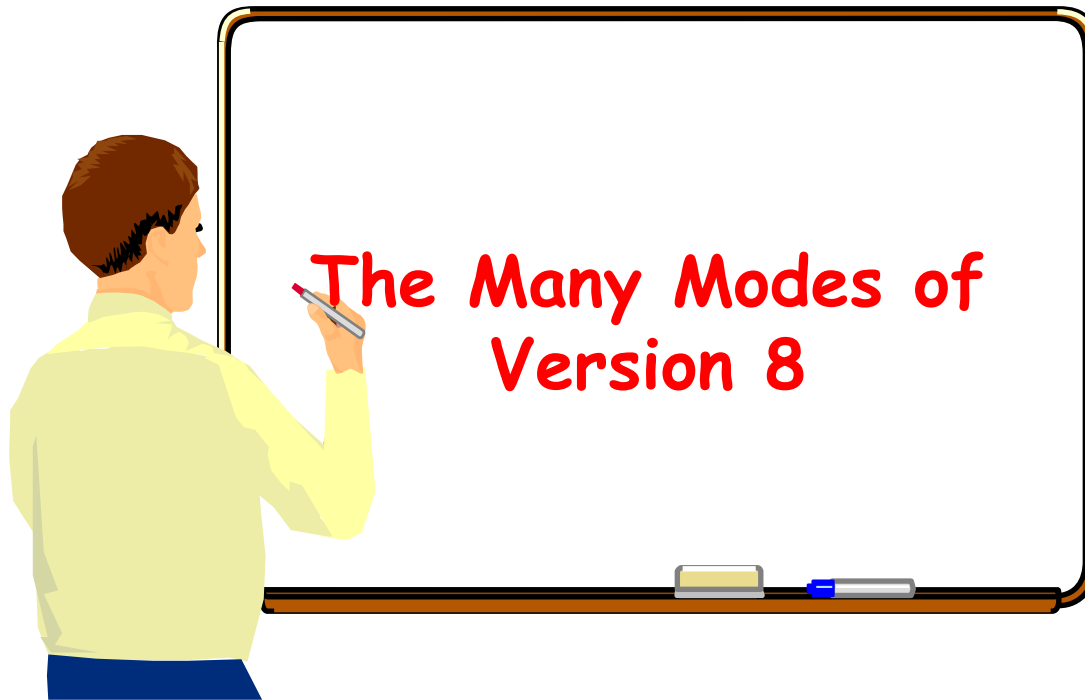
➔ It is **VERY** important to have the fallback SPE on before migrating to any new release of DB2.

▶ Now an enforced requirement!!!

▶ Fallback SPE (**PQ48486**) must have been applied to the migrating member on V7 and DB2 must have been

started before attempting to start DB2 V8. Otherwise DB2 V8 will not start (**DSNR045**).

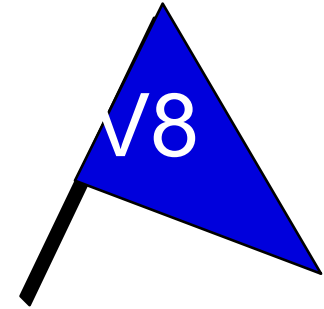




# The Many Modes of V8

➔ DB2 introduces 3 new modes in V8

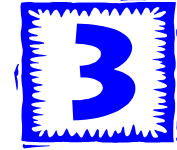
- ▶ Compatibility mode (V8 **cm**)
- ▶ Enabling new function mode (V8 **enfm**)
- ▶ New function mode (V8 **nfm**)



➔ Single code base for all V8 modes



# Migration Modes



Enable New Function Mode (ENFM)

New Function Mode (NFM)

Compatibility Mode (CM)

In CM you get:

- 64 bit
- Hiperpools / Dataspaces converted to primary bufferpools
- Bufferpool page fixing option
- New access paths
- New V8 Utilities (EXCEPT System Backup & Recovery)
- No new SQL function

- Fallback maintenance required to enter CM (and other prerequisites)
- Can last as long as necessary
- No new function available
- Run tests to ensure that no regression is taking place
- V7 and V8 Data Sharing coexistence supported
- Fallback to V7 is allowed
- Minimize time in CM with SAP

V8 Binaries/V7 Catalog

V7 changes to V8 Catalog

New V8 Feature

# Migration Modes



## Compatibility Mode (CM)

- Fallback maintenance is required to enter CM (and other prereqs)
- Can last as long as necessary
- No new function available
- Run tests to ensure that no regression is taking place
- V7 and V8 Data Sharing coexistence supported
- Fallback to V7 is allowed
- Minimize time in CM with SAP

## Enable New Function Mode (ENFM)

- Convert DB2 so that it is ready to support new functions
  - ▶ Series of ALTERS and ONLINE REORGS of the Catalog and Directory
  - ▶ Process can be halted and restarted
  - ▶ This is a group wide event - no V7 allowed even if ENFM not running
  - ▶ However, no fallback to DB2 V7 is allowed once this job is started

## New Function Mode (NFM)

- Administrator runs a job that triggers the ability to use new features of V8
- No Fallback allowed to V7
- No Returning to CM
- Can return to ENFM
  - ▶ No change to the catalog data or structure

V8 Binaries/V7 Catalog

V7 changes to V8 Catalog

New V8 Feature



# Compatibility Mode

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



- ▶ Same migration process as in previous releases
- ▶ Most new release function is not available
  - V8 exceptions? Online Reorg, 64 bit, ?



# Enabling New Function Mode

➔ Enabling new function mode (enfm)

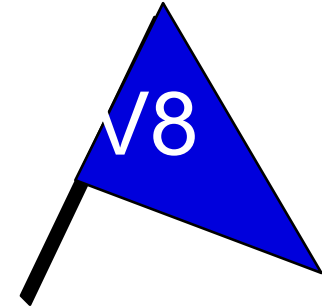
- ▶ In V8 this is a **conversion** process
  - Long name processing
  - Unicode conversion of catalog data
  - NOT PADDED catalog indexes (non-user)
- ▶ New release function is still not available
- ▶ Same V8 code level as in other modes
- ▶ **Group wide event**





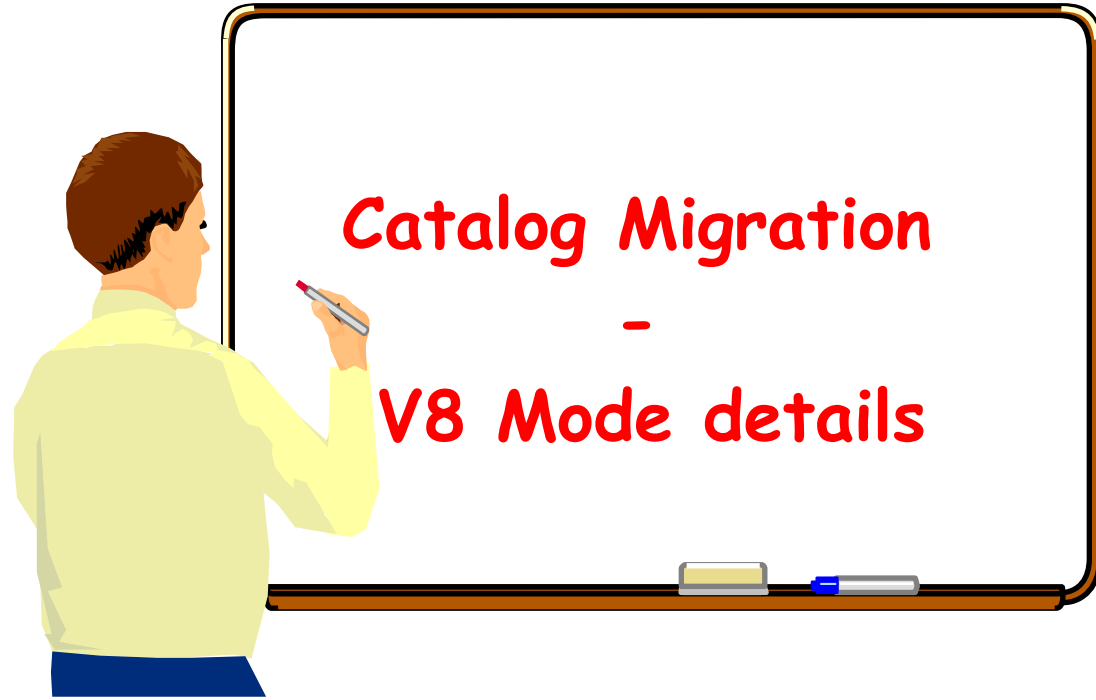
# New Function Mode

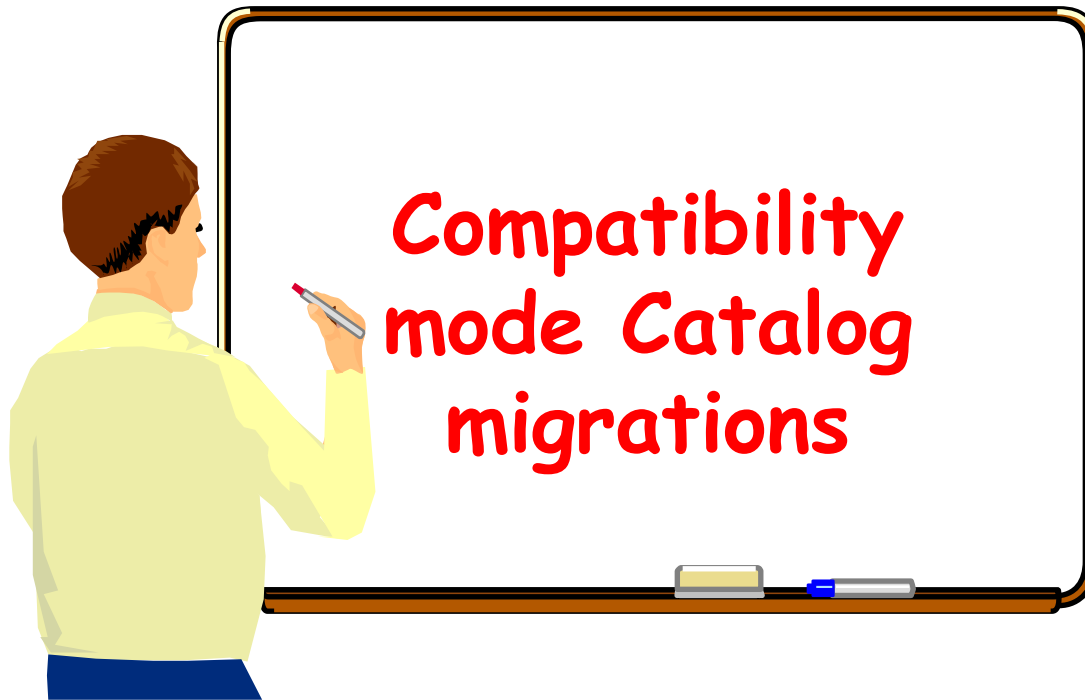
 New function mode (nfm)



- ▶ Available only after successful completion of the entire V8 enfm process
- ▶ New release function is now available
- ▶ New V8 Installed subsystems are in nfm
- ▶ Same V8 code level as in other modes
- ▶ **Group wide event**







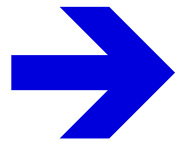
# Compatibility mode Catalog migrations - what really happens?

➔ The **DSNTIJTC** migration process is not the 'black hole' that everyone thinks it is. Mandatory migration processing. Two steps in V8.

- ▶ 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
  - ➔ Look for unsupported objects - Type 1 indexes in V8
- ▶ Update directory header page and BSDS/SCA with new release information
- ▶ CCSID updates (step 2)

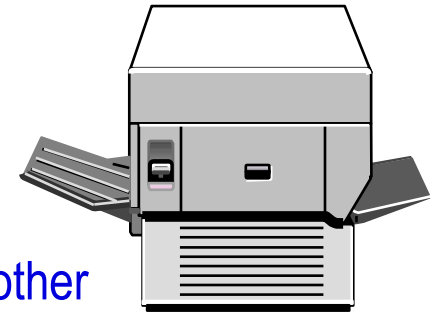


# Catalog migration - Additional updates made during migration



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



# Catalog migration - Additional data checking



Varies from release to release

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

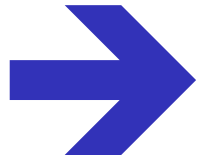
- ▶ Look for values or functions that are no longer allowed

- In V6 and V7, Type 1 indexes were tolerated but could not be used.

- In V8: Migration processing will terminate if you have any Type 1 indexes.



# Catalog migration - Is catmaint doing anything?



Can tell what catmaint is doing

- ▶ Look at SYSPRINT dataset for migration status messages (**DSNU777**)
  - These messages are issued at beginning of each catmaint step. These messages will vary from release to release.
- ▶ 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.



# Catalog migration - Using DISPLAY to check catmaint progress

➔ Information available varies from release to release

▶ SYSDBASE table space scan processing (V8)

```
.DISPLAY UTIL(*)  
DSNU105I . DSNUGDIS - USERID = SYSADM 217  
    MEMBER = V81A  
    UTILID = RELODCAT  
    PROCESSING UTILITY STATEMENT 1  
    UTILITY = CATMAINT  
    PHASE = CATMAINT  COUNT = 4287  
    STATUS = ACTIVE  
DSN9022I . DSNUGCCC '-DISPLAY UTIL' NORMAL  
COMPLETION
```





# V8 catmaint progress messages

 Information available varies from release to release

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. STATUS -**

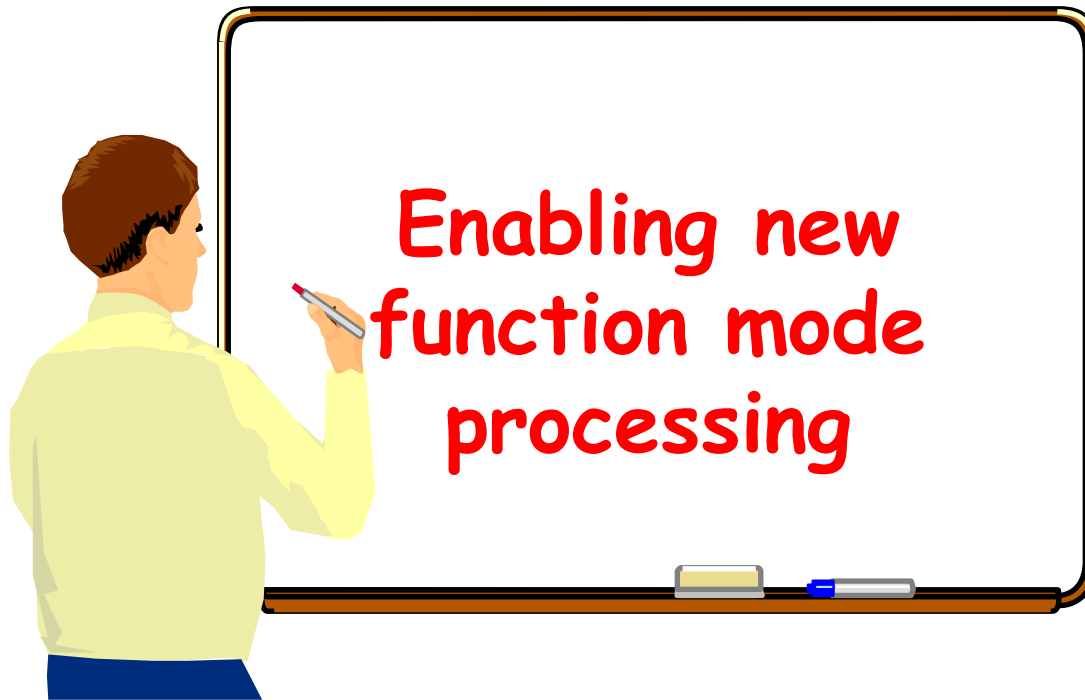
**DSNU777I DSNUECM0 - CATMAINT UPDATE STATUS - UPDATING DIRECTORY WITH NEW RELEASE MARKER**

DSNU752I DSNUECM0 - CATMAINT UPDATE PHASE 1 COMPLETED

**DSNU777I = DSNUECM5 - CATMAINT UPDATE STATUS - CCSID UPDATES COMPLETED.**

DSNU010I DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0





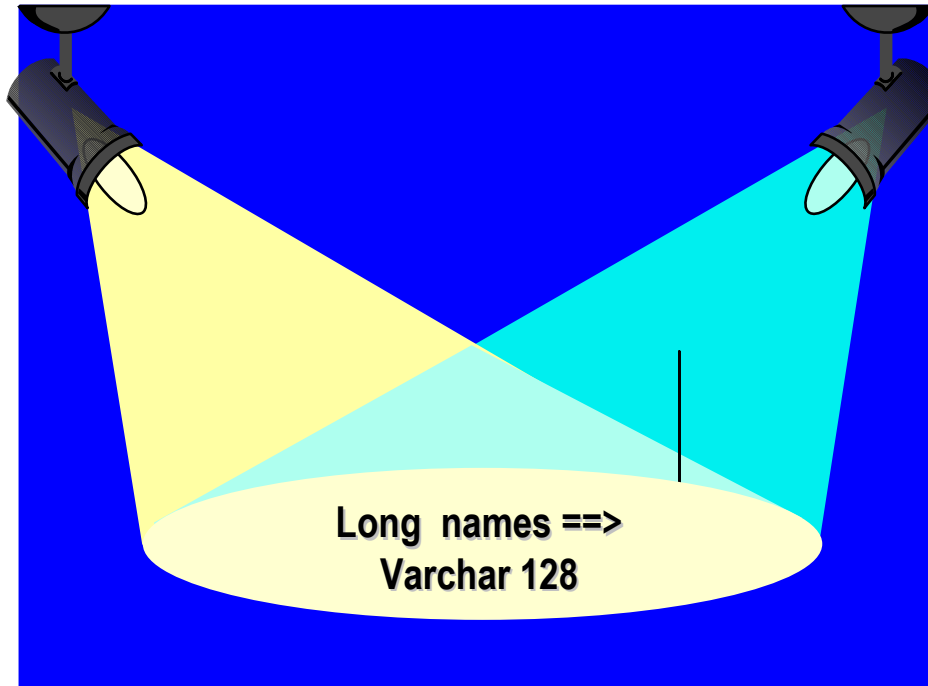
# Enabling new function mode (enfm)

- ➔ The process by which the catalog and directory are readied for V8 new function mode (nfm)
  - ▶ Long names changes
    - ➔ Types and lengths of existing catalog columns are changed
      - ➔ **CHAR -> VARCHAR**
      - ➔ **VARCHAR(x) -> VARCHAR(y)**
      - ➔ **FOR BIT DATA**
  - ▶ Catalog and directory data converted to Unicode
  - ▶ Page size/buffer pool changes
  - ▶ NOT PADDED catalog indexes

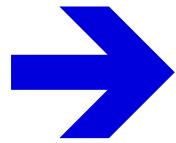


# Changes to the Catalog during enabling new function mode (enfm)

➔ Long names in the catalog



# Changes to the Catalog during enfm



Long Names ==> Varchar(128)

- ▶ **Table, View, Alias, ...**
- ▶ **Column (Varchar(30)), Data Type**
- ▶ **Schema, Authorization ID (Varchar(8))**
- ▶ **Routine: UDF, Stored Procedure, ...**
- ▶ **Trigger, Package**
- ▶ **...**

# Changes to the Catalog during enfm

- Long Names ==> Varchar(128)
  - Table, View, Alias, ...
  - Column
  - Schema, Authorization ID
  - Routine: UDF, Stored Procedure, ...
  - Trigger, Package...

## → Immediate Effects

- **Keys > 255**
- **Rows > 4056**
  - BP8K0 and BP16K0 (and BP32K)
- **18 Table Spaces processed**



# Changes to the Catalog during enfm

- ➔ Long Names ==> Varchar(128)
  - ▶ Table, View, Alias, ...
  - ▶ Column
  - ▶ Schema, Authorization ID
  - ▶ Routine: UDF, Stored Procedure, ...
  - ▶ Trigger, Package...
- ➔ Immediate Effects
  - ▶ Keys > 255
  - ▶ Rows > 4056
    - BP8K0 and BP16K0 (and BP32K)
  - ▶ 18 Table Spaces

## ➔ **UNICODE**

**Ordering - different from EBCDIC!**

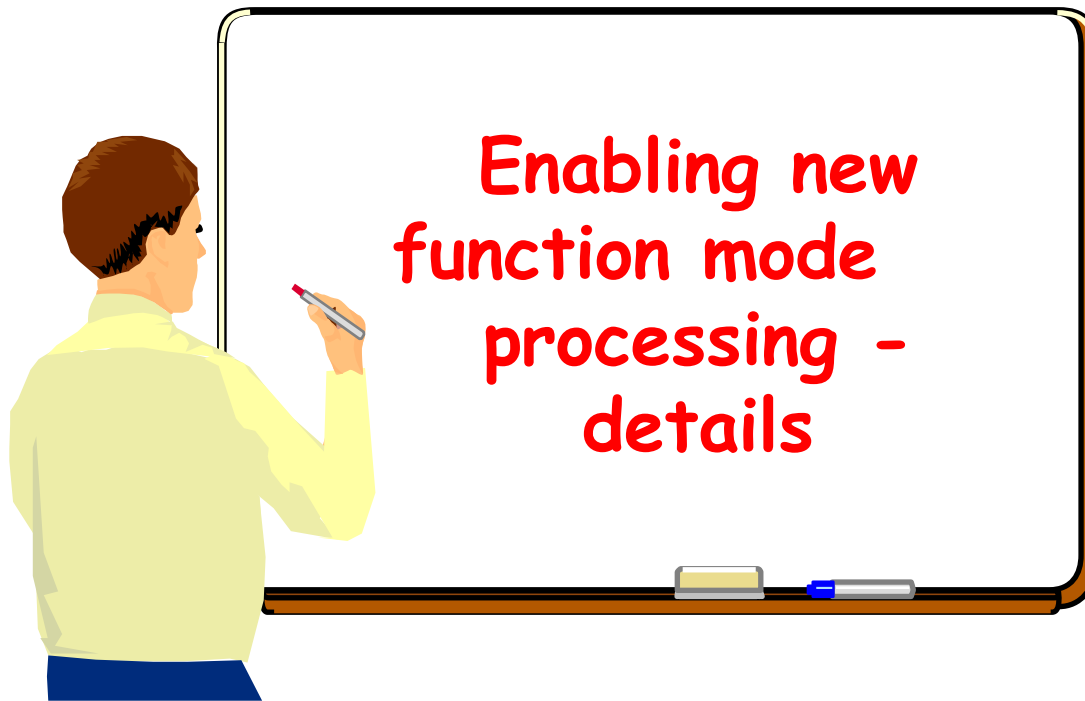


# Related Functional Enhancements

- ➔ Long Keys (255 => 2,000)
- ➔ Variable Length Keys (NOT PADDED)
  - ▶ Index-only access
- ➔ Long comparisons (255 => 32,704)
  - ▶ Sorting (4,000 => 16,000)
- ➔ Long SQL statements (32K => 2 Meg)
  - ▶ (Parse Tree Restructure)
- ➔ Multiple Encodings per SQL statement
- ➔ Online Reorg of all Catalog and Directory table spaces
- ➔ Unicode Parsing

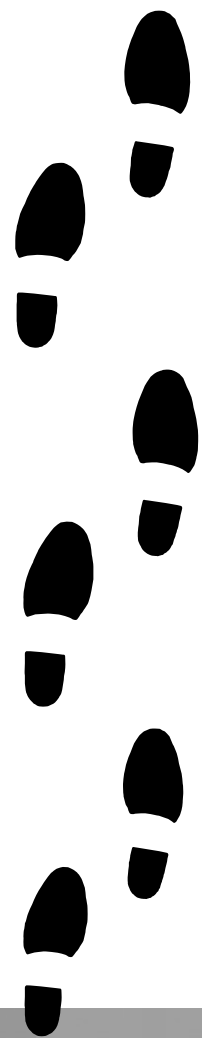






# Enabling New Function Mode

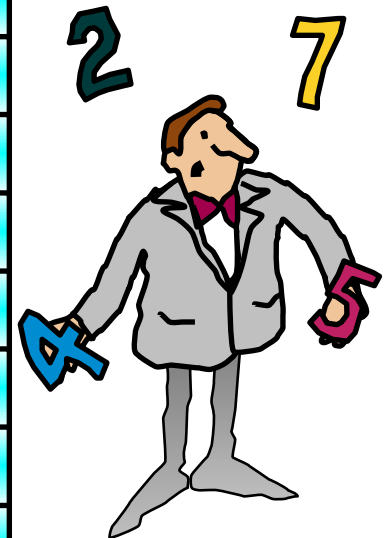
- ➔ Job named **DSNTIJNE**
- ➔ First step enters enabling new function mode (enfm)
  - ▶ No Fallback or Coexistence with V7
  - ▶ No returning to compatibility mode
  - ▶ Can't enter enfm if any member is at V7
  
- ➔ Several steps for each table space to be processed
  - ▶ Test for table space conversion - Long name DDL ALTERs - **ENFMnnn0**
  - ▶ Delete old data sets - **ENFMnnn1**
  - ▶ Define data sets needed for Online reorg - **ENFMnnn3**
  - ▶ Reorg the table space SHRLEVEL(REFERENCE) - **ENFMnnn7**
    - Record Formats - Page Sizes - Unicode - **NOT PADDED** indexes
    - Inline Image Copy
  - ▶ Delete old data sets - **ENFMnnn9**



# Enabling New Function Mode (cont.)

18 table spaces processed - order enforced!

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



# Enabling New Function Mode (cont.)

➔ Can be stopped after long name DDL (ENFMnnn0) and reorg steps (ENFMnnn7) for each table space

▶ **DSNTIJNH** job

➔ Can be restarted without modification

▶ Skips already processed table spaces

▶ Resumes processing at first table space not successfully converted

➔ **Insufficient space**

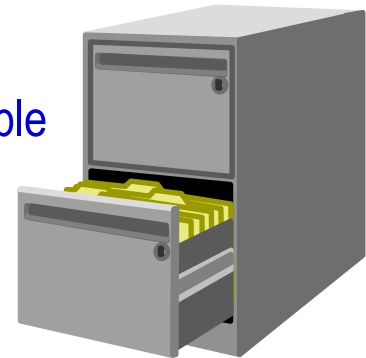
▶ Succeeding steps are skipped

▶ -TERM UTIL issued to make table space available

▶ Can change space parameters and restart

–Skips already processed table spaces

–Resumes processing at first table space not successfully converted

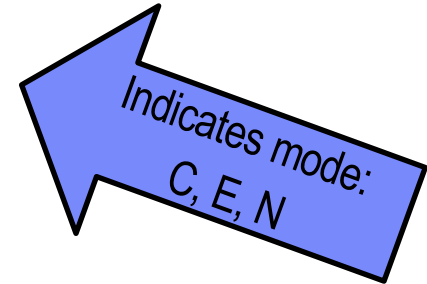


# Enabling New Function Mode (cont.)

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
IRLMPROC
-----
.....  0 V81A =    ACTIVE  810 ZS13PD  PR21  PRLMPR21
-----
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(.....)
DSN9022I = DSN7GCMD 'DISPLAY GROUP ' NORMAL COMPLETION
    
```



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

# Enabling new function mode (contd.)

## ➔ User defined indexes on the catalog

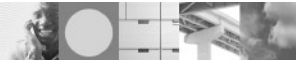
- ▶ Remain PADDED after completion of enfm process
- ▶ Can be converted to NOT PADDED in NFM
- ▶ User managed indexes require DSNTIJNE changes! Enfm process will fail if user defined index data sets are not added to the DSNTIJNE job.
  - Can drop indexes before enfm and recreate as NOT PADDED in nfm.
- ▶ DB2 managed indexes do not require DSNTIJNE modification. The enfm process will work.

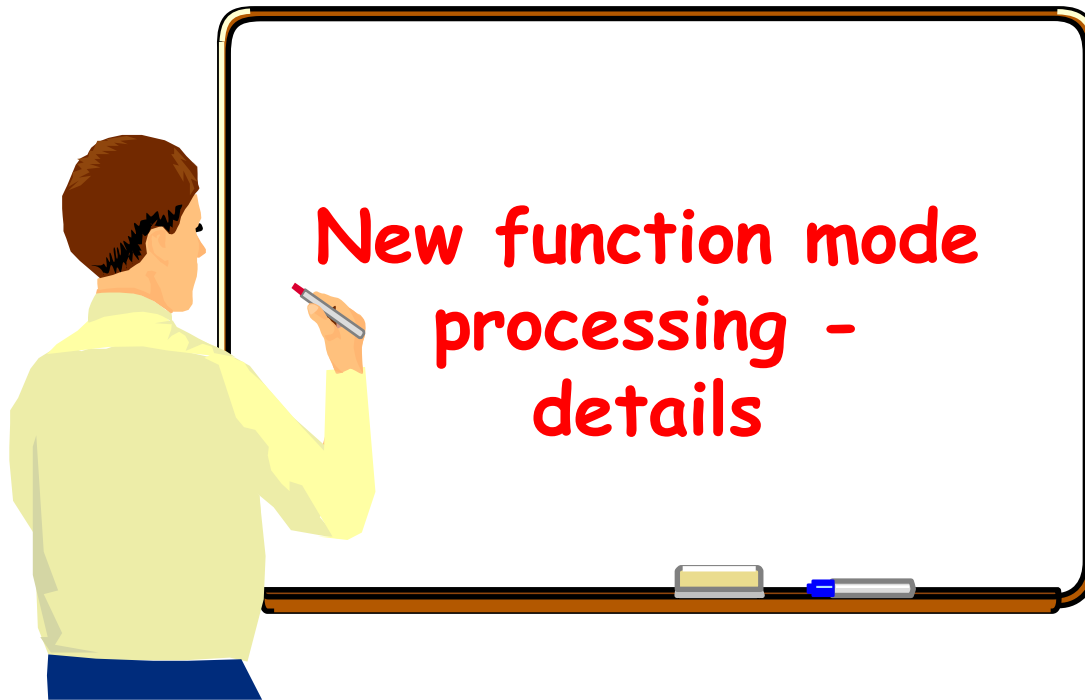


## Enabling new function mode (contd.)

### ➔ Application programs that reference the catalog

- ▶ Long name ALTERs will invalidate plans and packages
- ▶ Programs that reference objects in multiple catalog table spaces may have to be rebound multiple times
  - Only an issue if the enfm process is spread out over a period of time.







# New Function Mode

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

➔ No Fallback or Coexistence with V7

➔ No returning to compatibility mode (cm)

- ▶ 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

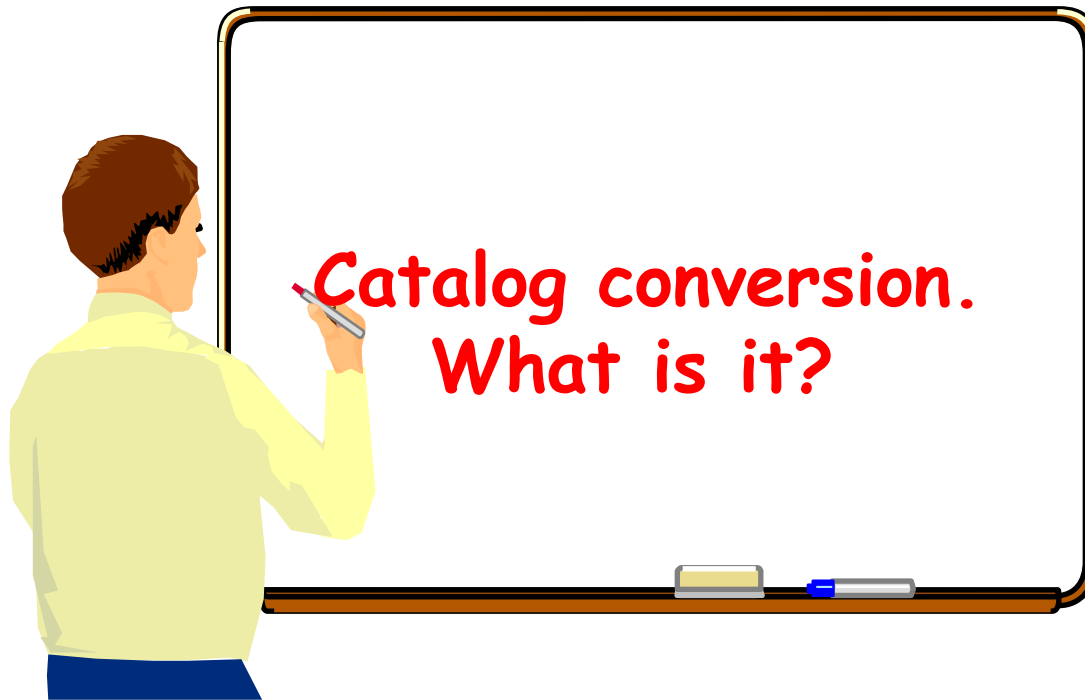
Recover to Point In Time prior to Enabling

- ♦ Recover Catalog and Directory to PIT
- ♦ Recover Customer data to PIT

➔ Job named **DSNTIJEN** to return to enfm

- ▶ No change to the catalog data or structure





## Unicode conversion of Catalog data

- ➔ During the enabling new function mode (enfm) process, some catalog data is converted from EBCDIC CCSID to Unicode CCSID
  - ▶ 18 table spaces converted to Unicode
    - 17 catalog
    - 1 directory
  - ▶ Other catalog and directory table spaces still EBCDIC
    - SYSCOPY, SYSEBCDC, SCT02, DBD01, SYSLGRNX, SYSUTILX
  
- ➔ *User data is not converted to Unicode*



# Unicode Catalog

- ➔ In compatibility mode (cm), all table spaces are EBCDIC
  
- ➔ In enabling new function mode (nfm), some are EBCDIC, some are Unicode
  - ▶ **-DISPLAY GROUP DETAIL**
    - Used for both data sharing and non-data sharing
  - ▶ SYSTABLES - ENCODING\_SCHEME
  - ▶ SYSTABLESPACE - ENCODING\_SCHEME
  
- ➔ In new function mode (nfm), all 18 converted table spaces are Unicode, others are still EBCDIC (i.e. SYSCOPY)



# Unicode Catalog

## ➔ Ordering

- ▶ ORDER BY
- ▶ GROUP BY
- ▶ Range Predicates



→, <, BETWEEN

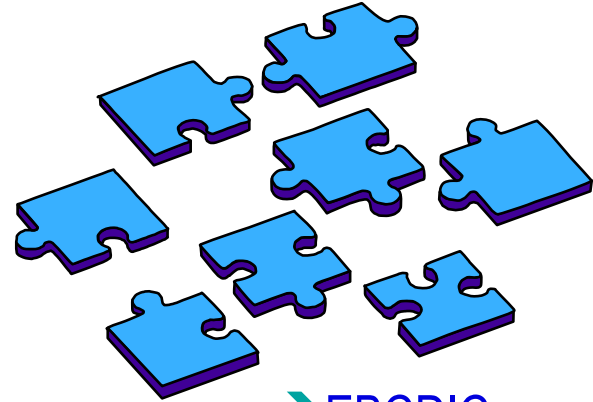
☒ > '5'

- ◆ EBCDIC - 'a' - 'z', 'A' - 'Z', '0' - '9'
- ◆ UNICODE - '0' - '9', 'A' - 'Z', 'a' - 'z'



# Unicode Catalog

## ➔ Order change example



### ▶ Catalog table names

EBCDIC order	Unicode Order
SYSCOLDIST	SYSCOLDIST
SYSCOLDIST_HIST	SYSCOLDISTSTATS
SYSCOLDISTSTATS	SYSCOLDIST_HIST
....	....
SYSTABLES	SYSTABLES
SYSTABLES_HIST	SYSTABLESPACE
SYSTABLESPACE	SYSTABLES_HIST
....	....

### ➔ EBCDIC

➔ '40'X - blank

➔ '6D'X = '\_'

➔ 'D7'X = 'P'

➔ 'E2'X = 'S'

### ➔ Unicode

➔ '20'X = blank

➔ '50'X = 'P'

➔ '53'X = 'S'

➔ '5F'X = '\_'

# Unicode Catalog

➔ Catalog growth during enfm conversion??

▶ Depends!

–Ever done a catalog/directory reorg?

–Migrated from a release prior to V5?

–CCSID used

☒ CCSIDs 37, 500 should see 1-1 conversion: one EBCDIC char converts to one Unicode char

☒ Some CCSIDs could see growth from conversion: one EBCDIC char could become several Unicode characters (i.e. @ in some EBCDIC CCSIDs)



# Unicode Catalog

➔ Catalog growth during enfm conversion??

▶ Enfm catalog changes. Example: SYSDBASE

Table Name	# CHAR to VARCHAR Columns	Added bytes per row	V8 CM min row length	V8 NFM min row length
SYSTABLESPACE	6	12	180	144
SYSTABLEPART	5	10	180	150
SYSTABLES	8	16	240	188
SYSCOLUMNS	4	8	148	124
SYSFIELDS	3	6	57	39
SYSSYNONYMS	3	6	53	35
SYSRELS	4	8	80	56
SYSFOREIGNKEYS	2	4	39	27
SYSTABAUTH	8	16	170	96
SYSCOLAUTH	5	10	123	75
SYSINDEXES	6	12	210	174
SYSINDEXPART	3	6	171	153
SYSKEYS	1	2	32	26
SYSLINKS	0	0	60	Deleted



# Unicode Catalog

➔ Catalog growth during enfm conversion??

▶ Catalog indexes?

–Will use less space when NOT PADDED

☒ Example: DSNDTX03 index on SYSTABLES

◆ Key: TBCREATOR, TBNAME

Padded length: 256 data bytes

NOT PADDED length: as few as 4 bytes

–Can convert back to PADDED.



# V8 Migration/Conversion Performance

➔ No internal measurements yet

➔ Migration performance V8 - customer actual

▶ Migration to V8 CM:

–'Last month, cloned our development catalog (27 GB) and it ran in 12.31 minutes...about 27.3 seconds per GB. Largest catalog at 32 GB should come in at 14.5 min.'

–'V7 CATMAINT clocked in at 1 min/GB, quite an improvement'

▶ V8 ENFM process:

–'Elapsed times for clone of Development(27 GB), 3 hrs. 11 min. (CPU 67.40 min) the last full reorg on that catalog about 2 years ago'





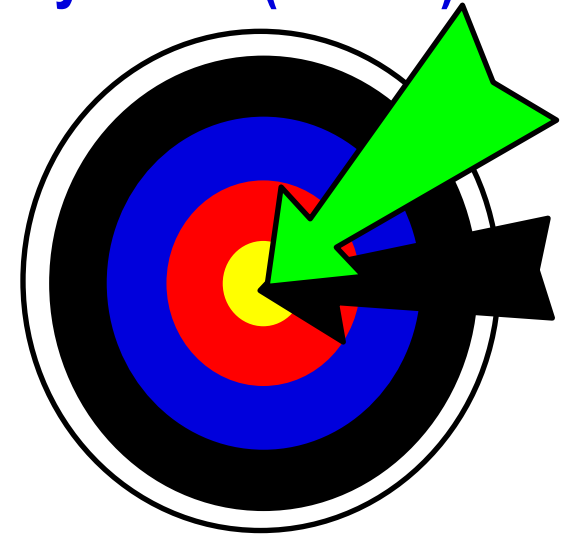
## Migration Summary

➔ Migration to V8 requires normal CATMAINT (DSNTIJTC) process

▶ Enters Version 8 compatibility mode (V8 cm)

➔ No new function (almost)

➔ Precompile NEWFUN=NO



## Migration to V8 summary

➔ **Can only migrate to V8 from V7**

- ▶ No Skip-Release Migration



➔ **Data Sharing Coexistence only between V7 and V8 Compatibility Mode**

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

- ▶ Will not start with a V7 running anywhere in the group
- ▶ V7 will not start after Enabling process has begun
  - Regardless of whether or not Enabling process is running
  - Now in Enabling New Function Mode (ENFM)

# Migration Summary

## ➔ Changes to the Catalog are not optional

- ▶ Everyone must run through the enabling new function mode (enfm) process
- ▶ Will only be able to migrate to v-next from Version 8 new function mode

## ➔ Enabling new function mode process is a single job (**DSNTIJNE**)

- ▶ Several steps for each table space to be processed
- ▶ Table space order processing enforced
- ▶ Online, SHRLEVEL(REFERENCE) Reorg of each table space
- ▶ Can be stopped after any table space
- ▶ Can be restarted without modification
  - Skips already processed table spaces
- ▶ No Fallback to V7
- ▶ No returning to V8 compatibility mode (cm)



# Migration Summary

- ➔ Upon enabling new function mode process completion, run job (**DSNTIJNF**) to enter new function mode
  - ◆ All new version function available
    - ▶ No Fallback to V7
    - ▶ No returning to V8 compatibility mode (cm)
    - ▶ Run DSNTIJNG job to rebuild DSNHDECP
      - Specify NEWFUN=YES in DSNHDECM
    - ▶ You **CAN** return to enabling new function mode (enfm) with the **DSNTIJEN** 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

¿Preguntas?



Gracias.





**John R Lyle**  
**IBM DB2 Development**  
**[jlyle@us.ibm.com](mailto:jlyle@us.ibm.com)**





# IBM Software Group Presentation Template and User Guidelines

- Available in Microsoft PowerPoint (.pot) and Lotus Freelance (.mas) formats
- Available in either black background (blueonyx) or white background (bluepearl) formats
  - ▶ Blue Pearl Template
    - Recommended for one-on-one laptop presentations
    - Web-based presentations
    - Report-style presentations
    - Easy black and white printing
  - ▶ Blue Onyx Template
    - Recommended for projected customer-facing presentations
    - Large audience presentations (keynotes, seminars)
    - Maximum contrast for high-level presentations
- See additional usage tips in Notes page below

