

IBM UK LTD
Product Introduction Centre
Data Management Team



DB2 UDB for z/OS Version 8

"The Migration Process"

Including Hints & Tips along the way

Ian Cook

DB2 Specialist

Email:- Ian_Cook@uk.ibm.com

Tele:- (Office) 0117-929 5049
(Mobile) 07802-458242

TABLE OF CONTENTS

INTRODUCTION	4
A NEW-TIERED APPROACH TO MIGRATION	5
WHY WE INTRODUCED THE NEW "TIERED" APPROACH?	5
ADDITIONAL USER BENEFITS	5
MODES OF THE TIERED MIGRATION PROCESS	7
DB2 VERSION 8 COMPATIBILITY MODE (CM)	7
DB2 VERSION 8 ENABLE-NEW-FUNCTION MODE (ENFM)	7
NEW FUNCTION MODE PROCESSING	8
SUMMARY OF TIERED MIGRATION PROCESS	9
CONSIDERATIONS ON MIGRATION AND ROLLOUT STRATEGY	9
COEXISTENCE AND FALLBACK	10
COEXISTENCE	10
FALLBACK	11
UNICODE CONVERSION SERVICES	12
WHAT DO I NEED TO DO TO SET UP UNICODE CONVERSION SERVICES?	12
INTRODUCTION TO SETTING UP UNICODE CONVERSION SERVICES (UCS)	13
PREREQUISITE UCS MAINTENANCE	13
SETTING UP UNICODE CONVERSION SERVICES	13
ADDITIONAL NOTES ON SETTING UP UNICODE CONVERSION SERVICES	15
DB2 VERSION 8 AND 64-BIT VSTOR SUPPORT	16
Z/OS 64-BIT VIRTUAL ADDRESSING SUPPORT	16
DB2 BENEFITS OF 64-BIT VSTOR SUPPORT	16
SO WHAT'S CHANGING? WHAT DO I NEED TO KNOW?	18
DB2 LIMIT CHANGES	18
INITIAL SIZING OF BUFFER POOLS	18
SIZING ALGORITHM DURING MIGRATION	19
ADDITIONAL BUFFER POOL INFORMATION	19
PLANNING FOR VSTOR EXPLOITATION	20
EDM POOL - DBD AND STATEMENT CACHING	21
ISV SERVICE FOR LRA (LOAD REAL ADDRESS) ISSUES	21
MIGRATION TO VERSION 8 COMPATIBILITY MODE	22
THE PROCESS	22
CATALOG OBJECT CHANGES	22
DSNHDECP CONSIDERATIONS	23
ENABLING NEW FUNCTION MODE	26
PREPARING FOR THE ENFM PROCESS	26
ENFM - WHEN SHOULD I RUN IT?	28
THE ENFM PROCESS	29

WHAT'S CHANGED NOW THAT I AM IN NFM?	35
LONG NAMES CONSIDERATIONS	35
IMMEDIATE EFFECTS	35
DSNZPARM PARAMETER CHANGES	36
REORG CONSIDERATIONS	37
SOME THOUGHTS ON INSTALLATION OF NEW DB2 SUBSYSTEMS	38
CATMAINT DURING INSTALL?	38
APPENDIX 1: SAMPLE UCS CONTROL STATEMENTS - GERMAN	39
APPENDIX 2: SAMPLE UCS CONTROL STATEMENTS - JAPAN	40
GLOSSARY OF TERMS	42

Introduction

IBM DB2 Universal Database for z/OS Version 8 represents a general purpose 64-bit exclusive database management system for large systems. The release breaks through many of the limits which have affected scalability and availability by delivering large address spaces with exploitation of 64-bit virtual addressing, providing management of more data with bigger buffers in memory.

DB2 for z/OS Version 8 also enhances distributed and dynamic applications through the provision of long SQL statement support and breaks barriers to family compatibility and application portability with long name support in the DB2 catalog.

Increased application availability with schema evolution support now permits DDL changes without stopping data access while the changes are installed. Data partitioned secondary indexes provides access to your data through more partitions, enhancing availability and minimizing partition management for historical data with support for rolling partitions.

In addition to the rich collection of new features and functionality previewed above, DB2 for z/OS Version 8 also introduces some significant changes to the DB2 catalog, with 18 of the 22 table spaces converting from EBCDIC to Unicode. During the conversion process, long name support (VARCHAR (128)) will also be introduced providing enhanced compatibility across the DB2 family.

As part of this release of DB2, a new multi-phased approach to migration will be introduced, to provide for a smooth transition from DB2 Version 7 through Version 8 compatibility mode and into Version 8 new function mode.

This paper is intended as an introduction to this new migration concept, detailing the new tiered approach and also highlighting useful information and considerations for a successful migration process.

As a prerequisite to this document and to performing an actual migration to DB2 Version 8, it is recommended that the user review the document entitled **"DB2 for z/OS Version 8 - Planning for Migration"**.

A new-tiered approach to migration

DB2 for z/OS Version 8 will introduce a new tiered (phased) approach to migration. Three phases will now represent the process of migrating from DB2 for z/OS and OS/390 Version 7 through to DB2 for z/OS Version 8 new function mode.

Importantly, the migration to DB2 for z/OS Version 8 is **only** supported from DB2 for z/OS and OS/390 Version 7. No other migration routes will be considered.

Why we introduced the new "tiered" approach?

The DB2 for z/OS Version 8 product introduces a significant set of new features and functions, including support for long names and Unicode. Thus the new "tiered" approach is primarily aimed at ensuring simplification of the transitional migration process and will also help to protect your systems during the migration process.

The aim is also to deliver enhanced client value by providing a tighter, more robust migration process, with the probability of fewer migration and fallback errors (e.g., forced SPE installation), making migration to a new release less of a challenge.

A tiered approach to migration will also provide you with increased control over the actual migration process. Following migration to DB2 Version 8 in compatibility mode (a business-as-usual process using CATMAINT), we now provide control over the migration to new function mode via the enable-new-function mode (ENFM) process. This process can be run in line with your system availability / maintenance window by scheduling the process over a number of planned sessions. You are now in control of exactly when you want to complete the ENFM process and set the switch for the subsystem to enter NFM, thus controlling the time for the introduction and use of the new functions.

Additional user benefits

Users have generally acquired a wealth of experience of performing DB2 migrations. A typical strategy might be to first test the install and migration procedure on a test system, followed by a rollout across strategic development environments. The process usually concludes with the migration of the production environment(s) to the new release.

Often during the migration process, there is no plan to test the new functions delivered with the new release. Instead, time is generally used to verify the compatibility of existing (old) functionality on the new code base, a process commonly termed "regression testing". When satisfied that existing applications and procedures work with the new code base, the user then begins to test and evaluate the new features and functions.

However, in reality, this process was always open to various problems. During the "regression testing" phase, various user groups often become eager to test / validate the new DB2 software and were often apt to experiment with the new functions.

The awareness that someone had experimented with the "new functions" was often not discovered until a fallback scenario actually took place. In fact, the undisclosed use of the new functions sometimes precluded fallback and customers were then forced into a fire-fighting approach in an endeavour to work through the problems. In addition, fallback required application of the fallback SPE, which was also sometimes bypassed during the migration planning phase.

So what controls does Version 8 provide?

DB2 for z/OS Version 8 has a mandatory requirement for the fallback SPE to be installed and activated on the Version 7 subsystem (or, if using DB2 data sharing, on every member of the group) prior to migration to Version 8 compatibility mode. Previous experiences have shown that the fallback SPE can often have a prerequisite chain of PTFs (depending on the current maintenance level of the DB2 system), demanding research and implementation overhead to the install of the actual fallback SPE. The new mandatory requirement for the fallback SPE introduced with Version 8 means that we can provide a fallback SPE that has a much smaller prerequisite chain.

Importantly, by forcing this work to take place under Version 7, you will eliminate the possible overhead of performing this task as part of a key fallback scenario, when minds generally tend to be focused on application and/or system recovery issues.

Another key benefit provided by this release is control over the point when you want to activate the new functions. Remember that new functions will **ONLY** be available once the DB2 subsystem is in new function mode and this point is represented by the successful running of the DSNTIJNF job.

Modes of the tiered migration process

This section briefly outlines the various modes of the tiered migration process.

DB2 Version 8 compatibility mode (CM)

The first phase of the tiered migration process which migrates the DB2 subsystem from Version 7 to Version 8 CM. Basically, it represents a business as usual release migration process as we have seen for previous releases, using the CATMAINT function. Completion of this phase places the DB2 subsystem into Version 8 CM, which represents a transitional phase, where no new functions are available that would preclude a fallback to Version 7, e.g., long names.

Some exceptions are:

- An online REORG of the entire DB2 catalog can now be performed as this functionality will be used during the actual ENFM process.
- The benefits of the 64-bit architecture are now available placing a number of subsystem pools and cache storage above the 2 GB bar, they include:
 - Buffer pools which can be page fixed for improved performance. (Page fixing to avoid performance regression is very important, and needs to be done early).
 - RID pool and SORT pool
 - Compression dictionaries
 - EDM pool for DBD cache and dynamic statement cache

Customers have the opportunity in DB2 Version 8 CM of ensuring that the new code is running stable in their test, key development and production environments, before moving forward through the ENFM process to NFM. While in DB2 Version 8 CM, fallback is, of course, permitted back to Version 7 in the event of a problem. Coexistence of DB2 Version 8 CM with DB2 Version 7 in a data sharing environment is also permitted.

DB2 Version 8 enable-new-function mode (ENFM)

Enable-new-function mode processing encompasses the 'additional' Version 8 conversion processing phase and consists of converting a DB2 subsystem or group from Version 8 CM to Version 8 new function mode. During the process, eighteen of the DB2 catalog and directory table spaces are converted from EBCDIC to Unicode (UTF-8) format and names are expanded to allow for the support of long names. The complete conversion period is performed by the DSNTIJNE job through a phase known as 'CATENFM'. For data sharing environments, the status of ENFM is a "group wide" condition.

Be aware that after commencement of DB2 Version 8 enabling-new-function mode there is no supported return to V8 compatibility mode and no fallback to Version 7. The only way to recover to a previous catalog level is by doing a point-in-time (PIT) restoration of the entire DB2 subsystem or group: logs, BSDSs, catalog, directory and associated user data and indexes. In data sharing, all the CF structures must be purged.

Catalog data during ENFM

Prior to new function mode, DB2 Version 8 will already be parsing data in Unicode (UTF-8) and during the enable-new-function mode process, most catalog table spaces will be converted to Unicode. In the enable-new-function mode, a catalog table space that has already been converted to Unicode will store its data in Unicode. A catalog table space that has not yet been converted to Unicode during the enable-new-function mode process will still store its data in EBCDIC.

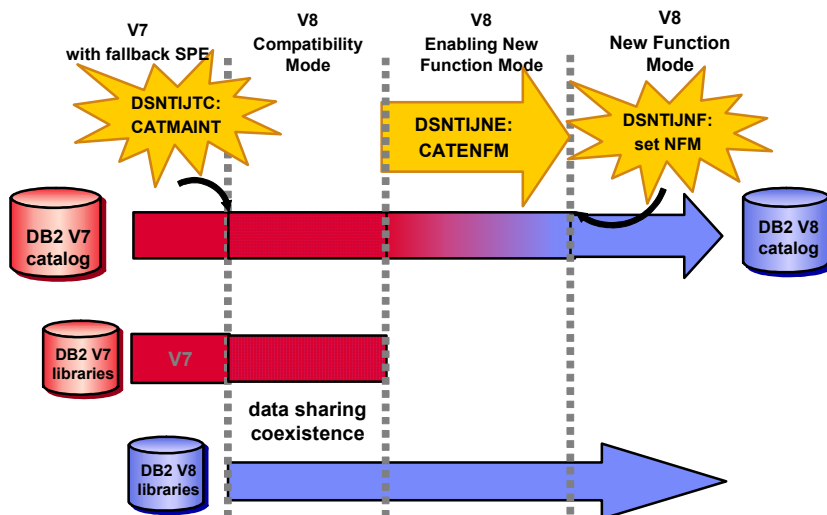
New function mode processing

Following the successful completion of the CATENFM phase, the user can then perform various inhouse test procedures in order to validate the environment, before running the DSNTIJNF job to set a switch in the DB2 subsystem or group which signifies that this environment is now in new function mode. This point signifies the end of DB2 Version 8 conversion processing and, from this point on, all new functions are now available for use. The DB2 catalog is now Unicode (UTF-8) and catalog columns have been expanded to allow for long names.

Summary of tiered migration process

The following diagram summarises the entire migration process from DB2 Version 7 through to Version 8 new function mode and highlights the jobs and processes deployed during the process. Note should also be made of the phases where we permit fallback, coexistence in a data sharing environment and the changes which take place to the DB2 catalog. Some reference notes are listed below to highlight these changes.

Overview of migration



Considerations on migration and rollout strategy

During early testing and evaluation of the tiered migration process, the following thoughts and considerations were suggested:

- Users may wish to consider migrating all their subsystems to Version 8 CM, before moving each forward into the ENFM phase. This allows the opportunity of maintaining a consistent code base throughout the various stages of migration. E.g., applying maintenance in a controlled fashion across the entire set of DB2 environments.
- With the controlled process above, users who have weekly, monthly and quarterly processing suites can ensure that each of these processes runs successfully with a stable DB2 Version 8 code base in CM before entering the ENFM phase.
- It is expected that users will remain in compatibility mode for some time, validating existing applications against DB2 Version 8 base code and preparing for the ENFM phase.
- Ensure you have good backup and restore procedures in place before beginning the ENFM process, as this is the **ONLY** way of returning to Version 8 CM.
- Do not confuse 'compatibility mode' with 'coexistence mode'. Please see the following section on coexistence mode for details of the requirements and options available.

Coexistence and fallback

Some coexistence and fallback considerations:

Coexistence

Coexistence can only take place between DB2 Version 7 and Version 8 CM subsystems within the same data sharing group. Importantly, when a Version 8 member starts, all currently running V7 members of the group, must have been started with the fallback SPE (PQ48486) applied. The process will not check the members that are currently not running, which includes members that just haven't started yet as well as members which may not have been started for a long while, perhaps even since a much earlier release of DB2 (quiesced members).

The important point is that when they do start, they must be running on V7 with the fallback SPE applied or V8. Once CATMAINT has started, only V7 systems that have the fallback SPE on will be allowed to start.

This requirement is enforced for both data sharing and non-data sharing environments. Information will be stored in the BSDS / SCA which will be checked at DB2 startup time to ensure that all group members have the SPE on. If a new member is added to the group, the starting member's code levels will also be checked to ensure the correct level with the associated DB2 catalog.

There is NO coexistence between an ENFM and/or NFM DB2 subsystems and once ENFM processing begins all active members of the group are considered to be in a status of ENFM. At this point you cannot start a Version 7 subsystem; however, you could start new a Version 8 member and it would join the group in ENFM status.

******* Remember the ENFM and NFM processing modes are group wide events *******

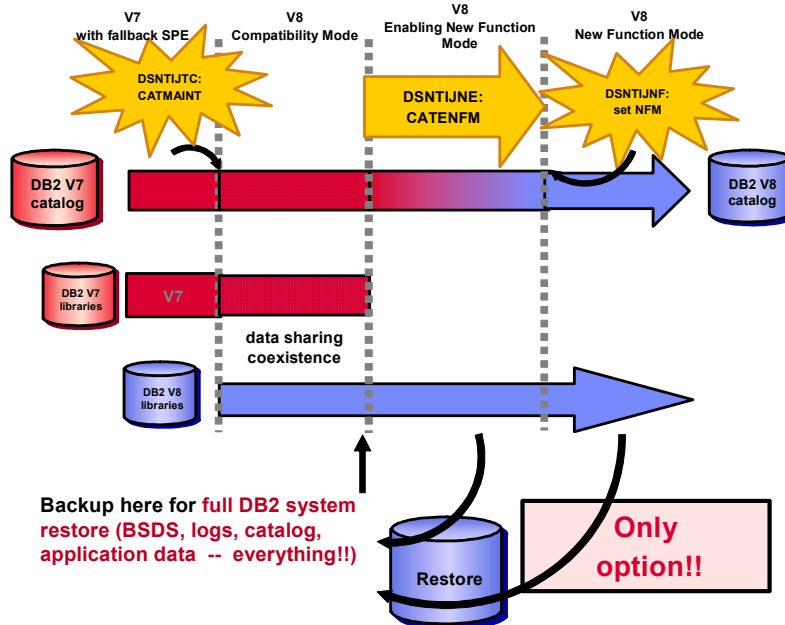
For data sharing, DB2 for z/OS supports n and $n-1$ releases of DB2 within a single DB2 data sharing group as defined in the "*DB2 for z/OS Version 8 Data Sharing: Planning and Administration Guide*". However, please note that DB2 Version 7 members can only coexist with DB2 Version 8 compatibility mode members. No DB2 Version 7 members can exist once you begin ENFM processing.

Distributed connections are supported to DB2 versions which are in service. For non-DB2 products, any level of DRDA is supported.

Fallback

Fallback to Version 7 is ONLY permitted from Version 8 CM and the procedure is the same as it is for previous releases of DB2. The fallback SPE must be on before attempting to start DB2 in a fallback release and the structure of the Version 8 catalog is used in Version 7 after fallback. The following diagram illustrates the fallback scenarios and please note that returning from ENFM or NFM mode to CM mode is ONLY possible using restore processing.

Overview of fallback



Unicode Conversion Services

DB2 for z/OS Version 8 requires the z/OS Unicode Conversion Services and appropriate conversion definitions to be set up for your environment. Even if your application user data is not stored as Unicode, you still need to set up these services as many of the DB2 catalog objects are now Unicode and all SQL parsing is now done in Unicode (UTF-8).

This is the case not only for V8 NFM (new function mode) but also for compatibility mode, because most of DB2 operates in Unicode even in CM.

z/OS Support for Unicode was first pre-installed into z/OS 1.2 and the process of defining UCS is detailed below.

For additional information on setup, it is recommended that the following Information APARs are reviewed and that you understand in detail, the coded character sets which you use:

- II13048
- II13049
- II13277
- II13695

What do I need to do to set up Unicode Conversion Services?

First, you must set up conversion tables in z/OS Unicode Conversion Services, providing a set of definitions which mark the total representation of the entire distributed and local environment associated with your DB2 Version 8 subsystem. Mappings must be included for every possible combination of code page translation, and these will differ, depending on your particular language settings. For example, if a work station inserts some data into a DB2 for z/OS system, the data is converted from ASCII to EBCDIC using a conversion table, which maps the code points from the source (ASCII) CCSID to the target (EBCDIC) CCSID. Below is a starter set of basic settings required to run in new function mode and the user must append their particular conversion requirements to this set. Many of the DB2-supplied Installation Verification Procedures (IVPs) specify the precompiler CCSID option, so a complete set of conversions to and from CCSIDs 37 and 1047 are needed to run the IVPs. (Please note that the set illustrated below contains entries for both code pages 37 and 500 as examples, and in a real situation unused code page entries would be removed):

```
CONVERSION:  00850-01047-ER 01047-00850-ER
              00037-01200(13488)-ER 01200(13488)-00037-ER
              00037-01208-ER 01208-00037-ER
              00037-00367-ER 01252-00037-ER
              00037-01252-ER 00367-00037-ER
              00500-01200(13488)-ER 01200(13488)-00500-ER
              01047-01200(13488)-ER 01200(13488)-01047-ER
              01208-01200-ER 01200-01208-ER
              01383-01200-ER 01200-01383-ER
              00932-01200-ER 01200-00932-ER
              00939-01200-ER 01200-00939-ER
              00300-01200(13488)-ER 01200(13488)-00300-ER
              00500-00850-ER 00850-00500-ER
              00500-01208-ER 01208-00500-ER
              00367-01208-ER 01208-00367-ER
```

Introduction to setting up Unicode Conversion Services (UCS)

Here we will summarise the steps necessary to set up UCS for DB2 for z/OS Version 8, based upon our Early Support Program experiences. Importantly, you need to define and customize z/OS Unicode support as described in *z/OS V1R3.0 Support for Unicode Using Conversion Services*, SA22-7649-01 (or the corresponding book for your level of z/OS). Instructions can also be found in the *DB2 Universal Database for z/OS Version 8 Installation Guide*, GC18-7418

Prerequisite UCS maintenance

The following APARs are required to be installed prior to running DB2 Version 8:

- OW56073 is the APAR that add 64-bits support for UCS. This APAR is closed and superseded by PTF UA01885 for z/OS 1.3 and UA01886 for z/OS 1.4
- OW56074 provides enhancements to UCS and DB2 will also require this APAR. The APAR is closed and superseded by PTF UA01925 and UA01926 for z/OS 1.3 and 1.4 respectively.
- z/OS 1.5 users will also require UA05790 for Unicode Conversion Services.

Setting up Unicode Conversion Services

The following steps outline the process of setting up UCS in preparation for DB2 for z/OS Version 8:

Generate the conversion table

The following JCL invokes utility program CUNMIUTL to generate a conversion image table (DD SYSIMG).

Please note that the ER option must be included on the CONVERSION command.

```
//CUNJIUTL JOB
//CUNMIUTL EXEC PGM=CUNMIUTL
//STEPLIB DD DSN=SYS1.LINKLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//TABIN DD DISP=SHR,DSN=SYS1.SCUNTB
//SYSIMG DD DSN=SYS1.MV50.IMAGES(CUNIMG01),DISP=SHR
//SYSIN DD *
/*****
*INPUT STATEMENTS FOR THE IMAGE GENERATOR *
*****/
CASE NORMAL; /*ENABLE TOUPPER AND TOLOWER */
CONVERSION 1140,367,ER; /*EBCDIC EURO ->UNICODE */
CONVERSION 367,1140,ER; /*UNICODE ->EBCDIC EURO */
. . . plus additional conversions . . .
/*
```

Copy conversion image to PARMLIB

Before activating the Unicode translation table, the generated conversion image (member from the SYSIMG DD in the JCL above) must first be copied into SYS1.PARMLIB as CUNIMGxx (where xx can be any valid suffix e.g. 01).

Create new PARMLIB member CUNUNIO1

Next create a new member in SYS1.PARMLIB called CUNUNIO1, the contents of which are listed below, with specifically the IMAGE parameter referencing the conversion image table produced earlier:

```

/**/
/*CUNUNIXX -UNICODE CONVERSION CONTROL PARAMETERS */
/**/
/*****/
/*ESTABLISH A NEW ENVIRONMENT */
/*****/
/*REQUIRED KEYWORD REALSTORAGE */
/*MAXIMAL USED PAGES OF REAL STORAGE,MIN=0 MAX=524287 */
/*WHERE 0 MEANS NO EXPLICITE LIMIT (=524287)*/
/*****/
REALSTORAGE 51200;/*E.G.200 MB */
/*****/
/*REQUIRED KEYWORD IMAGE WITH */
/*REQUIRED PARAMETER:MEMBER NAME */
/*THIS MEMBER MUST BE PLACED IN A DATA SET FROM THE */
/*LOGICAL PARMLIB CONCATENATION (DEF ' D IN LOADXX)*/
/*****/

```

Activate the Unicode table

Activate the new Unicode table with command T UNI=01. The SYSLOG output is displayed below:

```

T UNI=01
IEE252I MEMBER CUNUNIO1 FOUND IN SYS1.PARMLIB
537 CUN2036I INACTIVE CONVERSION ENVIRONMENT (UNICODE1)WILL BE
DELETED.ARE YOU SURE?(Y/N)
R 537,Y
IEE600I REPLY TO 537 IS;Y
CUN2038I INACTIVE CONVERSION ENVIRONMENT (UNICODE1)282
WAS SUCCESSFULLY DELETED
CUN2020I START LOADING CONVERSION IMAGE CUNIMG01
IEE252I MEMBER CUNIMG01 FOUND IN SYS1.PARMLIB
CUN2022I LOADING CONVERSION IMAGE CUNIMG01 FINISHED:401434 BYTES
LOADED
CUN2034I SET UNI COMMAND SUCCESSFULLY EXECUTED
IEE536I UNI VALUE 01 NOW IN EFFECT

```

Verify the Unicode table facilitation

Verify the Unicode table activation with command D UNI,ALL. The SYSLOG output is shown below:

```

D UNI,ALL
IEF196I IEF237I 1A20 ALLOCATED TO SYS00110
IEF196I IEF285I SYS1.LINKLIB KEPT
IEF196I IEF285I VOL SER NOS=P13D02.
CUN3000I 14.26.27 UNI DISPLAY 303
ENVIRONMENT:CREATED 04/10/2003 AT 10.02.01
MODIFIED 04/10/2003 AT 14.24.25
IMAGE CREATED 04/10/2003 AT 11.35.12
SERVICE:CUNMCNV CUNMCASE
STORAGE:ACTIVE 98 PAGES
INACTIVE 98 PAGES SINCE 04/10/2003 AT 11.37.23
LIMIT 51200 PAGES
CASECONV:NORMAL
CONVERSION:01140-00367-ER 00367-01140-ER

```

Additional notes on setting up Unicode Conversion Services

- An IPL is required for the first activation of Conversion Services.
- Dynamic alteration of conversion images is controlled by SET UNI=xx

DB2 Version 8 and 64-bit VSTOR support

Before we embark on a detailed discussion about the various phases of migration, let's recap on 64-bit virtual storage support and how DB2 Version 8 takes advantage of it.

z/OS 64-bit virtual addressing support

64-bit virtual storage now provides applications and middleware the option of storing and manipulating data above the 2 GB bar and was introduced with z/OS 1.2. For a comprehensive introduction to the architecture please review the *z/OS 64-bit Virtual Storage Roadmap*, GM13-0076, at:

<http://www.ibm.com/servers/eserver/zseries/library/whitepapers/gm130076.html>

DB2 64-bit virtual exploitation in DB2 for z/OS Version 8 requires at minimum z/OS 1.3, with the architecture providing a number of important enhancements including:

- Enhanced performance with enhanced data caching in memory
- Improvements in vertical scalability resulting from virtual storage constraint relief.
- Increased maximum buffer pool sizes and the elimination of hiperpools and data spaces, helping to simplify DB2 systems management and operational tasks.

Please note, the maximum configuration available is, of course, dependent upon the central storage available on the processor and close monitoring of paging must be planned to ensure sufficient auxilliary storage availability.

A technical bulletin on setting up and monitoring a 64-bit environment "z/OS Performance: Managing Processor Storage in a 64-bit Environment" can be found at the following URL:

<http://www.ibm.com/support/techdocs>

DB2 benefits of 64-bit VSTOR support

VSTOR relief utilises 64-bit virtual addressing to move a number of data areas above the 2 GB bar in the DBM1 address space, including the buffer pools, page manipulation blocks and castout buffers. DB2's "data access" modules are enhanced to access 64-bit addressable buffers "in place" which means no internal data movement as per data space buffer pools today.

Additional DBM1 data areas now resident above the 2 GB bar are sort pools, RID pools, compression dictionaries and DBDs. The limits on virtual buffer pools have increased

significantly, however, settings are of course dependent upon the amount of real storage available on the processor.

With the advent of moving more data pools above the 2 GB bar, the releasing of storage below will mean more available room for 31-bit application threads. However, care must be taken when increasing the concurrent thread parameters as they must also be backed by enough real memory on the processor.

Page fixing for performance

ALTER BUFFERPOOL has a new option that most users should use for subsystems which read or write frequently. Recommendation: alter your DB2 Version 8 buffer pools which have frequent page reads or writes to use PGFIX(YES) if you have sufficient real storage available for these buffer pools. Fixing the buffer page once and keeping it fixed in real storage avoids the processing time that DB2 needs to fix and free pages each time there is an I/O. In some cases, this processing time can be as much as 10% for I/O-intensive workloads. To use this option, issue the following command:

```
ALTER BPOOL(bpname) VPSIZE(vpsize) PGFIX(YES)
```

where bpname is the name of the buffer pool and vpsize is the size of the virtual pool.

IRLM considerations

The mandatory requirement for IRLM V2.2 means that locks are now resident above the 2 GB bar and "PC=YES" is forced, housing all locks in the extended private area above the 2 GB bar (IRLM no longer supports placing locks in ECSA). Actually, the PC=NO recommendation was based on earlier systems where the PC instruction was very expensive. In recent systems the performance difference between PC=YES and PC=NO is actually less than 2%, and of course, using PC=YES moves the locks out of common into IRLM private thus freeing all that space for use in other address spaces.

The PC and MAXCSA parameters are no longer used, but you must maintain them for compatibility reasons, although the parameters and values set for these are not used.

IRLM will reserve approximately 10% of the available private storage for must-complete lock requests. Use the MODIFY irlmproc,STATUS,STOR command to view and monitor the amount of private storage that IRLM has available. You can adjust the amount of private storage dynamically with the MODIFY irlmproc SET,PVT command. The new value remains in effect until the next time IRLM is stopped and restarted or until the MODIFY command is issued successfully again.

So what's changing? What do I need to know?

DB2 for z/OS Version 8 buffer pools will always be allocated above the 2 GB bar and it is worth noting a couple of changes in "terminology". As of V8, the terms "buffer pool" and "virtual pool" become synonymous and data space pools and hiperpools will no longer exist, as they are not supported.

The maximum size of a buffer pool OR the total size of all buffer pools, will now be 1TB, although the actual maximum will equal the REAL storage available on the processor.

In case you need to fallback to Version 7, all buffer pool information will be saved internally.

DB2 limit changes

The maximum number of read, write, castout engines is increased to 600 and DB2 caps virtual storage above the bar at 4 TB. Additional enhancements include:

- BPO minimum size increased from 56 to 2000 buffers with default 20000
- BP8KO default now 1000 (was 0)
- BP16KO default now 500 (was 0)
- BP32K default now 250 (was 24)
- CTHREAD limit now increased from 2000. N.b., your setting for this parameter is dependent upon the REAL storage availability on the processor. ECSA allocation may also need increasing if CTHREAD is raised.

Initial sizing of buffer pools

It is very important not to over-allocate your buffer pools relative to the available central storage you have in the LPAR. Message DSNB536I will indicate when your total VSTOR BPOOL requirement is greater than the available REAL capacity of the z/OS image. DB2 will actually limit the allocation of buffer pools, if the aggregated (allocated) pool size is greater than twice the REAL storage size of your system image. If this happens, message DSNB610I will indicate a reduction to the default pool size for your allocation, where the default is 2000 pages for 4K, 1000 pages for 8K, 500 pages for 16K and 250 pages for 32K.

Sizing algorithm during migration

So, what happens during migration? Well it depends upon what mix of virtual pools and hiper pools you have in Version 7 prior to migration:

- If you have data spaces and virtual pools but NO Hiperpools, then the VPSIZE is used to define the new Version 8 pool size.
- If you have virtual pools with a corresponding hiperpool then DB2 adds (VPSIZE + HPSIZE) to calculate your new Version 8 pool size.
- VPSEQT, VPPSEQT and VPXSEQT keep their sizes even if buffer pool size is determined by VPSIZE + HPSIZE
- DB2 V8 maintains original virtual pool and hiperpool sizes for fallback
- No change for new Version 8 installations, with the values taken from the install process

In general, the pool sizes you have in Version 7 should be the sizes you have after migration, except for the changes in the minimum sizes of BPO and BP32K. These begin to take effect during V8 CM. BP8KO and BP16KO are required at the point in ENFM when the first catalog table space is converted to those page sizes.

Additional buffer pool information

Well, is there anything else I should know about buffer pools in Version 8?

- Buffer pool attributes can still be changed via ALTER BUFFERPOOL command
- The names remain unchanged, e.g., BPO, BP1
- Pages size options remain at 4K, 8K, 16K and 32K
- Here are some ALTER BUFFERPOOL command parameters which are no longer supported. If they are specified, *only* a warning message is issued:
 - VPTYPE, HPSIZE, HPSEQT
 - CASTOUT
- Parameters remaining unchanged:
 - VPSEQT, VPPSEQT, VPXPSEQT
 - DWQT, VDWQT
 - PGSTEAL
- LSTATS report removes references to hiperpool related counters on DIS BUFFERPOOL.

Planning for VSTOR exploitation

A few things to consider when planning for DB2 Version 8 and 64-bit VSTOR exploitation:

- Start by running a DB2 for OS/390 Version 7 subsystem under a 64-bit virtual O/S
- Perhaps set reasonably low values for buffer pools, EDM pool and thread limits for your first DB2 for z/OS Version 8 invocation.
- Remember, DB2 for z/OS Version 8 is an exclusive 64-bit virtual release of the DB2 product. Hence, the hardware prerequisite of a zSeries (or equivalent) processor running z/OS 1.3 (minimum) in 64-bit mode is mandatory for all DB2 for z/OS Version 8 subsystems / members. This means that when rolling out DB2 Version 8 through test, development and production environments, each must satisfy the prerequisite criteria.
- The same prerequisites must also apply for your disaster recovery environments, with your secondary sites also configured to support 64-bit virtual processing requirements.
- If your DB2 for z/OS Version 8 subsystem or member runs in a Parallel Sysplex environment, please be aware of the 64-bit virtual implications of "cross system restart".
- When estimating storage, remember that DB2 Version 8 now allocates space above 2 GB bar in the DBM1 address space. This will subsequently free up a significant amount of storage in the 31-bit addressable area, hence you may wish to change parameters to take advantage of this, by perhaps permitting more concurrent threads (CTHREAD). Importantly, you must ensure you have the central storage available to back an increase in concurrent thread processing to avoid problems. Increasing concurrency will of course lead to a higher transaction throughput, which should be continually monitored.
- The new MEMLIMIT JCL keyword provided by z/OS 1.2 controls how much virtual storage above the 2 GB bar is available in each address space. By default, DB2 sets the MEMLIMIT value to 4 TB in field **STG AVAILABLE ABOVE 2GB** on the DSNTIPC panel, ensuring sufficient memory for operation.
- Running DB2 Version 8 may be your first experience of a 64-bit virtual environment and, in the early days of testing, you should ensure that your dump and diagnostic procedures are in place in the event of problems. Specifically, you should ensure that you have fully tested your stand alone dump procedures and that you have installed the z/OS Version of SAD with the "High Virtual Option".
- When increasing thread limits (CTHREAD), track and monitor usage of ECSA and increase accordingly. Remember that ECSA which was formerly used by IRLM, is now freed up.
- MVS runs out of auxiliary storage slots at 4 TB

EDM Pool - DBD and Statement Caching

A few thoughts on changes to EDM Pool caching:

- The use of a data space for dynamic caching will be deprecated. A new cache will be created above the 2 GB bar.
 - Today if "cache dynamic" is on, statements are cached in the data space EDM pool
 - Now "cache dynamic" statements ALWAYS cached in the dynamic statement cache pool above 2 GB bar
- A new EDM DBD cache will be created above the 2 GB which will give DBDs the needed space to grow and thus relieve contention with other objects

There are some changes to EDM pool definitions worth noting on the install panel DSNTIPC:

- Change of definition EDMPOOL DATASPACE SIZE to EDMPOOL DBD CACHE SIZE (above 2G bar)
- Change of definition EDMPOOL DATA SPACE MAX to the EDM Pool Statement cache size (above 2 GB bar)
- No changes to estimation and defaults of EDMPOOL storage except that instead of dataspace, it will be storage above 2G bar

ISV service for LRA (Load Real Address) issues

Its worth watching for products or applications that may experience problems as a result of dependencies on LOAD REAL ADDRESS (LRA) programming requirements. These are rarely used by customer application programmers, but can typically be found in middleware products, such as performance reporting tools or capacity planning tools, etc. Tools that need to monitor and report on the additional storage architecture are candidates, with most problems having something to do with LRA - OD3 abends. You may have programs that play with real storage which all of a sudden will not work, i.e., monitoring tools.

With regards receiving service for vendor products, ensure that you also obtain the service for 64-bit mode (zSeries 900) processing as some ISVs are bundling the OS/390 R10 31-bit and 64-bit services together.

Migration to Version 8 compatibility mode

The process

Assuming you have installed all the documented prerequisites, including Unicode Conversion Services and the fallback APAR PQ48486 to your Version 7 subsystem(s), you are ready to run through the traditional "CATMAINT" process of migrating DB2 to Version 8 CM. When you have migrated to Version 8 CM, you will, of course, not be able to exploit any of the new DB2 V8 features and functions. However, you will be able to now run the online REORG (SHRLEVEL REFERENCE) process against ALL table spaces in the DB2 catalog. (This has been a restriction in previous releases for some table spaces).

Migrations from V7 to V8 CM (CATMAINT) will be longer than timings experienced going from V6 to V7, although the bulk of the work will actually take place when you run the enable-new-function mode (ENFM) process later. The CATMAINT process (DSNTIJTC) remains as a single commit scope phase, in other words, all or nothing, carrying out the following processes:

- Authorization checking
- Ensuring the DB2 catalog is at the correct level
- DDL processing
- Additional processing and tailoring
- Directory header page and BSDS / SCA updates

Catalog object changes

The following object changes have been made with DB2 Version 8:

New tables

The following catalog tables will be added during the CM migration process:

- SYSIBM.IPLIST - permits multiple IP addresses to be specified for a given LOCATION
- SYSIBM.SYSSEQUENCEAUTH - records the privileges that are held by users over sequences
- SYSIBM.SYSOBDS - created in new table space SYSALTER, contains all Version history information

Removed objects

The following catalog tables, which have not been used for some time, will be dropped during DB2 Version 8 ENFM processing:

- SYSIBM.SYSLINKS
- SYSIBM.SYSPROCEDURES

N.b., VSAM data sets for indexes on the above tables can be deleted later.

DSNHDECP considerations

DB2 for z/OS Version 8 has introduced some significant changes in this area which will be discussed below:

Pre-DB2 Version 8 DECP

In previous releases of DB2, the DB2-supplied data-only load module DSNHDECP, containing the application programming defaults has always been provided in SDSNSAMP, with the user then producing a customised DSNHDECP in SDSNEXIT by running job DSNTIJUZ.

Migration implications

You are now required to create a new DSNTIJUZ-generated module called DSNHMCID in SDSNEXIT and SDSNLOAD. DSNHMCID is a data-only load module that specifies the EBCDIC CCSIDs to be used for message token translation by DB2 facilities that run in allied address spaces, such as attaches and utilities.

You need to create DSNHMCID in SDSNEXIT (using new job steps in DSNTIJUZ) and take one of the following additional actions:

- Place DSNHMCID in SDSNLOAD as well as SDSNEXIT (this is the DB2 recommendation, and DSNTIJUZ provides for doing so).
- Linklist SDSNEXIT (or whatever library holds DSNHMCID) ahead of SDSNLOAD.
- JOBLIB or STEPLIB all jobs and tasks that use DB2 utilities or execute DB2 application programs to SDSNEXIT (or whatever library contains DSNHMCID) then SDSNLOAD.

Failure to deploy DSNHMCID in this way, will cause DB2 utilities and applications to abend with S806 or SA06.

DSNHMCID

The DSNHMCID module is created by assembling and linking a new executable macro called DSNHMCIM. The parameters are illustrated below and each has the same name and meaning as their counterparts in the DSNHDECM macro:

```
DSNHMCIM
  &SCCSID=,
  &MCCSID=65534,
  &GCCSID=65534,
  &MIXED=NO
```

What if I share SDSNLOAD among DB2s with different EBCDIC CCSIDs?

You may observe minor conversion errors in messages generated by offline utilities and IMS attach. If you cannot tolerate these errors, you need either to refrain from sharing SDSNLOAD among systems with different EBCDIC CCSIDs or to modify affected jobs and tasks to STEPLIB or JOBLIB to SDSNEXIT.

Can I still place a DSNTIJUZ-generated DSNHDECP in SDSNLOAD?

As with previous releases, DB2 V8 requires that a DSNHDECP module reside in SDSNLOAD, and therefore provides the so-called DB2-supplied DSNHDECP during the SMP/E APPLY phase of installation. A new internal field has been added to the DB2-supplied DSNHDECP so that it can be differentiated from DSNHDECP modules created by running DSNTIJUZ.

If you share SDSNLOAD among DB2s having different EBCDIC CCSIDs, it is suggested that you leave the DB2-supplied DSNHDECP in SDSNLOAD and make your installation-defined DECP in SDSNEXIT available, ahead of the DB2-supplied DECP in SDSNLOAD.

DB2 startup considerations

You will no longer be able to start DB2 using the DB2-supplied DSNHDECP module. This is to permit the DB2-supplied DSNHDECP module to reside in SDSNLOAD but not be used inadvertently to start DB2. When DB2 start up detects the DB2-supplied DSNHDECP, it will discontinue with return code S04E, reason code 00E80162 and the following error message:

**DSNT527I ERROR: DB2 ATTEMPTED TO START USING THE DB2-SUPPLIED
DSNHDECP MODULE. NOTIFY THE SYSTEM PROGRAMMER.**

DB2 precompiler considerations

You will no longer be able to run the precompiler using the DB2-supplied DSNHDECP module. The precompiler now returns a code 8 and an error message when it uses the DB2-supplied DSNHDECP:

**DSNH527I ERROR: THE PRECOMPILER USED THE DB2-SUPPLIED DSNHDECP
MODULE.**

Changes to DSNTIJUZ

New job steps are introduced to let you assemble and link edit the DSNHMCID module into the SDSNEXIT and SDSNLOAD libraries. Three steps are added to prepare the DSNHMCID module:

- Job step DSNTCIDA executes the assembler and invokes the DSNHMCIM macro
- Job step DSNTCIDE links the DSNHMCID module in SDSNEXIT
- Job step DSNTCIDL links the DSNHMCID module in SDSNLOAD

These steps are placed ahead of job step DSNTLOG (runs the change log inventory for DDF) because DSNJU003 needs to load DSNHMCID.

Fallback

There are no additional considerations for fallback, because in V7 the offline message generator does not attempt to load anything to get the EBCDIC CCSIDs.

Coexistence

There are no additional considerations for coexistence. Members that have the operative PTF applied will use DSNHMCID.

Summary of DSNHDECP changes:

Here is a summary of the changes introduced in DB2 Version 8:

- You are now required to create a new DSNTIJUZ generated module called DSNHMCID in SDSNEXIT and SDSNLOAD.
- DB2 can no longer start with the DB2-supplied DSNHDECP.
- The precompiler now returns code 8 and an error message when it uses the DB2-supplied DSNHDECP.
- DSNTIJUZ has new job steps to let you assemble and link edit the DSNHMCID module in the SDSNEXIT and SDSNLOAD libraries.

Enabling new function mode

Enabling new function mode (ENFM) is the process by which the DB2 catalog is converted from DB2 Version 8 compatibility mode to DB2 Version 8 new function mode and the entire process is summarised below:

Preparing for the ENFM process

Some mandatory prerequisites to ensure before beginning the ENFM process:

- You cannot begin ENFM processing if any active DB2 Version 7 subsystems still exist in your data sharing group. All members of the data sharing group must have migrated to Version 8 compatibility mode successfully.
- You will NOT be able to start a DB2 Version 7 member in this group once the ENFM process begins. This is true regardless of whether or not the ENFM process is actually running at the moment of starting the Version 7 member.
- With no fallback or coexistence with DB2 Version 7, do not plan to run the ENFM process until you are absolutely sure that you will not need to return to DB2 Version 7. Restoring the entire system from backup copy is the **ONLY** option and it's a time consuming one so the important message is "don't short change compatibility mode". Make sure you've been through all the major cycles / batch routines of your applications (1 month, 3 months even 6 months), before moving on through ENFM to NFM as it is very difficult to go back.
- Recommendation is that you reorganise your DB2 catalog objects prior to the ENFM phase. This will also provide a useful estimation of the time required to REORG the catalog during the ENFM process.
- Plan to cleanup your catalog prior to ENFM, removing unwanted objects.
- Take an image copy of the DB2 catalog and directory after job DSNTIJNE has completed processing or if the DSNTIJNE processing is paused.

Things to consider before starting ENFM

- Ensure that the BP8KO, BP16KO, and BP32K buffer pools exist as processing will fail otherwise. For data sharing groups, you must also define the GBP8KO, GBP16KO, and GBP32K group buffer pools.
- Increase the size of the catalog table space and index space VSAM data sets prior to ENFM in order to accommodate for the expansion of adding a 2-byte length to existing CHAR columns. This will also position you for the possible future use of long names in the catalog, once you get to NFM.

Backing up your system before ENFM processing

Taking a backup of the entire subsystem is recommended prior to commencing the ENFM process, as it's the only way of restoring your system in the event that the ENFM process fails and you are forced to restore. Ensure that you quiesce the V8 CM system before running DFSMS to create a copy of the entire DB2 Version 8 compatibility mode subsystem, including the DB2 catalog and directory, logs, BSDSs and all your application data.

ENFM – when should I run it?

The process of enabling your subsystem for new function mode can span a number of maintenance windows giving you the flexibility to plan for and schedule the process inline with business processes. Of course, the size of your DB2 catalog will have a lot of input into the planning of outage windows and it is recommended that you run a complete REORG of your catalog in compatibility mode, to help estimate the timing of the ENFM event.

The DSNTIJNE job can be stopped (halted) after the REORG of any table space (or long name DDL steps - the ENFMxxx0 steps) and can be restarted without modification by submitting the DSNTIJNE job again. The job will skip table spaces already processed and resume processing at first table space not successfully converted.

Recommendation is NOT to stay in ENFM phase any longer than necessary

The ENFM Process

The process of ENFM centres around the running of the DSNTIJNE job supplied in SDSNSAMP, which invokes the CATENFM utility (not CATMAINT). The first step of this job invokes entry into enable-new-function mode, followed by an iteration of several steps for each of the eighteen table spaces included in the conversion from EBCDIC to Unicode process. The final step of entering new function mode is accomplished by running DSNTIJNF, which sets the appropriate bit in the catalog, marking final completion of the ENFM process.

In summary, the conversion of the DB2 catalog involves the following end-to-end process:

Running DSNTINST

In the same way as you would initiate any previous migration, you begin by running the installation CLIST (DSNTINST) and entering the enabling-new-function mode option ENFM in the first field of the DSNTIPA1 panel.

You should leave the DATA SHARING and DATA SET NAME(MEMBER) fields blank and ensure that the INPUT MEMBER NAME field (specified for conversion to NFM) is set to the OUTPUT MEMBER NAME (previously specified during migration to CM).

Invoking the ENFM option, will display a subset of the panels presented for a standard migration and, in fact, the MIGRATE mode and ENFM mode have just a few panels in common, they are:

- DSNTIPA1 - the primary panel
- DSNTIPT - where you specify the prefix.NEW.SDSNSAMP data set

During ENFM processing, the DSNTINST CLIST generates installation jobs DSNTIJNE, DSNTIJNF, DSNTIJNG, DSNTIJEN, DSNTIJNH, DSNTIJMC and DSNTIJNR. It also customizes the Version 8 IVP jobs. It then calculates the preliminary space allocations for the shadow data sets (used later during the online REORGs) and displays the recommendation sizes on the DSNTIP00 panel for each of the eighteen DB2 directory and catalog table spaces to be converted. Panel DSNTIP00 also displays the DASD device and volume serial on which the catalog table space resides and a secondary space allocation is also provided. A line is also provided for catalog indexes highlighting DASD device and volume serial.

Panel DSNTIP01 lists image copy dataset allocation information for the eighteen DB2 directory and catalog table spaces. On this panel the user enters only a common data set name prefix and a single device type for all image copy data sets. Space for the image copy data sets is determined by DB2 for all table spaces except SPT01.

DSNTIP02 displays a summary of space requirements based on the data entered on the previous panels and this can be accepted by pressing the ENTER key, which also signals the CLIST to generate the ENFM jobs. To revise the image copy data set names and target devices by returning to panel DSNTIP01, press the RETURN key.

DSNTIJNE

DSNTIJNE is the key job in the ENFM migration process and it performs several functions including:

- Changes types and lengths of existing catalog columns.
- Converts catalog data to Unicode.
- Changes buffer pools for several catalog table spaces.
- Changes catalog indexes with varying length columns to NOT PADDED.
- Changes page size of several catalog table spaces.

Catalog changes during ENFM processing

A number of changes to the DB2 catalog take place during ENFM processing:

- The SYSOBDS catalog table is added with associated indexes
- The SYSDUMMY1 catalog table is moved to the SYSEBCDC table space
- NO columns added to existing catalog tables
- Columns changed in existing catalog tables
- Most catalog and one directory tables have columns that are changing type, length, or both and changes include:
 - VARCHAR ==> VARCHAR(128 or greater)
 - CHAR ==> VARCHAR(128 or greater)

Index padding

With DB2 Version 8, varying length columns are no longer always padded to their full length when part of an index key. A new parameter, NOT PADDED, can be used during the creation or altering of an index and padding will not occur, resulting in keys that will be varying in length.

During the migration process, many catalog indexes are changed to NOT PADDED indexes, although no user indexes on the catalog will be changed to NOT PADDED during the process. The system default will change to NOT PADDED when we are in NFM, a default which is different for a migration as opposed to an install:

- The migration default is PADDED
- The install default is NOT PADDED (this can be changed via a DSNZPARM parameter and is only applicable for indexes with at least one varying length key column).

What happens if I have a space problem during ENFM process?

If you have a space problem during the running of DSNTIJNE, succeeding steps will be skipped. The last step of the DSNTIJNE job, performs a -TERM UTIL command, to ensure that the table spaces become immediately available upon any failure. When restarted, the job will skip already processed table spaces and resume processing at the first table space not successfully converted.

How can I tell where I am in the process?

Use the DISPLAY GROUP DETAIL command to determine the current mode and use the list of objects to see how many are now enabled for new function mode. This method of analysis should be used, even in a non-data sharing environment.

```

DSN7100I -V81A DSN7GCM
*** 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 ZS13PE PR21    PRLMPR21
-----
TABLE          ENABLED
SPACE          NEW FUNCTION
-----
SYSVIEWS      YES
SYSDBASE      YES
SYSDBAUT      YES
SYSDDF        YES
SYSGPAUT      YES
SYSGROUP      YES
SYSGRTNS      YES
SYSHIST       YES
SYSJAVA       YES
SYSOBJ        YES
SYSPKAGE      YES
SYSPLAN       NO
SYSSEQ        NO
SYSSEQ2       NO
SYSSTATS     NO
SYSSTR        NO
SYSUSER       NO
SPT01         NO
-----
*** END DISPLAY OF GROUP(.....)

```

Can I change the order of table spaces in the DSNTIJNE job?

NO, the CATENFM order is fixed and job DSNTIJNE MUST be completed in the enforced order, which is SYSVIEWS, SYSDBASE, then alphabetical order with the SPT01 directory table space last. The process converts eighteen table spaces: seventeen catalog and one directory table space, with SYSCOPY and SYSEBCDC remaining EBCDIC. SYSJAUXA and SYSJAUXB are LOB table spaces that inherit their encoding scheme from their base table space, SYSJAVA. As SYSJAVA is converted to Unicode, so are both of these, even though SYSJAUXB contains a BLOB and no real data conversion occurs there.

The conversion process CANNOT happen if NO image copies are taken of the table spaces prior to conversion. If any RI relationships exist they will be handled during the process.

The Unicode conversion step

The Unicode conversion is accomplished using the online REORG with SHRLEVEL REFERENCE and no special locking considerations are deployed during the process. 'FOR BIT DATA' columns will NOT be converted to Unicode. Should the process fail during a particular table space

conversion, the old catalog data is still maintained, hence there should be no outage concern. Table spaces are marked 'UNICODE' during online REORG processing and once a table space is Unicode, there's NO going back other than via restoring the entire DB2 subsystem environment.

DSNTIJNF

Following a successful completion of DSNTIJNE, the DSNTIJNF job's sole purpose is to flip a bit in the DB2 catalog to place the DB2 subsystem in new-function mode. The V8 precompiler will now accept program source code that uses the new Version 8 syntax by specifying a value of YES for the NEWFUN parameter. (Important - The default will change from NEWFUN(NO) to NEWFUN(YES) as part of the running of DSNTIJNG). DBRMs produced by the precompiler in all modes when NEWFUN(YES) is specified, now use Unicode for its SQL statements. Only a V8 NFM system can bind the DBRM and the precompiler marks the DBRM as V8.1-dependent even if it does not use any new V8 syntax.

You can of course set NEWFUN to NO which will instruct the precompiler to disable DB2 Version 8 functions and produce a resulting DBRM that uses EBCDIC for SQL statements as before. It is subsequently not marked as DB2 Version 8 dependent under this option and any release of DB2 can continue to bind this DBRM.

Important - following the successful processing of DSNTIJNF and entry into new function mode, you can still return to ENFM, by running the supplied job DSNTIJEN. (In so doing, of course, you will be disabling usage of new functions in this environment).

DSNTIJNG

DSNTIJNG rebuilds the DSNHDECP module to specify new function mode as the default for the precompiler via NEWFUN(YES).

Enabling new function mode considerations

Here are some early findings and considerations from ESP migrations::

- Temporary data set sizes will increase for the linked table spaces, namely SYSDBASE, SYSVIEWS, SYSDBAUT, SYSGROUP and SYSPLAN. During sort processing for these objects, all input records must be blown up to the maximum size as it is not possible to sort variable length records. Also, SYSDBASE has a lot of columns (and links) that change to VARCHAR(128).
- Watch out for usage of the SYSREC2 DD in the DSNTIJNE job. It will never be used by a normal REORG job and is only used here in the ENFM steps which reorg SYSVIEWS, SYSGRTNS, and SYSOBJ. This DD card is required in these steps because an SQL statement must be concatenated in its entirety before it can be converted to Unicode. SYSREC2 is used to facilitate this concatenation and conversion.
- It is recommended that you increase the size of the catalog table space/index space VSAM data sets in preparation for possible future use of long names in the catalog. Most of these changes could be done at any time, even in new function mode, but some should be done before beginning the enable-new-function mode process.
- As a result of the SQL parse tree restructuring in Version 8, parse trees stored in SYSVIEWS will take up much more space than in previous releases, so it is recommended that you increase the allocation size.
- BP8KO and BP16KO buffer pools must exist prior to the beginning of enable-new-function mode processing. Enable-new-function mode processing will fail with a resource unavailable message if these buffer pools do not exist. Group buffer pools (GBP8KO and GBP16KO) would also need to be defined for data sharing groups.
- Change data capture (CDC) will also be turned off on any catalog table that has it enabled. This will be done automatically during enable-new-function mode processing. Once in new function mode, CDC can be turned back on, but this is a manual process.
- SYSCOLUMNS.STATSTIME statistics will be invalidated as part of the ENFM process.
- The VSAM data sets for the two dropped catalog tables (SYSLINKS and SYSPROCEDURES) can be deleted by the user at any time in new function mode.
- SYSDUMMY1 will be moved from the SYSSTR table space to the new SYSEBCDC table space during the ENFM process.
- There is no need to drop user indexes on the catalog prior to running DSNTIJNE. Although DSNTIJNE will not modify the attributes of these indexes in any way, they will be automatically rebuilt as a result of the REORG of the table space which contains the table(s) on which the index is defined. Please note that this may result in a considerably larger index, if its columns were increased in size and marked VARCHAR.

Once in NFM, for user indexes on catalog tables, if a column that constitutes a particular index's key has been altered to VARCHAR or the column length has been extended during ENFM, it would be a good idea to ALTER the index to NOT PADDED and rebuild it.

What's changed now that I am in NFM?

Some things to be aware of when you first enter new function mode:

Long names considerations

Once a DB2 subsystem/group has successfully been converted to V8 new function mode the DB2 catalog will be enabled for long names. In most cases this conversion process consists of altering/converting existing CHAR(8), CHAR(18), and VARCHAR(18) columns to be VARCHAR(128) to allow for longer names in the catalog.

For example, in a pre-new function mode catalog, a table name, column name, and index name are all VARCHAR(18) columns. This allows for names with a maximum length of eighteen characters. In new function mode the maximum lengths of these columns becomes 128 so that we can accommodate much longer names.

In another example, the owner, schema, and creator name columns change from CHAR(8) to VARCHAR(128) columns. Increasing the lengths of some of these catalog table columns will cause many catalog table rows to exceed the current BPO 4K page size maximum. In these cases, the table spaces that contain these tables will be moved to an appropriately sized buffer pool during enable-new-function mode processing.

Traditional eight character names (dbname, tname, package name etc) are now stored as VARCHAR(24), although names are still limited to eight characters as they need to fit in with the z/OS limits for data set and member names. It is also worth noting that although many of the DB2 catalog table spaces are now Unicode table spaces, their names are not changed in SYSCOPY.

Other logical name increases to watch for include:

- Table, view and alias names
- Column names now increased to (VARCHAR(30)), although the V8 infrastructure could support 128
- Schema
- Authorisation id limited to eight although the V8 infrastructure could support 128
- Routines: UDF, stored procedure
- Triggers, packages

Immediate effects

Other significant changes as a result of the ENFM migration process:

- Index keys can now be greater than 255 bytes.
- Some row sizes increased beyond 4056 bytes.
- BP8KO and BP16KO buffer pools required following catalog row expansions. (Corresponding group buffer pools also required if data sharing).

- Seventeen catalog table spaces plus one directory table space out of the twenty two catalog table spaces are now Unicode.
- Watch out for Unicode ordering changes, with ORDER BY now returning rows in Unicode order. This happens to match ASCII order for the first 128 code points including numbers and letters (numbers first).

DSNZPARM parameter changes

Several subsystem parameters have been added to the set available for online change using the -SET SYSPARM command which was introduced in Version 7, they are:

Parameter	Description	Parameter	Description
CHGDC	DPROP Support	EDPROP	DPROP Support
SYSADM1	System Admin 1 ^{note 1}	SYSADM2	System Admin 2 ^{note 1}
SYSOPR1	System Operator 1	SYSOPR2	System Operator 2
CACHEDYN	Cache Dynamic SQL	SRTPOOL	Sort Pool Size
XLKUPDLT	X Lock for Searched U/D	MAXKEEPD	Max Kept Dyn Stmts
PARTKEYU	Update Part Key Cols	RESYNC	Resync interval
POOLINAC	Pool Thread Timeout	TCPKPALV	TCP/IP Keepalive
IDTHTOIN	Idle Thread Timeout	TCPALVER	TCP/IP Already Verified
MAXTYPE1	Max Type 1 Inactive	EXTRAREQ	Extra Blocks Req
EXTRASRV	Extra Blocks Srv	IMMEDWRI	Immediate Write

note 1: Install SYSADM authority is needed to perform an online change of SYSADM or SYSADM2.

Online DSNZPARM parameter change considerations

For most parameters, online change is transparent, with the change taking effect immediately, however there are a few parameters for which this is not the case and these are discussed below:

- PARTKEYU - after dynamically changing the partitioning key columns parameter from YES to NO, attempts to inappropriately update the value in a partitioning key column result in a -904 resource unavailable SQL code. Changing the parameter to anything more restrictive would result in applications receiving the -904 code, that is, from YES to NO, from YES to SAME, or from SAME to NO.
- SYSADM1/2 - for some instances where the auth ids are cached, the old auth id remains in effect after online change.
- CACHEDYN - when YES is specified for the Cache Dynamic SQL parameter, statements are cached in the EDM pool as usual. When the value is changed to NO, existing statements in the cache will not be used by new threads, but the statements will remain in the pool until they are no longer referenced. Changing to NO does not reduce the size of the EDM pool allocated for dynamic statements. Changes to the CACHEDYN parameter will not affect the

previous settings of currently running statements; however, they will affect any future statements coming in.

- **MAXKEEPD** - changes to the **MAXKEEPD** parameter will take effect after the next **COMMIT** statement executes.
- **XLKUPDLT** - changes to **XLKUPDLT** will not affect the previous settings of currently running statements; however, they will affect any future statements coming in.

REORG considerations

When in new function mode, REORGs of the DB2 catalog or CHECK INDEX jobs will require more work / sort space, since VARCHAR columns have to be padded out to the full length for sort to have fixed length records to work with. The *DB2 Universal Database for z/OS Utility Guide and Reference, SC18-7427*, describes these requirements in the section entitled "Defining the work data set for CHECK INDEX".

Some thoughts on installation of new DB2 subsystems

CATMAINT during install?

A new change to the install process is the inclusion of DSNTIJTC (CATMAINT) for a fresh install of V8. As with a migration, you should expect console message DSNT5011 with reason code 00C900A6 the first time you start DB2 V8, and you should then run DSNTIJTC in response. You can then proceed to the remaining install jobs such as DSNTIJTM.

Please note also, that during the DB2 version 8 installation process, you now need to run the DSNTIJUZ job prior to DSNTIJID (the DB2 version 7 documentation tells you to run DSNTIJID ahead of DSNTIJUZ).

Appendix 1: Sample UCS control statements - German

Here is a sample set of German Unicode Conversion Services (UCS) control statements as displayed output from the D UNI,ALL command:

```
D uni,all
CUN3000I 12.44.42 UNI DISPLAY 898
  ENVIRONMENT: CREATED      09/02/2003 AT 23.19.43
                MODIFIED    09/02/2003 AT 23.19.44
                IMAGE CREATED 07/14/2003 AT 09.52.42
  SERVICE: CHARACTER      NORMALIZATION CASE
  STORAGE: ACTIVE        266 PAGES
                LIMIT      600 PAGES
  CASECONV: NORMAL
  NORMSERV: DISABLED
  CONVERSION: 00037-00273-ER          00037-00367-ER
                00037-00500-ER          00037-00819-ER
                00037-00858-ER          00037-01047-ER
                00037-01200 (13488) -ER  00037-01208-ER
                00273-00037-ER          00273-00367-ER
                00273-00819-ER          00273-00858-ER
                00273-01047-ER          00273-01200 (13488) -ER
                00273-01208-ER          00367-00037-ER
                00367-00273-ER          00367-00500-ER
                00367-00819-ER          00367-00858-ER
                00367-01047-ER          00367-01200 (13488) -ER
                00367-01208-ER          00500-00037-ER
                00500-00367-ER          00500-00819-ER
                00500-00858-ER          00500-01047-ER
                00500-01200 (13488) -ER  00500-01208-ER
                00819-00037-ER          00819-00367-ER
                00819-00500-ER          00819-01047-ER
                00819-01200 (13488) -ER  00819-01208-ER
                00858-00037-ER          00858-00273-ER
                00858-00367-ER          00858-00500-ER
                00858-01047-ER          00858-01200 (17584) -ER
                00858-01208-ER          00923-01200 (17584) -ER
                00923-01208-ER          01047-00037-ER
                01047-00367-ER          01047-00819-ER
                01047-00858-ER          01047-01200 (13488) -ER
                01047-01208-ER          01200 (13488) -00037-ER
                01200 (13488) -00273-ER  01200 (13488) -00367-ER
                01200 (13488) -00500-ER  01200 (13488) -00819-ER
                01200 (17584) -00858-ER  01200 (17584) -00923-ER
                01200 (13488) -01047-ER  01200-01208-ER
                01200 (13488) -01252-ER  01208-00037-ER
                01208-00273-ER          01208-00367-ER
                01208-00500-ER          01208-00819-ER
                01208-00858-ER          01208-00923-ER
                01208-01047-ER          01208-01200-ER
                01208-01252-ER          01252-01200 (13488) -ER
                01252-01208-ER
```

Appendix 2: Sample UCS control statements - Japan

Here is a sample set of Japanese Unicode Conversion Services (UCS) control statements as displayed output from the D UNI,ALL command:

```

D UNI,ALL
CUN3000I 10.04.38 UNI DISPLAY 969
ENVIRONMENT: CREATED      09/22/2003 AT 10.01.16
              MODIFIED    09/22/2003 AT 10.01.18
              IMAGE CREATED 09/04/2003 AT 20.16.24
SERVICE: CHARACTER      NORMALIZATION CASE
STORAGE: ACTIVE         698 PAGES
              LIMIT       1800 PAGES
CASECONV: NORMAL
NORMSERV: ENABLED
CONVERSION: 00367-01200(13488)-ER      00367-01208-ER
              01200(13488)-00367-ER    01200-01208-ER
              01208-00367-ER           01208-01200-ER
              00037-00367-ER           00037-01200(13488)-ER
              00037-01208-ER           00367-00037-ER
              01200(13488)-00037-ER    01208-00037-ER
              01047-00367-ER           01047-01200(13488)-ER
              01047-01208-ER           00367-01047-ER
              01200(13488)-01047-ER    01208-01047-ER
              00037-01047-ER           01047-00037-ER
              00290-00367-ER           00290-01200(13488)-ER
              00290-01208-ER           00367-00290-ER
              01200(13488)-00290-ER    01208-00290-ER
              00037-00290-ER           00290-00037-ER
              01047-00290-ER           00290-01047-ER
              00930-00367-ER           00930-01200-ER
              00930-01208-ER           00367-00930-ER
              01200-00930-ER           01208-00930-ER
              00300-00367-ER           00300-01200(13488)-ER
              00300-01208-ER           00367-00300-ER
              01200(13488)-00300-ER    01208-00300-ER
              01041-00367-ER           01041-01200(13488)-ER
              01041-01208-ER           00367-01041-ER
              01200(13488)-01041-ER    01208-01041-ER
              00037-01041-ER           01041-00037-ER
              01047-01041-ER           01041-01047-ER
              00943-00367-ER           00943-01200-ER
              00943-01208-ER           00367-00943-ER
              01200-00943-ER           01208-00943-ER
              00941-00367-ER           00941-01200(13488)-ER
              00941-01208-ER           00367-00941-ER
              01200(13488)-00941-ER    01208-00941-ER
              00290-01041-ER           01041-00290-ER
              00930-00943-ER           00943-00930-ER
              00300-00941-ER           00941-00300-ER
              00367-00500-ER           00500-00367-ER
              00500-01200(13488)-ER    01200(13488)-00500-ER
              00500-01208-ER           01208-00500-ER

```


00367-00897-ER	00897-00367-ER
00897-01200 (13488) -ER	01200 (13488) -00897-ER
00897-01208-ER	01208-00897-ER
00897-00037-ER	00037-00897-ER
00897-01047-ER	01047-00897-ER
00897-00290-ER	00290-00897-ER
01390-00367-ER	01390-01200-ER
01390-01208-ER	08482-00367-ER
08482-01200 (17584) -ER	08482-01208-ER
16684-00367-ER	16684-01200 (17584) -ER
16684-01208-ER	00367-01390-ER
01200-01390-ER	01208-01390-ER
00367-08482-ER	01200 (17584) -08482-ER
01208-08482-ER	00037-08482-ER
08482-00037-ER	00367-16684-ER
01200 (17584) -16684-ER	01208-16684-ER

Glossary of Terms

Abbreviation / Acronym	Description
CF, coupling facility	A shared memory device used in a Parallel Sysplex environment for sharing data used by all participating systems. The shared (CF) 'memory' is exploited using coupling facility structures.
CICS, Customer Information Control System	IBM online transaction management system.
CM, compatibility mode	Migrating from DB2 Version 7 via CATMAINT process, places the DB2 subsystem in DB2 Version 8 compatibility mode, the first tier of the migration process.
Data sharing	The concurrent access of databases by two or more DB2 subsystems.
DB2	IBM's premier relational database management system.
ENFM, enable-new-function mode	Migrating from DB2 Version 8 Compatibility mode to new function mode is accomplished through the enable-new-function mode process. A serial process of changing the DB2 via the CATENFM process, places the DB2 subsystem in DB2 Version 8 new function mode, the final tier of the migration process.
ESP, Early Support Program	A program run by the Product Introduction Centre, to validate the content and functionality of a new release of a software product.
GA, General Availability	Time when new product code becomes generally available to the public.
IMS, Information Management System	IBM hierarchical database management system.
ISV, Independent Software Vendor	A company that produces and/or sells software products.
NFM, new function mode	After successful completion of the enable new function mode process and the running of job DSNTIJNF, the subsystem or group will be in new function mode. At this point, all new functions of DB2 for z/OS Version 8 can be used.
OS/390	MVS operating system running under the 390 architecture. N.b., not a valid operating system in which to run DB2 Version 8.
Parallel Sysplex	IBM technology introduced in 1994, comprising of multiple processors connected together and processing workload. A development of sysplex to allow high speed datasharing between S390 images.
PIC, Product Introduction Centre	Formerly known as the ISC.
SVL	DB2 Silicon Valley Laboratory.
USS, UNIX Systems Services	Fully UNIX compliant operating system under MVS.
WLM, Workload Manager	Software used for managing workload scheduling and performance in a 390 environment.
XCF, Cross System Coupling Facility	Software used for messaging and communication between images in a Parallel Sysplex environment. (Not to be confused with the hardware coupling facility component).
z/OS	Latest MVS operating system running under the z/Architecture, supporting 64-bit addressing.