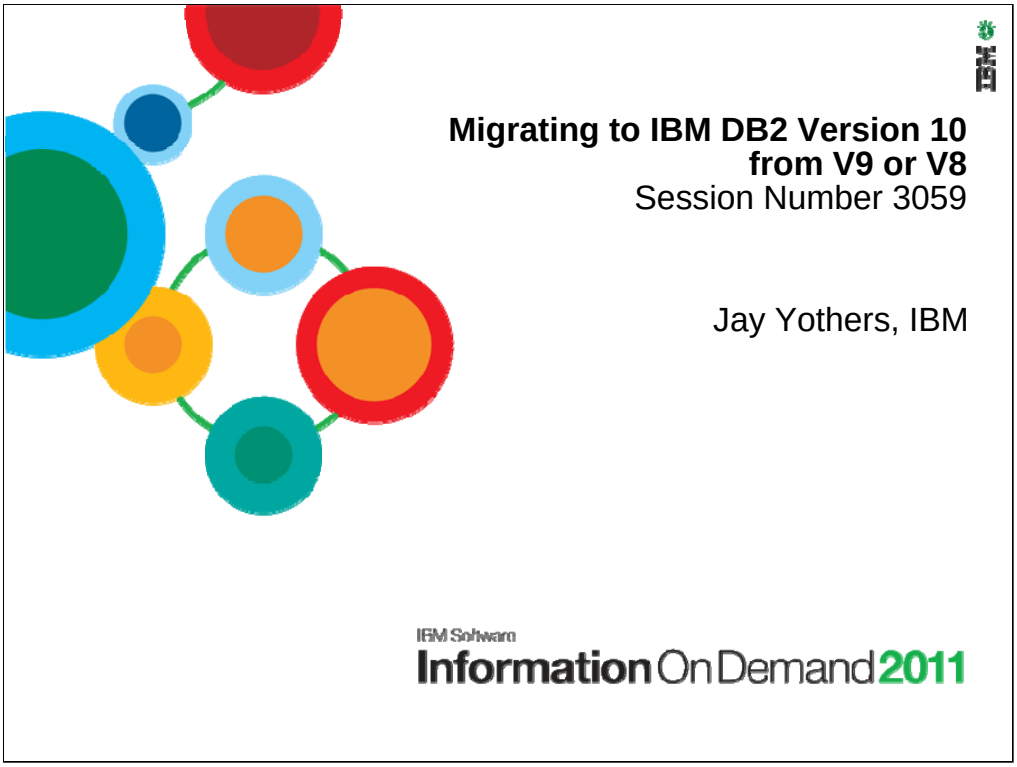

IBM
Migrating to IBM DB2 Version 10
from V9 or V8
Session Number 3059
Jay Yothers, IBM
IBM Software
Information On Demand 2011

Please Note:

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion.

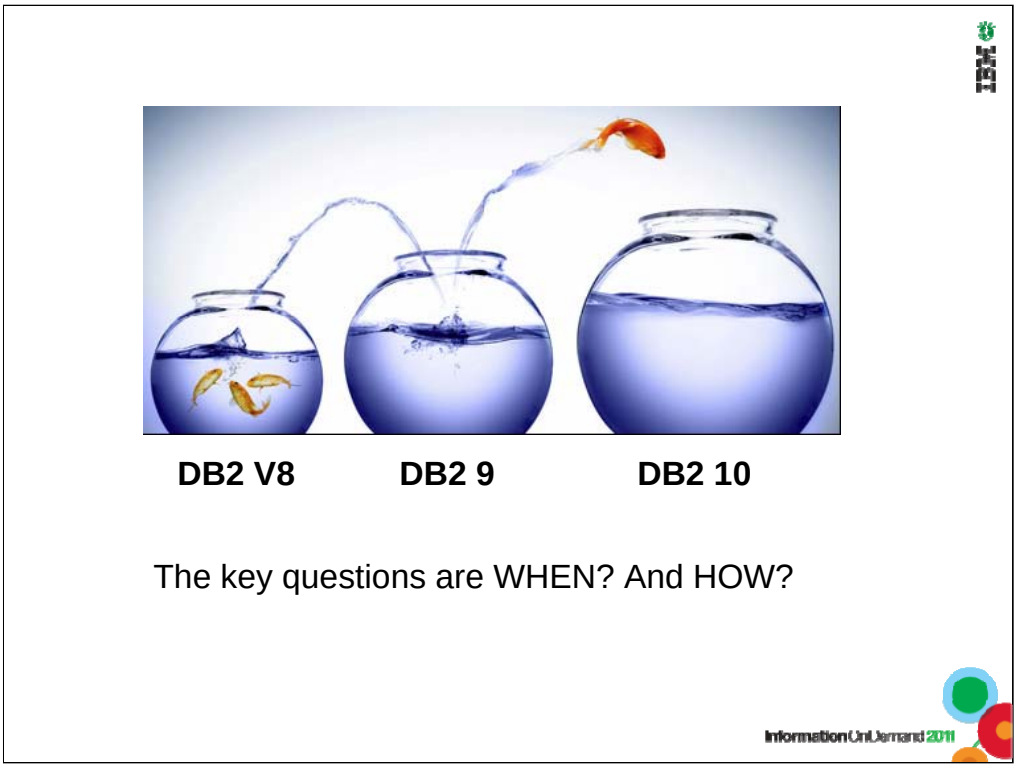
Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

Migration Agenda

- Why Migrate
- Hardware & Software Prerequisites
- Pre-migration planning
- Incompatibilities after migrating to V10
- Migration paths available
  - V9 to V10 "Standard release" Migration Process Overview
  - V8 to V10 "Skip release" Migration Process Overview
- The Migration Process
  - New Release Migration Planning
  - Migration to Conversion Mode (Running DSNTIJTC)
  - Enable New-Function-Mode
  - Catalog Restructure
- Fallback - Returning to a prior release
- Migration Considerations For Specific Features
- Eliminate DDF Private Protocol



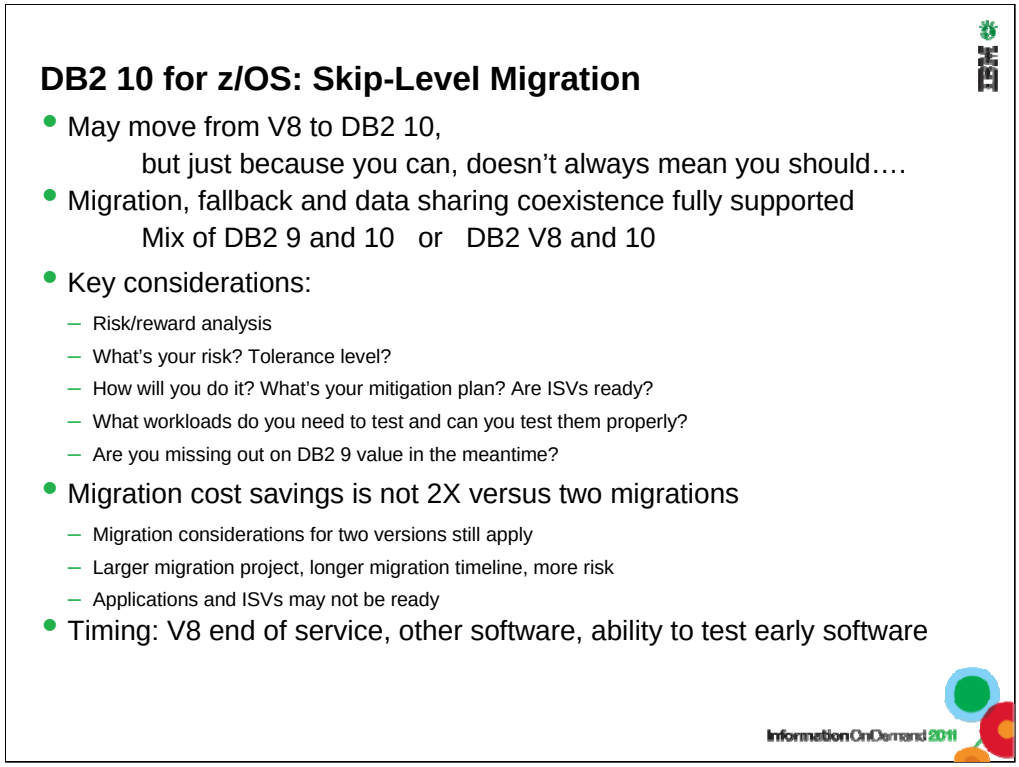
If you are running DB2 9 today, then DB2 10 is an opportunity. If you are running DB2 V8 today, then your need is more urgent. DB2 V8 goes out of service in less than a year, April 2012, so your choices are to migrate to DB2 10 or 9 or to purchase extended service. Is your current fish bowl getting constrained? What is limiting you? Is it CPU? Virtual storage? Latching? DB2 catalog and directory? Concurrent utilities? Are you currently running DB2 9? V8? V7?

Should you migrate to DB2 10? The answer is a definite Yes.

The question is not so much whether to migrate as when and how to migrate. If you are running DB2 9 today, then DB2 10 is in your future, giving you more room to

grow, with fewer limits, lower costs, and more for less. If you are running DB2 V8 today, then you have a choice of jumping to DB2 9 or directly to DB2 10. So the key question is, “When should I

migrate to DB2 10?” While DB2 10 is expected to be better than prior versions, it will have maturity and service delivery like other software, with more defects at first, then fewer as the software matures. Determining when the software is ready for a specific customer and when the customer is ready for the software depends upon the specific customer resources for testing, prior experience, and the value for the improvements versus the need for stability. Many customers depend upon tools or other software, and having that software work with DB2 is a prerequisite. When this information is known, we can answer the question.



DB2 10 supports migration from DB2 9 NFM or from V8 NFM. Customers not yet running V8 or DB2 9 should plan to migrate to DB2 for z/OS V8 NFM as preparation for a migration to DB2 10.

We estimate that about one customer in five migrated using a skip version technique for V5 to V7, and we'll see a similar fraction this time. The key value for skip customers on V7 or new on V8. When should I migrate to DB2 10? While DB2 10 is expected to be better than prior versions, it will have maturity and service delivery like other software, with more defects at first fewer as the software matures. Determining when the software is ready for a specific customer and when the customer is ready for the software depends upon the specific customer resources for testing, prior experience, and the value for the improvements versus the need for stability. Many customers depend upon tools or other software, and having that software work with DB2 is a prerequisite.

When this information is known, we can answer the question.

Normal migration is moving one version at a time every three years. For customers who have gotten behind, the ability to skip a migration cycle will be attractive, but this ability is not "something

for nothing". Customers need to consider the tradeoffs and challenges that we know about in skip version migration. Most customers who migrate to new versions by three years after GA are

already on DB2 9.

The project for skipping is larger. While the testing and rollout are only a little larger than a single version migration, the education and remediation work is roughly double the normal size. Most

project plans estimate 150%. Consider the timing carefully. Improvements in DB2 9 are delayed for 2 to 4 or more years with a skip plan. You may need to have extended service on V8.

http://www.ibm.com/support/docview.wss?uid=swg21006951

DB2 10 for z/OS

GA'ed October 2010

Completed Largest Beta Ever

- 23 WW customers
- +10 Extended Beta
- Over 80 vendors

Fastest uptake out of the gate

- As of July, 2011 3x more unique customers compared to prior release
- More than 4x the number of licenses
- About 25% are migrating from V8
- Every core beta customer is continuing with migration plans

First large customers already in production

Information On Demand 2011

Customers have been enthusiastic about DB2 10 for z/OS performance and scalability. Some customers are able to simplify their structure and many are seeing better productivity.

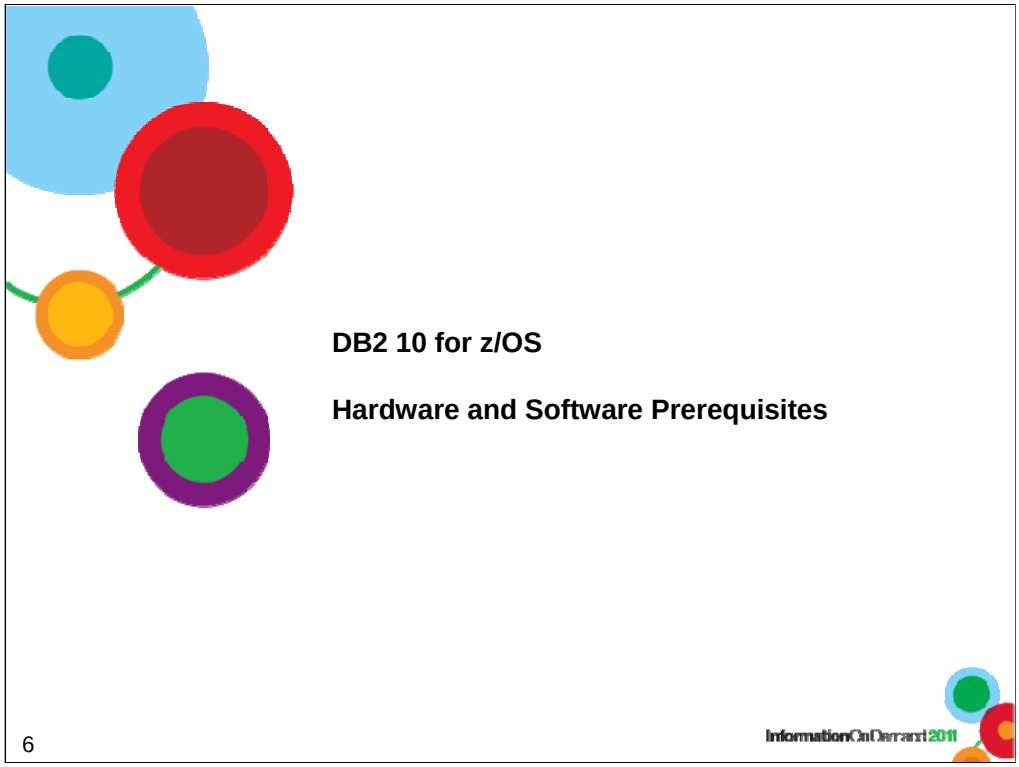
The security and temporal function are seeing strong early acceptance. A safer infrastructure with better audit function can help customers avoid the need for a new security structure.

Customers are signing up for the V8 to DB2 10 migration to save time and to get these improvements faster. The first production has started. Most of the 23 beta customers plan to move to production in 2011.

Over 100 customer orders of DB2 10,

More than 4x the number of DB2 10 licenses versus DB2 9 at 6 months after GA

Close to 25% of the DB2 10 customers are looking to move from V8


DB2 10 for z/OS
Hardware and Software Prerequisites
6
InformationOnDemand2011

Prerequisites – Hardware & Operating System

- Processor requirements:
  - z196, z10, z9, z990 and z890 processors supporting z/Architecture
    - z800 or z900 processors NOT supported
  - DB2 10 for z/OS will probably require increased real storage for a workload compared to DB2 9 for z/OS
- Software Requirements:
  - z/OS V1.10 Base Services (5694-A01) at minimum
  - DFSMS V1 R10 – DB2 Catalog becomes SMS managed
  - Language Environment Base Services
  - z/OS Version 1 Release 10 Security Server (RACF)
  - IRLM Version 2 Release 3 (Shipped with DB2 10 for z/OS)
  - z/OS Unicode Services and appropriate conversion definitions are required.
  - DB2 Connect 9 Fixpack 1 is required; 9.7 Fixpack 3a recommended

This slide lists only the key highlights! Also, this applies whether you are migrating or installing.

Processors

DB2 for z/OS Version 10 operates on System z (z196, z10, z9, z990, z890 and later) or equivalent processors running in 64-bit mode supported by z/OS V1R10. The processor must have enough real storage to satisfy the combined requirements of, DB2, z/OS and the appropriate DFSMS (tm) storage management subsystem components, access methods, telecommunications, batch requirements, and other customer required applications.

In addition, 64-bit virtual addressing support will probably require increased real storage for a workload compared to DB2 9 for z/OS.

DB2 Version 10 requires the function that is provided by z/OS Version 1 Release 10 Base Services (5694-A01) with the following base and optional elements:

• DFSMS Version 1 Release 10 features, part of the Systems Management optional feature of z/OS; specifically, DFSMShsm for archiving and DFSMSdss for concurrent copy in Utilities.

• Language Environment Base Services

• z/OS Version 1 Release 10 Security Server (RACF)

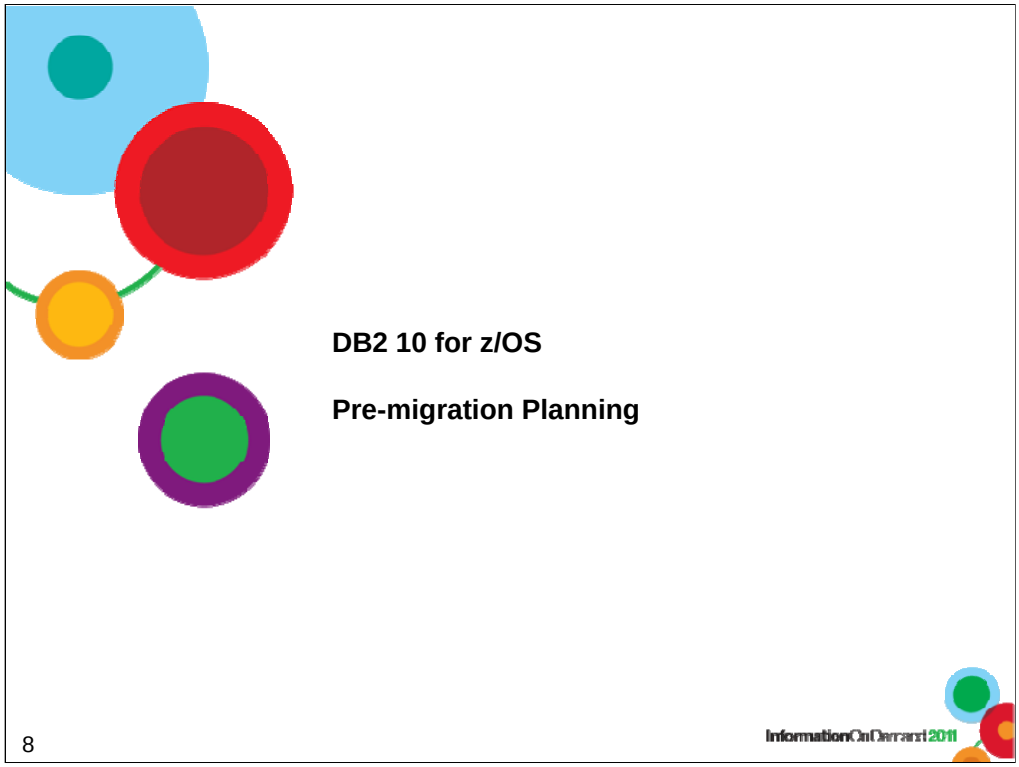
• IRLM Version 2 Release 3 (delivered with DB2 V10)

▪WebSphere Replication Server Version 9 Release 1 (5655-R55) and IBM Data Event Publisher Version 9, work with the following DB2 for z/OS versions:

- ➢DB2 9 for z/OS (5635-DB2)
- ➢DB2 Universal Database for z/OS Version 8 (5625-DB2)

• z/OS Unicode Services and appropriate conversion definitions are required. (As installed for V8 and V9).

• Some of the basic operation of a DBMS is provided by utility functions, such as backup, recovery, reorganization, loading and unloading data, gathering statistics and checking data, indexes, and large objects. Customers should ensure that these functions are provided either by ordering DB2 Utilities Suite for z/OS Version 10, or by obtaining equivalent function elsewhere.


DB2 10 for z/OS
Pre-migration Planning
8
InformationOnDemand2011

Run DSNTIJPM (DSNTIJPA) pre-migration job

- Check for incomplete object definitions in the catalog
  - Complete or drop all such objects ***before*** beginning migration
    - Avoids possible behaviour differences in DB2 10.
- Run DSNTIJPM or DSNTIJPA, to identify them:
  - DSNTIJPM ships with DB2 10 and should be run on previous releases to identify pre-migration catalog cleanup requirements.
    - DSNTIJPM may provide DDL or utility statements for the cleanup.
  - DSNTIJPA is the same job and is shipped for both DB2 V8 & DB2 V9, to maximize prepare time.

Important preparation

- PDSE requirement

Enter "/" to select action	Dsorg	Recfm	Lrecl	Blksz
SYS2.DB2.V10.SDSNLINK	PO	U	0	32760
SYS2.DB2.V10.SDSNLOAD	PO-E	U	0	32760
SYS2.DB2.V10.SDSNLOD2	PO-E	U	0	32760
SYS2.DB2.V10.SDSNMACS	PO	FB	80	3120

- Migrate only from a system with an expanded BSDS (DSNJCNVB)
- Old plans and packages V5 or before → REBIND
- Plans containing DBRMs → packages
  (PK62876 (V9), PK85833 (V9), PK79925 (V8), PM01821 (All))
- ACQUIRE(ALLOCATE) only applies to DBRMs in plan → packages, ACQUIRE(USE)
- Old Plan table formats → DB2 V8 or 9 format (59 columns) PK85068

Old plans and packages V5 or before → REBIND

Plans containing DBRMs → packages PK62876

ACQUIRE(ALLOCATE) → ACQUIRE(USE) Please note that since ACQUIRE(ALLOCATE) is only for plans, once everything is packages, this will just happen automatically. Hence, no need to do anything to make this happen.

Old Plan table formats → DB2 V8 or 9 format (59 columns)

Items deprecated in earlier versions - NOW eliminated

- Private protocol → DRDA (new help in DSNTP2DP, PK92339, PK64045)
- XML Extender → XML type
- DB2 MQ XML UDFs and stored procedures → XML functions
- DB2 Management Clients feature → IBM Data Studio application & administration services
- BookManager use for DB2 publications → Info Center, pdf
- The DB2 Customization Center, DB2's plugin for the z/OS msys for setup installation initiative, is discontinued in DB2 Version 10 → install panels

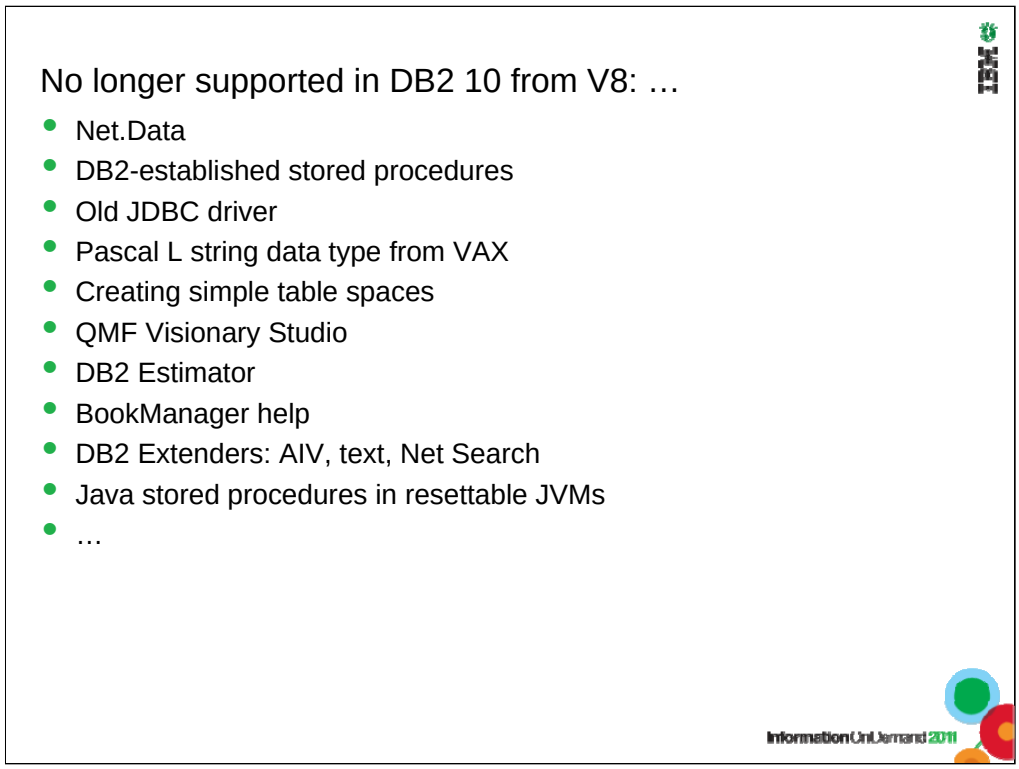
See the list of deprecated functions from prior versions.

The above features are still included in DB2 9 and may be dropped from future versions. Note the direction indicated to the right of the arrows, as these are the functions provided to replace the existing function. If you are using any of these functions, then move to the new function.

See the Installation Guide section, “Functions that are deprecated” and the announcement material for more information on these changes.

http://www.ibm.com/common/ssi/rep_ca/8/897/ENUS206-098/ENUS206-098.PDF

Private protocol → DRDA (new help in DSNTP2DP, PK64045)
XML Extender → XML type
DB2 MQ XML user-defined functions and stored procedures → XML functions
DB2 Management Clients feature (DB2 Administration Server, Control Center, & Development Center) → IBM Data Studio application & administration services
msys for Setup DB2 Customization Center → install panels
BookManager use for DB2 publications → Info Center, pdf
The DB2 Customization Center, DB2's plugin for the z/OS msys for setup installation initiative, was deprecated in DB2 Version 9.1 for z/OS and is discontinued in DB2 Version 10.



No longer supported …:

•Net.Data is removed. WebSphere is the strategic IBM solution for delivering DB2 data to Web applications.

•DB2-established stored procedure address spaces are no longer supported. Workload Manager (WLM) managed stored procedure address spaces is the strategic solution for stored procedure support, and migration to WLM managed stored procedure spaces is required for use of stored procedures in DB2 10.

•JDBC/SQLJ Driver for OS/390 and z/OS is no longer supported. All Java application programs and Java routines that are currently written to work with the JDBC/SQLJ Driver for OS/390 and z/OS need to be modified to work with the IBM DB2 Driver for JDBC and SQLJ (formerly known as the DB2 Universal JDBC Driver). The steps for migrating JDBC and SQLJ applications from the legacy JDBC/SQLJ Driver for OS/390 and z/OS to the IBM DB2 Driver for JDBC and SQLJ can be found in the Application Programming Guide and Reference for Java (SES1-3023). In addition, all WLM-managed stored procedures address spaces that are set up to execute Java routines must be modified to reference the IBM DB2 Driver for JDBC and SQLJ.

•Connections from VAX machines and the PASCAL L string data type are no longer supported.

•Creation of simple table spaces is no longer supported. DB2 10 for z/OS no longer implicitly creates simple table spaces nor allows customers to create simple table spaces. However, DB2 10 for z/OS continues to support simple table spaces created in previous versions.

•DB2 QMF Visionary Studio program is removed from DB2 QMF Enterprise Edition.

•DB2 Estimator is not available for DB2 10.

•BookManager-based online help has been removed. The prior help support has been replaced by the Information Management Software for z/OS Solutions Information Center (Information Center). The web-based Information Center is updated periodically during the life of each DB2 version, thus ensuring reader access to the most up-to-date information.

•AIV Extender, Text Extender, and Net Search Extender are removed.

•Java stored procedures no longer run in resettable JVMs.

Overview of the information center (IC)

- BookManager format is not supported for the DB2 V10 for z/OS library.
  - We offer an on-line information center that contains the information for supported versions.
  - As always, we support and provide PDF versions of our information.
  - We also offer an installable IC that you can run on your local system or intranet server. For ordering information, see:

http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/topic/com.ibm.dzic.doc/installabledzic.htm

- The IC is the IBM strategic direction for delivering information.
  - Easy to use! Easy to understand! Easy to find!
- The IC is a browser-based information portal—an industry-wide trend.
- The DB2 for z/OS IC has been available since V8 GA. Our IC is one of IBM's top visited ICs based on total number of hits!
- Current information for V8 and V9 is available at: http://publib.boulder.ibm.com/infocenter/imzic/

BookManager format continues to be supported for the V8 and V9 libraries. Although, documentation for the products in the DB2 Accessories Suite was never provided in BookManager format.

For V10, we are investigating whether we can continue to provide messages for the IBM LookAt tool. An efficient way to look up messages and codes is by using the information center. For more information, see:
http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/topic/com.ibm.support.dzic.doc/html/searchmsg.htm

Improvements to installation & migration information based on customer feedback

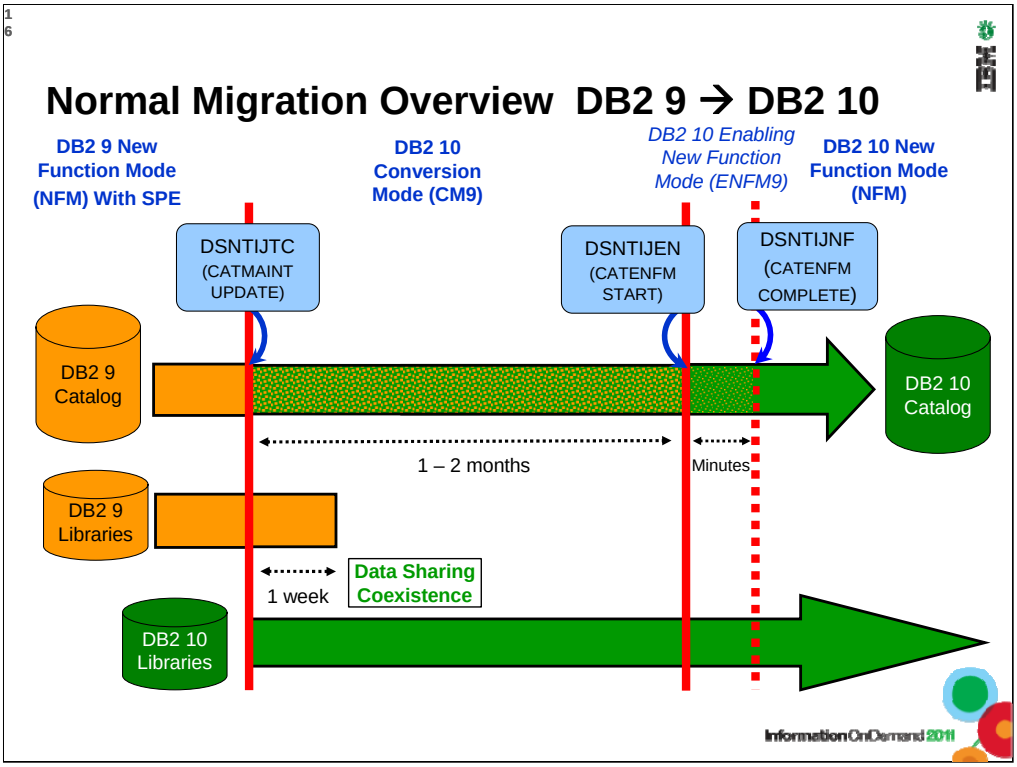
- Pre-migration and migration checklists were added to help plan for and keep track of the migration process. The checklists include links to the pre-migration and migration steps. Access the checklists from the following locations:
  - Introduction to migration from DB2 Version 8: http://publib.boulder.ibm.com/infocenter/ecsimzic/v1r0/topic/com.ibm.db2z10.doc.inst/db2z_intro2migrfromv8.htm
  - Introduction to migration from DB2 Version 9.1: http://publib.boulder.ibm.com/infocenter/ecsimzic/v1r0/topic/com.ibm.db2z10.doc.inst/db2z_intro2migrfromv9.htm
- Installation and migration steps clearly define tasks to complete. Related concept and reference information are included as links from the tasks.
- Several new jobs simplify the setup and installation of DB2-supplied stored procedures and user-defined functions. The process is documented in Installation step 20: Set up DB2-supplied routines (optional) and Migration step 25: Set up DB2-supplied routines (optional).

For the V10 development cycle, the information development team focused on addressing solutions to top customer pain points. Based on customer feedback, one of the areas we focused on improving is the installation and migration information.

DB2 10 for z/OS

Migration Paths Available

15



Migration from DB2 9 to DB2 10

The catalog changes will happen in two places. One is the migration from DB2 9 to DB2 10 conversion mode (CM9) using the DSNTIJTC job. The other is the DB2 10 enabling-new-function mode process (ENFM9) using the DSNTIJEN job.

When a system is migrating from DB2 9 NFM to DB2 10 conversion mode the possible DB2 10 modes are:

CM9 Conversion Mode entered when migrating from DB2 9 NFM to DB2 10

ENFM9 Enabling New Function Mode on a system that migrated from DB2 9 NFM to DB2 10. Once this mode has been entered the system can not fallback to DB2 9 and a DB2 9 member can not be started in a data sharing group.

NFM This is the NFM mode when all system changes are made on a system that migrated from DB2 9 NFM to DB2 10 and the system is ready for DB2 10 new function

CM9* The system migrated from DB2 9 NFM to DB2 10 and at one point was in either ENFM9 or NFM on DB2 10.

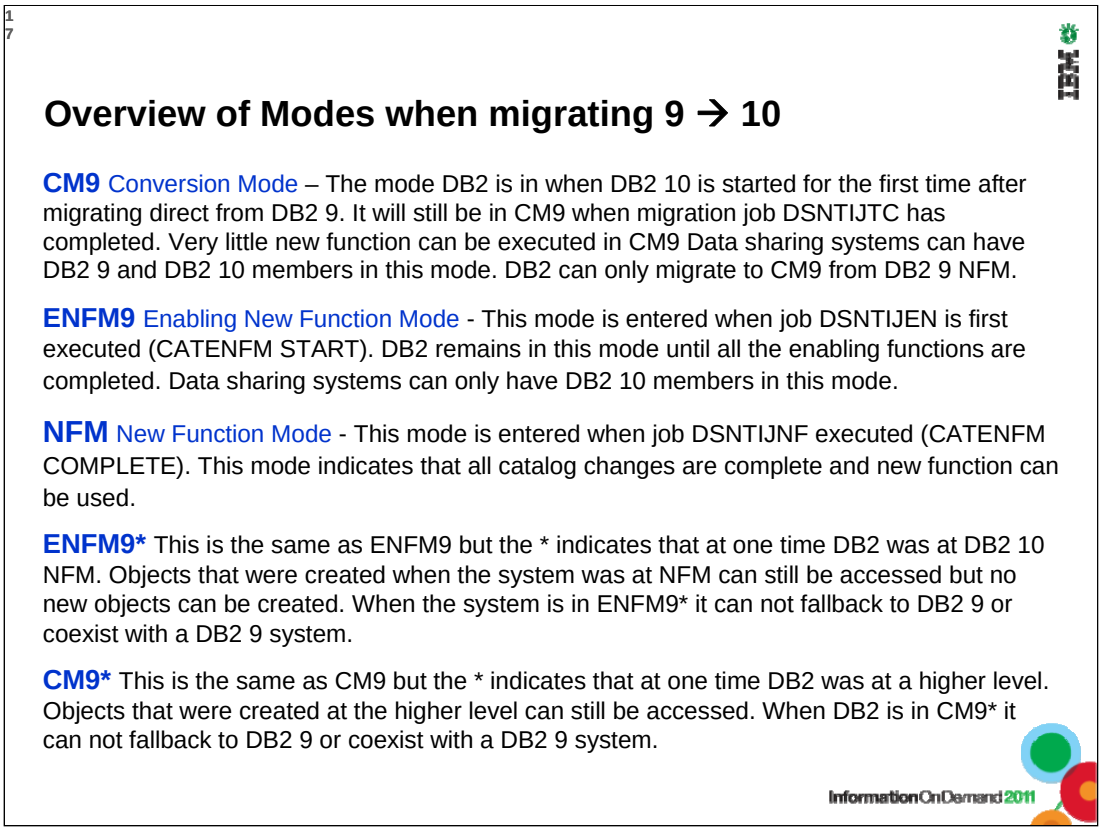
ENFM9* The system was migrated from DB2 9 NFM to DB2 10 and at one point was in NFM on DB2 10.

Some migration considerations are:

- A DB2 9 system that has started the migration to DB2 10 can only fallback to DB2 9.
- A data sharing group that migrated from DB2 9 NFM to DB2 10 can not have any V8 members.

Note – this is not necessarily to scale!

Note – ONE WAY – fallback to CM9* is possible but not to DB2 9



Enabling New Function Mode 9 (ENFM9) is entered when job DSNTIJEN is first executed (CATENFM START), following a previous migration DSNTIJTC directly from DB2 9 to DB2 10. DB2 remains in this mode until all the enabling functions are completed. Data sharing systems can only have DB2 10 members in this mode. The enabling functions are:

- Place the DB2 subsystem in enabling-new-function mode 9 (ENFM9).
- The first time that you run job DSNTIJEN, DB2 saves the RBA or LRSN of the system log in the BSDS.

ENFM9* is the same as ENFM but the * indicates that the at one time the system was at NFM. Objects that were created when the system was at NFM can still be accessed but no new objects can be created. When the system is in ENFM9* it can not fallback to DB2 9 or coexist with a DB2 10 system. When DB2 is in ENFM9* the mode can be changed with the following:

- Migration job DSNTIJCS takes the system to CM9*.
- Migration job DSNTIJNF takes the system to DB2 10 NFM.

CM9* is the same as CM but indicates that at one time DB2 was at a higher level. Objects created at the higher level can still be accessed. When in CM9*, DB2 cannot fallback to DB2 9, or coexist with a DB2 9 system. CM9* is entered when DSNTIJCS is run and DB2 is in ENFM9, ENFM9* or NFM. When DB2 is in CM9*, the mode can be changed as follows:

Migration job DSNTIJEN

- If DB2 went from NFM or ENFM9* to CM9*, DSNTIJEN takes it to ENFM9*.
- If DB2 went from ENFM9 to CM9*, it completes ENFM9 processing and leaves DB2 in ENFM9.

Migration job DSNTIJNF

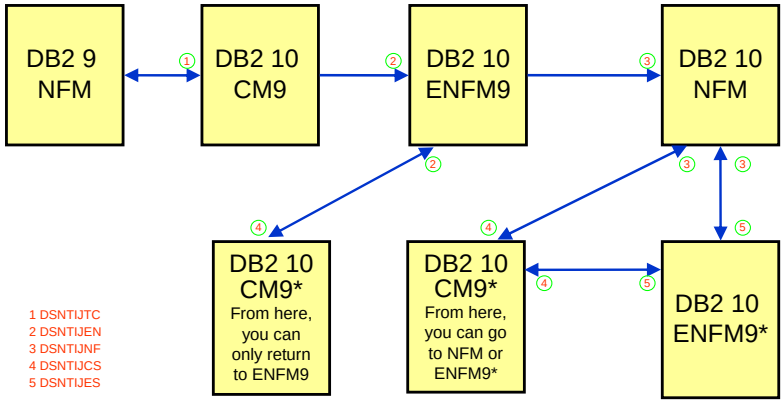
- If DB2 went from NFM or ENFM9* to CM9*, DSNTIJNF takes it to NFM.
- If DB2 went from ENFM9 to CM9*, DSNTIJNF does one of two things:
  - If ENFM9 processing had completed it takes DB2 to NFM.
  - If not, DSNTIJNF indicates that DSNTIJEN must be run.

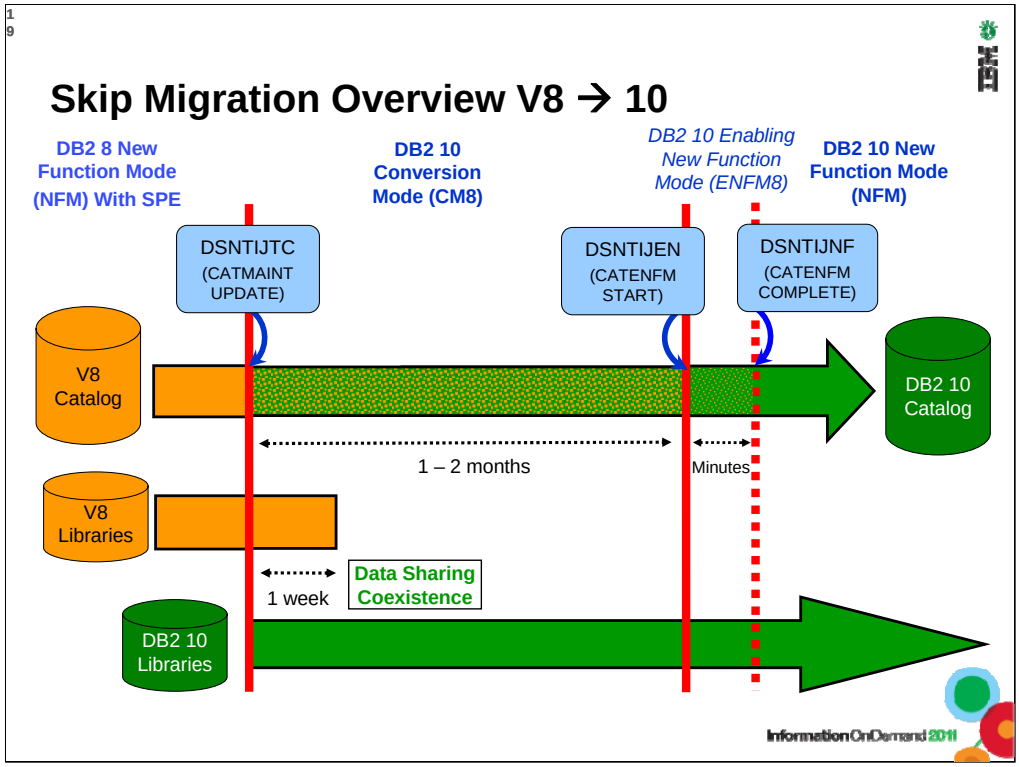
Migration job DSNTIJES

- If DB2 went from NFM or ENFM9* to CM9*, DSNTIJES takes it to ENFM9*.
- If DB2 went from ENFM9 to CM9*, DSNTIJES indicates that DSNTIJEN must be run.

Migration and Fallback Paths when migrating (V9 ==> V10)

- With DB2 10, you can always drop back to the previous stage
- Cannot fallback to DB2 9 after entry to DB2 10 (ENFM9), but can return to DB2 10 (CM9*)





Migration from V8 directly to 10

The catalog changes will happen in two places. One is the migration from DB2 V8 to DB2 10 conversion mode (CM8) using the DSNTIJTC job. The other is the DB2 10 enabling-new-function mode process (ENFM8) using the DSNTIJEN job.

DB2 will support migrating from V8 NFM to DB2 10 without ever starting the system in DB2 9. When a system is migrating from V8 NFM to DB2 10 conversion mode the possible DB2 10 modes are:

CM8 Conversion Mode entered when migrating from V8 NFM to DB2 10

ENFM8 Enabling New Function Mode on a system that migrated from V8 NFM to DB2 10. Once this mode has been entered the system can not fallback to V8 and a V8 member can not be started in a data sharing group.

NFM This is the New Function Mode when all system changes have been made on a system that migrated from V8 NFM to DB2 10 and the system is ready for DB2 10 new function

CM8* The system migrated from V8 NFM to DB2 10 and at one point was in either ENFM8 or NFM on DB2 10.

ENFM8* The system was migrated from V8 NFM to DB2 10 and at one point was in NFM on DB2 10.

Some migration considerations are:

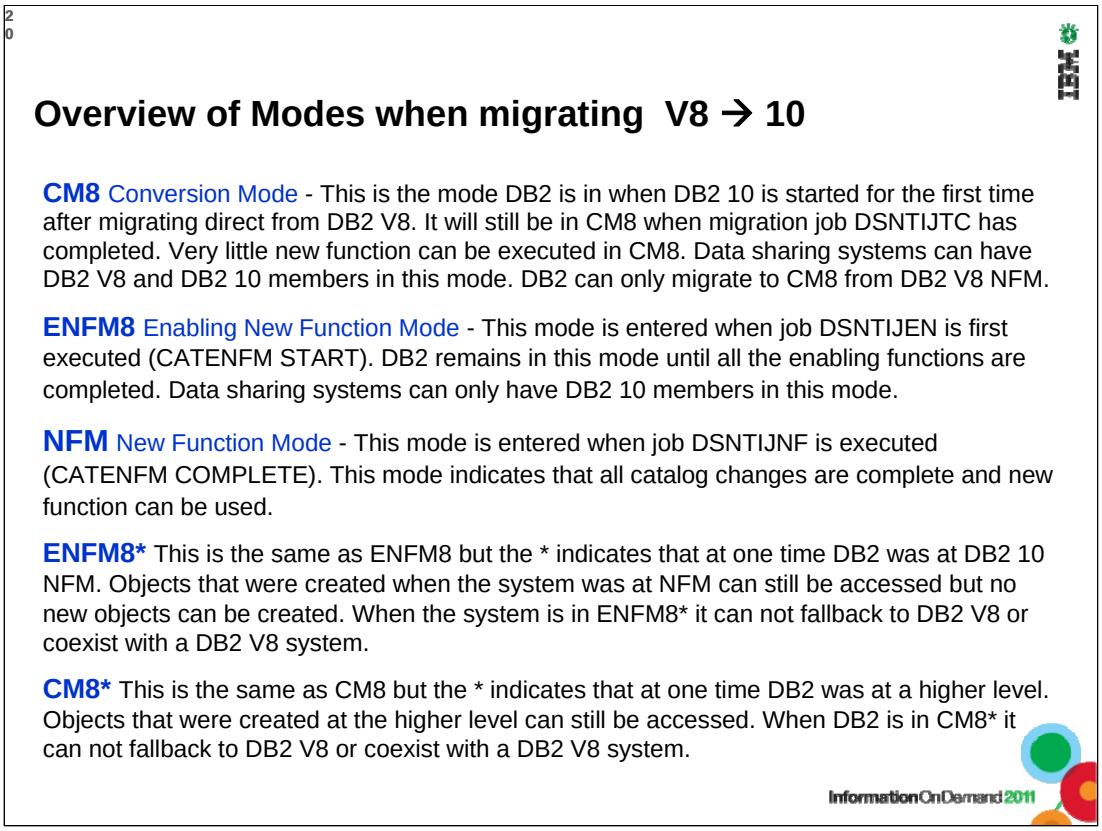
- A V8 system that has started the migration to DB2 10 can only fallback to V8.
- A V8 system that has started the migration to DB2 10 and then performed a fallback to V8 can not migrate to DB2 9.
- A system that migrates from V8 NFM to DB2 10 can not use DB2 9 new function until DB2 10 NFM is reached.
- A data sharing group that migrated from V8 NFM to DB2 10 skipping DB2 9 can not have any DB2 9 members.

Some differences in the C8 and C9 catalog are:

- The RTS is still in a user database.
- The SEQNO column of SYSPACKSTMT is a SMALLINT.
- The SYSOBJ table space uses 4k pages in CM8 and 8K pages in CM9.

Note – this is not necessarily to scale!

Note – ONE WAY – fallback to CM8* is possible, but not to DB2 V8



Enabling New Function Mode 8 (ENFM8) is entered when job DSNTIJEN is first executed (CATENFM START), following a previous migration DSNTIJTC directly from DB2 V8 to DB2 10. DB2 remains in this mode until all the enabling functions are completed.

Data sharing systems can only have DB2 10 members in this mode. The enabling functions are:

- Place the DB2 subsystem in enabling-new-function mode 8 (ENFM8).
- The first time that you run job DSNTIJEN, DB2 saves the RBA or LRSN of the system log in the BSDS.

ENFM8* is the same as ENFM but the * indicates that the at one time the system was at NFM. Objects that were created when the system was at NFM can still be accessed but no new objects can be created. When the system is in ENFM8* it can not fallback to DB2 V8 or coexist with a DB2 10 system. When DB2 is in ENFM8* the mode can be changed with the following:

- Migration job DSNTIJCS takes the system to CM8*.
- Migration job DSNTIJNF takes the system to DB2 10 NFM.

CM8* is the same as CM but indicates that at one time DB2 was at a higher level. Objects created at the higher level can still be accessed.

When in CM8*, DB2 cannot fallback to DB2 V8, or coexist with a DB2 V8 system. CM8* is entered when DSNTIJCS is run and DB2 is in ENFM8, ENFM8* or NFM. When DB2 is in CM8*, the mode can be changed as follows:

Migration job DSNTIJEN

- If DB2 went from NFM or ENFM8* to CM8*, DSNTIJEN takes it to ENFM8*.
- If DB2 went from ENFM8 to CM8*, it completes ENFM8 processing and leaves DB2 in ENFM8.

Migration job DSNTIJNF

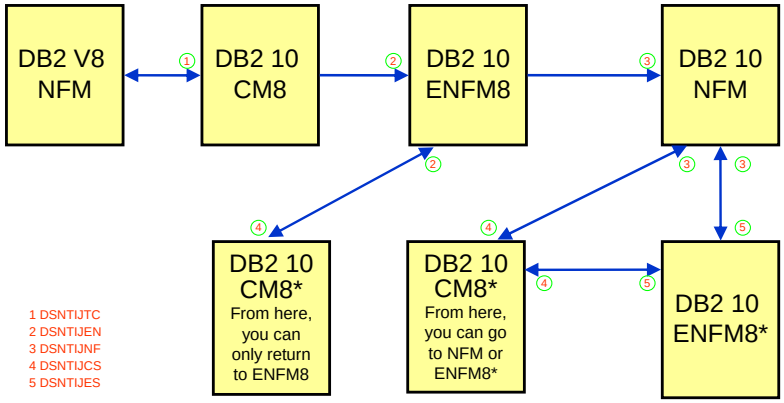
- If DB2 went from NFM or ENFM8* to CM8*, DSNTIJNF takes it to NFM.
- If DB2 went from ENFM8 to CM8*, DSNTIJNF does one of two things:
    - If ENFM8 processing had completed it takes DB2 to NFM.
    - If not, DSNTIJNF indicates that DSNTIJEN must be run.

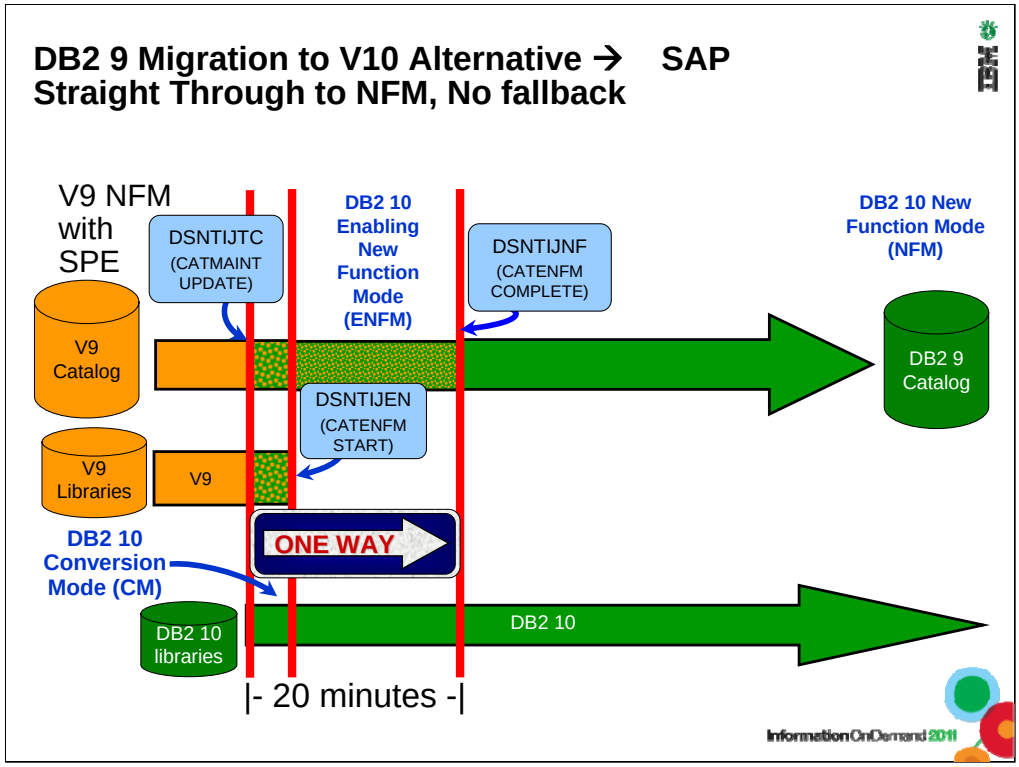
Migration job DSNTIJES

- If DB2 went from NFM or ENFM8* to CM8*, DSNTIJES takes it to ENFM8*.
- If DB2 went from ENFM8 to CM8*, DSNTIJES indicates that DSNTIJEN must be run.

Migration and Fallback Paths when migrating (V8 ==> V10)

- With DB2 10, you can always drop back to the previous stage
- Cannot fallback to DB2 V8 after entry to DB2 10 (ENFM8), but can return to DB2 10 (CM8*)

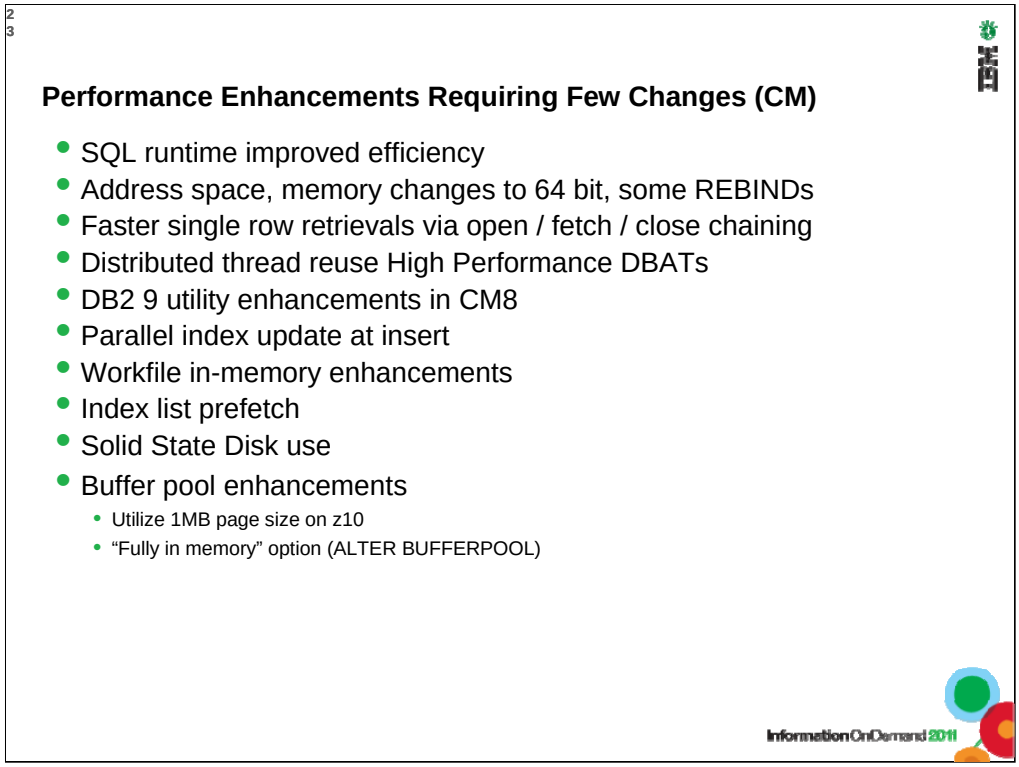




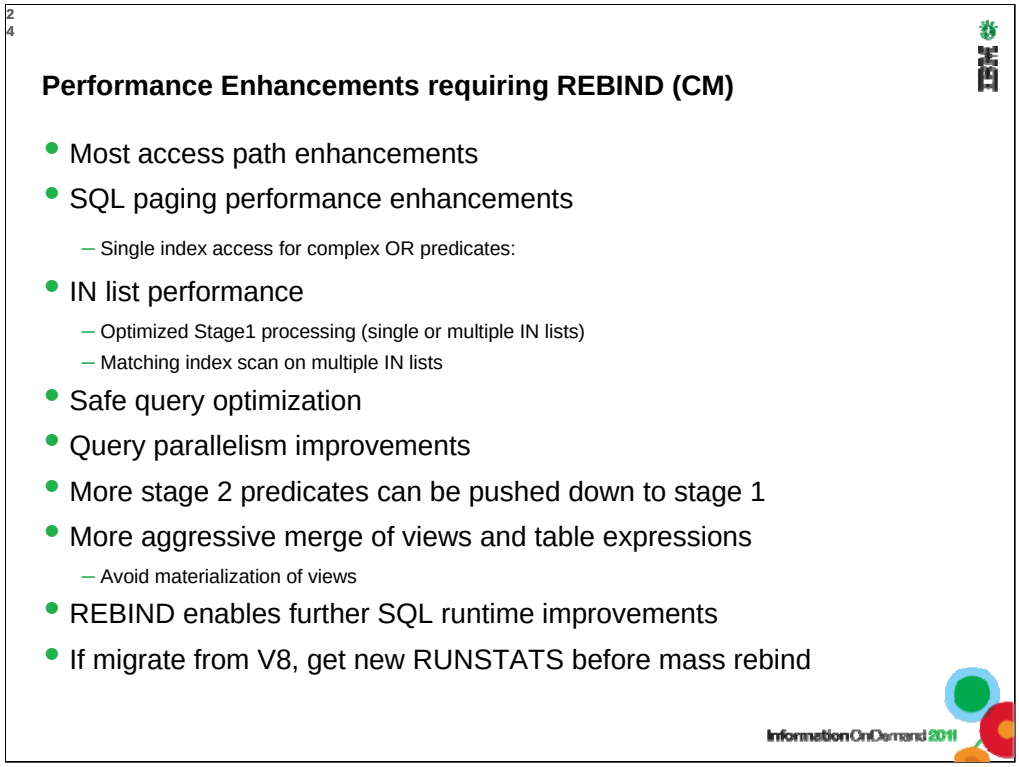
Some customers, such as those who use SAP, will use an alternative migration process, migrating immediately from V9 to DB2 10 NFM. This process means that fallback to V9 is not possible, so thorough testing is needed. The advantages are

- single window for the migration
- test in only one mode, NFM
- faster delivery of improved value in performance, function and availability

Some vendors, such as SAP, use this process. Customers who cannot allow multiple windows for migration need to consider this alternative and the work and controls needed to migrate while processing continues. This process is like Chinese cooking, three minutes to cook, after three hours of preparation. If you do months of planning and testing, ensure that you don't need to fall back, then the steps can be compressed from two to one.



These are the improvements which we expect almost every customer to see as soon as DB2 10 is running, even in conversion mode.
Many paths within DB2 processing leverage better 64-bit memory capabilities. This results in better SQL performance for many existing SQL access plans.
All of the memory improvements provide immediate relief for all of many memory constrained systems. With some REBINDS, memory usage is reduced, allowing you to use memory more effectively for example in buffer pools for performance in your environment.
Better handling of singleton Cursor Selects through chaining, combines the open fetch and close work ONLY once across the network improving network bandwidth efficiency and overall performance.
Also for DDF transactions, there is increased DDF thread reuse. This enhancement starts to handle and reuse DDF threads though the same methods that we have had for a long time through CICS thread interfaces for robust consistent type transactions.
For people that are directly coming to DB2 10 through migrations from Version 8 they immediately get long list of the Version 9 enhancements especially all the Utility performance improvements. Some are experiencing 20% elapse time savings in Version 9.
Updates to index columns are done in parallel in DB2 10 improving insert performance. This out of the box enhancement along with DB2 10 List Pre-fetch capabilities improve all existing applications that use list pre-fetch activities across all existing accesses paths and especially when using indexes that may be a little disorganized and in need of a reorg.
DB2 can now use solid state disk devices. These devices are great for the workfiles, GTTs and other high performance table spaces within your environment.
Also, the enhanced way DB2 10 uses in-memory Workfiles and the improvements related to RID pool overflows helps all application systems avoid the deadly table space scan at the peak processing times.
DB2 can utilize the new bigger 1MB page size on z10 and provide additional buffer pool options to put a table fully in memory with an easy simple table space ALTER.
CPU times are reduced for SQL running transactions and batch which are generally the peak customer workload. These techniques take very little change, but the buffer pool enhancements do need an ALTER BUFFERPOOL command.



Rebind is required for a long list of improvements in optimization and parallelism. The key improvements from REBIND in CM include SQL IN list improvements, SQL paging enhancements, query parallelism improvements, and more aggressive view and table expression merge.

Improvements with WHERE OR clauses that have columns that all reference the same index can be optimized for a single Stage 1 evaluation instead of being evaluated multiple times and then retrieved multiple times. This consolidates and improves performance dramatically for processing.

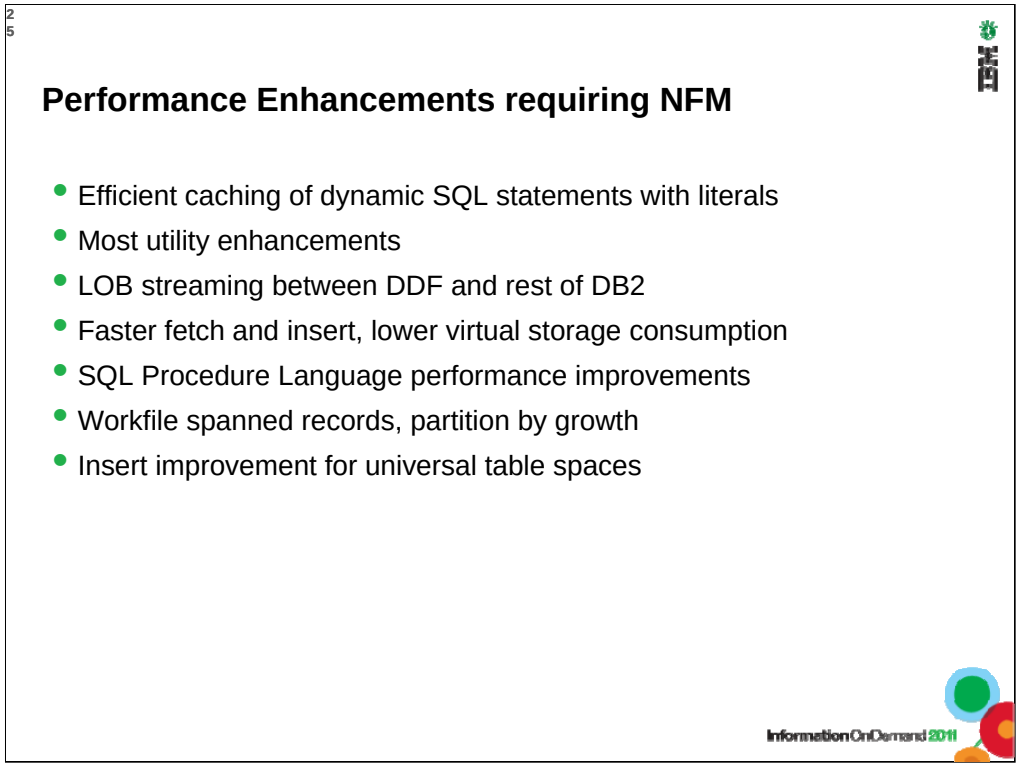
IN-List predicates are now evaluated through Stage 1 processing and provide a matching index access of multiple IN-list Where clause criteria. This is common and great for applications that have cursor pagination within their application.

More SQL query parallelism offloads precious CPU main engine cycles and pushes more processing into your specialty zIIP engines.

REBINDS also let DB2 push down more predicate evaluations from Stage 2 to Stage 1 during data retrieval. This helps reduce the rows evaluated in each step of the SQL access path improving performance significantly for complex access paths.

This also comes into play with View and expression materialization. Since the amount of data through the materialized steps can be sometimes big, cutting down the amount of data in Stage 1 processing interim result sets can really help performance of these complex SQL statement and their processing situations.

If you are migrating from DB2 V8, then you will want to get improved statistics for cluster ratio, data repeat factor and high cardinality non-uniform distribution of data by running RUNSTATS before you REBIND.



Some of the performance improvements require new function mode and some work by database administrators to tune the database design and often to REBIND. Efficient caching for literals needs a rebind in NFM.

One of the enhancements that will help everyone with large dynamic SQL applications are the improved processes that handle the Dynamic Statement Cache. They now consolidate SQL statements that are the same but have different literals. This reduces the SQL statement space used in the Dynamic Statement Cache and reuses the security and object verification and access path already developed for the SQL statement. This dramatically improves Dynamic Statement Cache cache hits, reduces the duplicate SQL statement previously held, makes room for more SQL improving the overall workload performance.

Also all the concurrency and performance enhancements through the DB2 10 Utilities improvements are available for more on-line ALTER and maintenance activities.

Better streaming and minimized LOB default sizes helps improve when LOB materialization is happening with the system and this is especially important for DDF type applications because of the network impact of large objects.

Small WORKFILEs are now available for simple predicate evaluation for improving performance

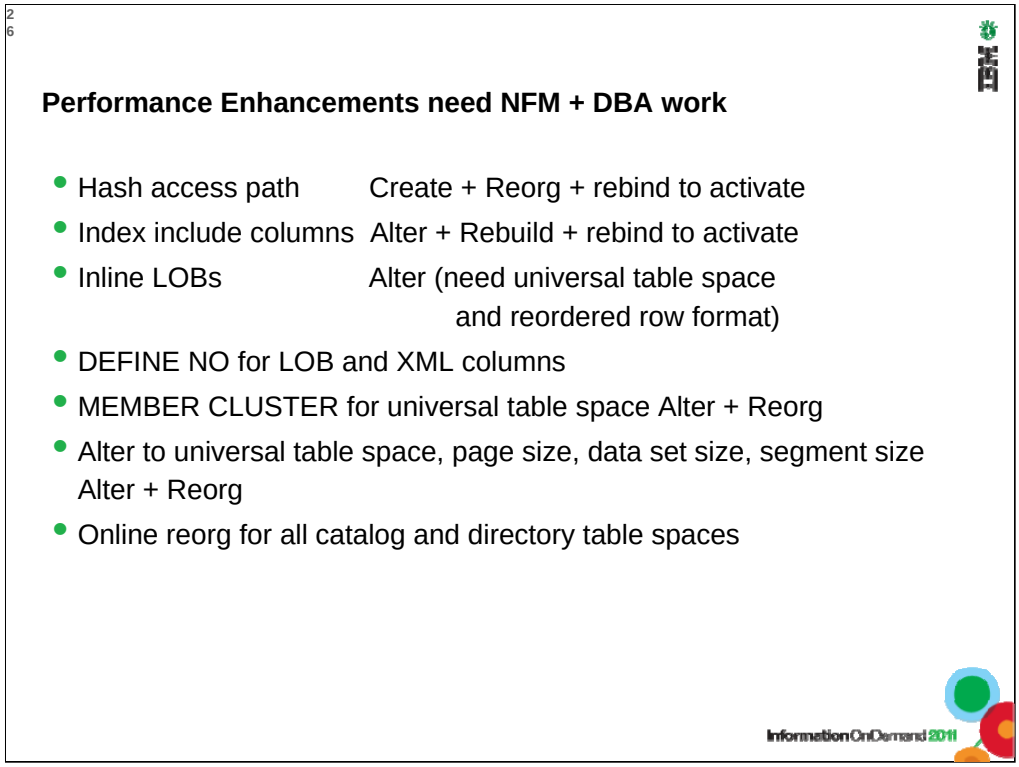
DB2 provides native support for the SQL procedural language eliminating the cumbersome requirement to generate a C program from the SQL procedure that would then execute as an external stored procedure.

DB2 10 SQL procedures are better optimized to execute more efficiently more common constructs are optimized within the DB2 code making SQL procedures very efficient for performance within the SQL procedure language.

WORKFILE can have expanded records up to 65K so larger Joins and answers set can be generated from DB2.

DB2 10 supports partition-by-growth table spaces in the WORKFILE database and provides in-memory work file enhancements in the WORKFILE database.

In the WORKFILE database, DB2 supports simple predicate evaluation for work files. This enhancements reduces the CPU time for workloads that execute queries that require the use of small work files.



Some of the performance improvements require new function mode and some work by database administrators to tune the database design and often to REBIND. The DBAs can improve on their direct keys access and implement the new Hash access table space type reducing a key access to potentially a single I/O. This is great for the customer or product id lookouts that are done millions or billions of times a day.

The Index include non-unique columns within a UNIQUE index definition provides great way to eliminate indexes and consolidate other indexes for better optimization and overall performance. This is great also for better access paths because all the columns are now in the single index for better access paths and fewer indexes to reference.

Inline LOBs use the reorder row format and handle the LOB better for overall streaming and application performance. This along with the DEFINE NO feature allows the row to be used and the data set for the LOB not to be defined. The application can still reference the row and get the proper results. The system doesn't define the LOB data set until the LOB is saved which can save a lot of space for tables that only sometimes had LOB entries.

The MEMBER CLUSTER setting can help eliminate contention and the clustering requirements within a data sharing environment table. This boosts performance and relieves contention across your data-sharing environment.

And now almost any attribute within the definition of the table space can be Altered within DB2 and then applied through an on-line reorg. This is great for application availability to keep your business rolling with DB2.

Things to be considered when “skipping” past V9?

- DB2 9 for z/OS education
  - Migration Planning workshop materials
  - Redbooks
  - Transition class
- Check for DB2 9 release incompatibilities
  - DB2-managed stored procedures -> WLM-managed SPs
- Familiarity with the new modes
  - CM*, ENFM* introduced in V9
- Deprecated items
  - Create simple table space -> segmented or partition by growth
- Performance benefits
- Storage changes
  - Temp databases & table spaces -> work files (favor 32K)
- Utilities updates
- Key enhancements e.g. pureXML, Native SQLPL
- Comprehensive regression test plan
- Plan for more frequent maintenance

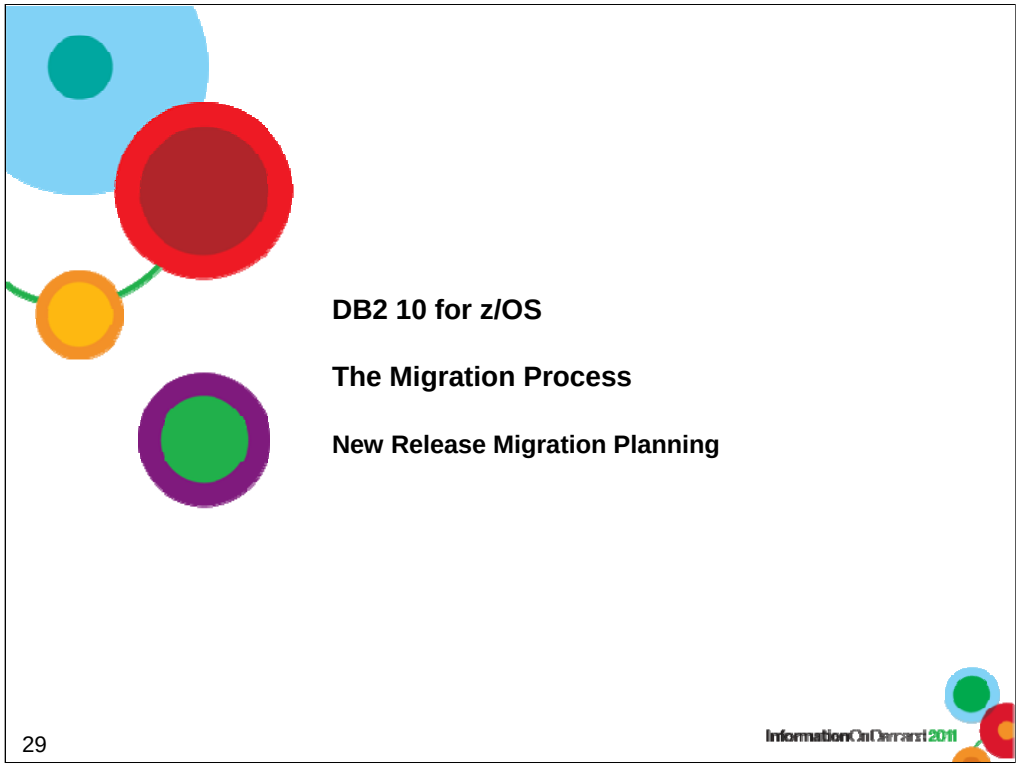
Migration considerations to REMEMBER

- A V8 system started migration to V10, can only fallback to V8.
- **A V8 system started the migration to V10, then performed fallback to V8, can not then migrate to V9**.
- A V8 system started the migration to V9, then performed fallback to V8, can not then migrate to V10.
- A V8 system migrating to V10, can not use V9 new functions until V10 NFM is reached.
- A data sharing group that started migrating from V8 to V10 can not have any V9 members.
- A V9 system that has started the migration to V10 can only fallback to V9.
- A data sharing group that started migration from V9 NFM to V10 can not have any V8 members.

Some migration considerations are:

- A V8 system that has started the migration to V10 can only fallback to V8.
- A V8 system that has started the migration to V10 and then performed a fallback to V8 can not migrate to V9.
- A system that migrates from V8 NFM to V10 can not use V9 new function until V10 NFM is reached.
- A data sharing group that migrated from V8 NFM to V10 skipping V9 can not have any V9 members.

- A V9 system that has started the migration to V10 can only fallback to V9.
- A data sharing group that migrated from V9 NFM to V10 can not have any V8 members.


DB2 10 for z/OS
The Migration Process
New Release Migration Planning
29
InformationOnDemand2011

Preparing your current DB2 for Migration to V10 CM

- Apply the Fallback SPE APAR, **PK56922** and any prerequisite fixes
  - **Your DB2 (V8 or V9) systems *MUST* be at the proper service level**
  - **See Info APARs II14477 (9) or II14474 (8)**
- Non-Data Sharing
  - Current DB2 (V8 or V9) must be started with the SPE applied, or migration to DB2 10 will terminate.
- Data Sharing
  - Before migrating a member to DB2 10, all other started DB2 (V8 or V9) members must have the fallback SPE applied.
  - The fallback SPE must be on all active DB2 (V8 or V9) group members for DB2 10 to start.

Important – Apply SPE to ALL Data Sharing Members Before Starting Migration!

DB2 Version 10 Migration Considerations

Before attempting to migrate to DB2 Version 10, make sure that the release being migrated from is at the proper service level. A migration to DB2 Version 10 places the DB2 subsystem into DB2 Version 10 Conversion Mode. In Conversion Mode, no new DB2 Version 10 function is available for use that would preclude a successful fallback to the previous release.

Ensure you apply the fallback SPE APAR, PK56922 to your current DB2 (V8 or V9) subsystems.

Code Level Checking:

Non-Data Sharing

At DB2 startup time, the code level of the starting DB2 will be checked against the code level required by the current DB2 catalog. If the starting DB2 has a code level mismatch with the catalog then the DSNX208E message will be issued and DB2 will not start. A code level mismatch indicates that the starting DB2 is at a level of code that is down level from what it needs to be for the current catalog.

If the catalog has been migrated to DB2 Version 10 then the starting DB2 must be an allowed DB2 Version with the appropriate fallback SPE on.

Before attempting to migrate to DB2 Version 10, it's necessary to run with a maintenance level on the previous DB2 Version with the fallback SPE prior to attempting to migrate to DB2 Version 10. If the fallback SPE was not on the previous DB2 Version and DB2 started, before attempting to migrate to DB2 Version 10, then the DB2 Version 10 migration will be terminated.

Data Sharing

At DB2 startup time, the code level of the starting DB2 will be checked against the code level required by the current DB2 catalog and against the code level of the other DB2s that are active. If the starting DB2 has a code level mismatch with the catalog or any of the other DB2s that are running then either the DSNX208E or DSNX209E message will be issued and DB2 will not start.

A code level mismatch indicates that the starting DB2 is at a level of code that is down level from what it needs to be for the current catalog or that one or more of the already running DB2s are down-level from where they need to be.

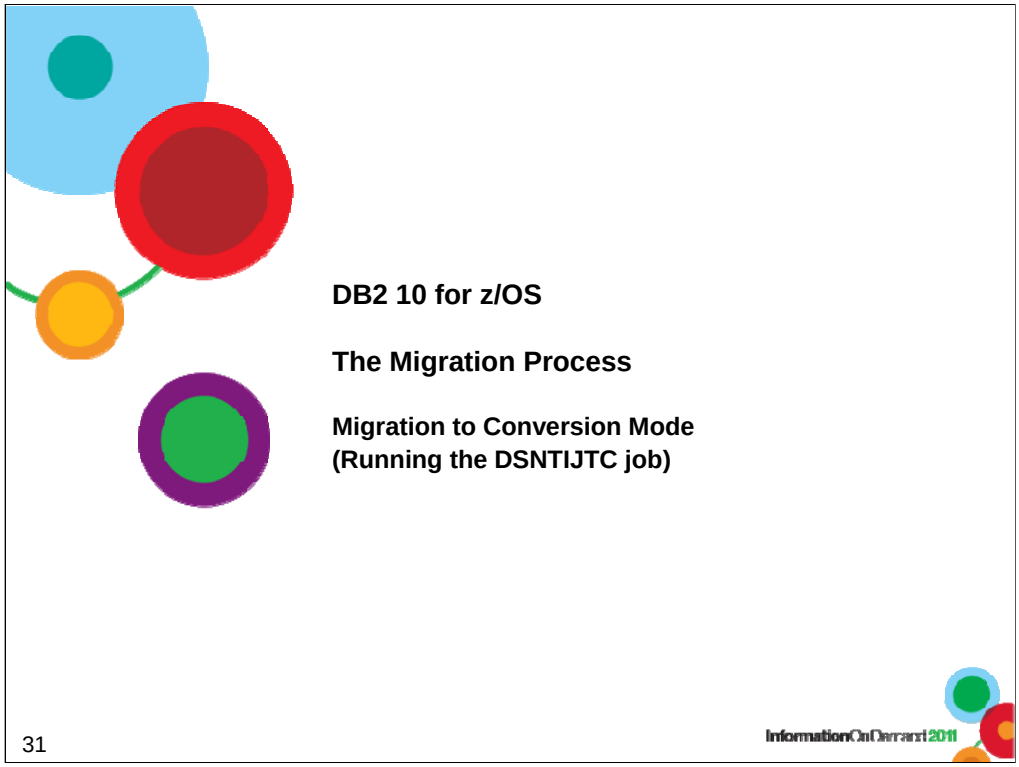
Before attempting to migrate to DB2 Version 10, all started DB2 subsystems must have maintenance through the DB2 Version 10 fallback SPE on before any attempt is made to migrate any member to DB2 Version 10. If the fallback SPE is not on all active group members then DB2 Version 10 will not start and you'll not be able to attempt the DB2 Version 10 migration. Message DSNX208E or DSNX209E will be issued in these cases.

It's recommended that only one DB2 subsystem be started at DB2 Version 10 in Data Sharing for migration processing. Once the migration to DB2 Version 10 completes, the other group members can then be brought up to DB2 Version 10 at any time.

During a migration to DB2 Version 10, the other group members may be active. DSNTIJTC processing will get whatever locks are necessary for the processing it needs to do. The other active group members may experience delays and/or timeouts if they try to access the catalog objects that are being updated or locked by migration processing.

So what about coexistence?

DB2 V8 NFM and DB2 10 CM8 will coexist OR DB2 V9 NFM and DB2 10 CM9 will coexist.


DB2 10 for z/OS
The Migration Process
Migration to Conversion Mode
(Running the DSNTIJTC job)
31
InformationOnDemand2011

Migration to Conversion Mode

- Run the **DSNTIJTC** job to migrate:
  - From DB2 for z/OS Version 8 to DB2 10 for z/OS Conversion Mode (**CM8**)
  - From DB2 9 for z/OS to DB2 10 for z/OS Conversion Mode (**CM9**)
    - Authorization check (INSTALL SYSADM)
    - Verify Catalog is at correct level for migration
    - New release DDL
    - Additional Catalog updates
    - Update Directory header page and BSDS/SCA with new release information

```
DSNU000I  194 14:58:16.06 DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = RELODCAT
DSNU1044I 194 14:58:16.08 DSNUGTIS - PROCESSING SYSIN AS EBCDIC
DSNU050I  194 14:58:16.49 DSNUGUTC -  CATMAINT UPDATE
DSNU750I  194 14:58:16.59 DSNUECM0 - CATMAINT UPDATE PHASE 1 STARTED
DSNU777I  194 14:58:16.59 DSNUECM0 – CATMAINT UPDATE STATUS – VERIFYING CATALOG IS AT CORRECT LEVEL FOR MIGRATION.
DSNU777I  194 14:58:17.29 DSNUECM0 – CATMAINT UPDATE STATUS – BEGINNING MIGRATION SQL PROCESSING PHASE.
DSNU777I  194 14:58:43.15 DSNUEXUP – CATMAINT UPDATE STATUS – BEGINNING ADDITIONAL CATALOG UPDATES PROCESSING.
DSNU777I  194 14:58:43.16 DSNUEXUP – CATMAINT UPDATE2 STATUS – BEGINNING SYSCOPY TABLE SPACE MIGRATION PROCESSING.
DSNU777I  194 14:58:43.16 DSNUECM0 – CATMAINT UPDATE STATUS – UPDATING DIRECTORY WITH NEW RELEASE MARKER.
DSNU752I  194 14:58:43.18 DSNUECM0 - CATMAINT UPDATE PHASE 1 COMPLETED.
DSNU010I  194 14:58:43.23 DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0
```


DSNTIJTC job - Migration to DB2 10 Conversion Mode (CM8 or CM9)

The DSNTIJTC job invokes the CATMAINT UPDATE utility in order to migrate the current (V8 or V9) catalog and directory to DB2 10 for z/OS. The end result of this migration process is a current DB2 Version 10 catalog.

In DB2 Version 10, DB2 will continue to enforce a restriction that there must be no outstanding utilities started from prior releases when running CATMAINT UPDATE on a non data-sharing system, otherwise message DSNU790I will be issued.

The CATMAINT UPDATE utility performs some or all of the following tasks during any given migration:

- Adds entries in the catalog and directory for new catalog objects:
- Table Spaces
- Tables
- Indexes
- Table Check Constraints
- Referential Constraints
- Adds new columns to existing catalog tables.
- Adds additional values to existing catalog columns.
- Changes the definitions of existing catalog tables.
- Makes additional updates to the migrated catalog and directory.
- Additional updates.

Unique to each migration are additional changes that need to be made to the existing catalog and directory.

SMS is now a prerequisite of V10. During the process of migrating from V8/V9 to V10 conversion mode, all the new indexes and new table spaces in the catalog and directory will be created as SMS-controlled.

SMS now a prerequisite

Important – All new indexes and new table space data sets WILL be SMS controlled.

"SMS now a prerequisite of DB2 10".

So be prepared!

During migration from (V8 or V9) to V10 conversion mode, all the new indexes and new table spaces in the catalog and directory will be created as SMS-controlled with the extended addressability (EA) attributes.

- DSNTIJSS sample provides SMS classes for customers without SMS in use.
  - The environment created by DSNTIJSS is ONLY for DB2 Catalog and Directory data sets.
    - Other DB2 data sets such as logs and BSDS not covered.
    - On developerWorks "DB2 for z/OS Exchange", "samples"
- Useful SMS reference material includes:
  - z/OS DFSMS Introduction.
  - z/OS DFSMS Implementing System Managed Storage.
  - z/OS DFSMS Storage Administration Reference.
  - The Redbook: Maintaining your SMS environment

SMS is now a prerequisite of V10.

During the process of migrating from V8/V9 to V10 conversion mode, all the new indexes and new table spaces in the catalog and directory will be created as SMS-controlled.

The environment created by the DSNTIJSS sample is only for DB2 Catalog and Directory data sets, which MUST be SMS managed in DB2 V10. Other DB2 subsystem data sets such as logs and the BSDS are not accounted for in this environment. The DSNTIJSS sample is on developerWorks. Once you login to developerWorks, go to "DB2 for z/OS Exchange", "samples".

Useful SMS reference material includes:

- z/OS DFSMS Introduction.
- z/OS DFSMS Implementing System Managed Storage.
- z/OS DFSMS Storage Administration Reference.
- The Redbook: Maintaining your SMS environment.

DB2 Catalog Evolution

The DB2 catalog continues to grow with every DB2 release.

DB2 Version	Table Spaces	Tables	Indexes	Columns	LOB Columns	Table Check Constraints
V1	11	25	27	269	0	N/A
V3	11	43	44	584	0	N/A
V5	12	54	62	731	0	46
V6	15	65	93	967	0	59
V7	20	84	118	1212	2	105
V8	22	85	132	1265	2	105
V9	28	104	165	1643	3	119
DB2 10	**95 (104-9)**	**134**	**233**	**1922**	**18**	**119**

Does not include objects for XML Schema Repository

Complete release summary of V10 changes (CM + ENFM)

60(2+11+47) new table spaces, 7 old table spaces dropped.

18(3+11+4) new tables

44(22+11+11) new indexes

33(25+8) new columns (on old tables)

Full details

Here is the list of changes during DSNTIJTC to migrate to V10 CM9

25 new columns

22 new index

2 new table spaces

3 new tables

no change in check constraints or new check constraints.

changing the meaning of the catalog columns: there is nothing to change in terms of DSNTIJTC, nothing is done in DSNTIJTC to change the meaning.

it is just a change of how other db2 components use the columns and decide what values to put in or how to interpret the values..

DSNTIJTC doesn't actively change the values from A to B.

During the ENFM, we create more objects:

8 additional new columns

11 new LOB table spaces

11 new aux tables

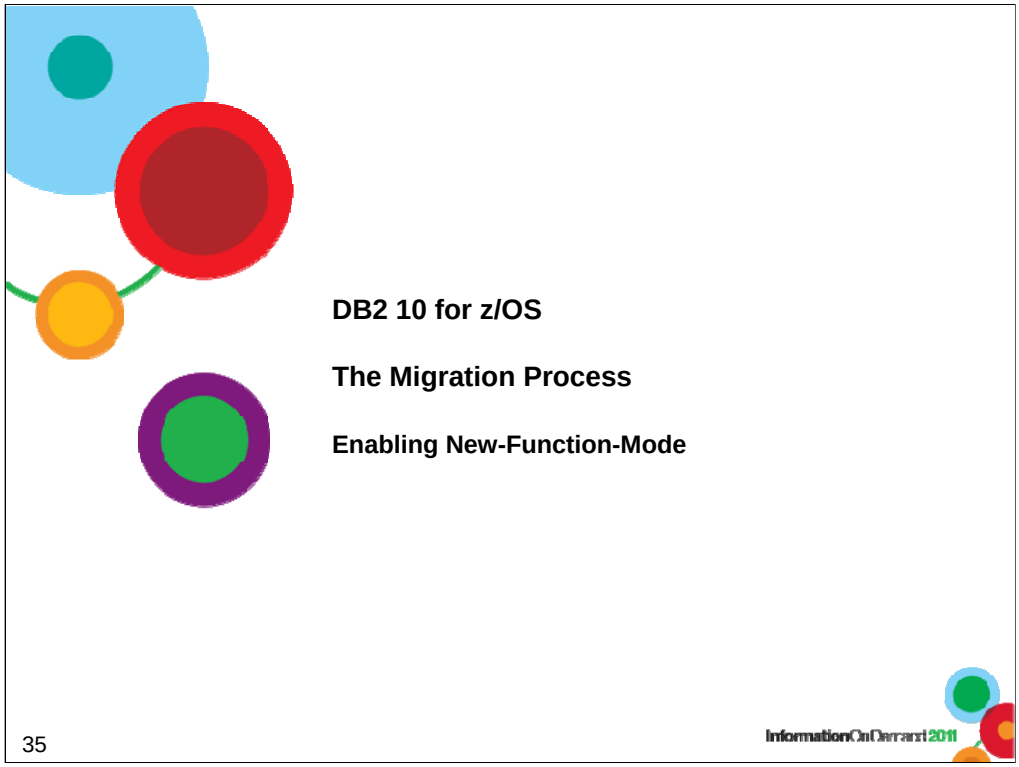
11 new aux indexes

47 new table spaces

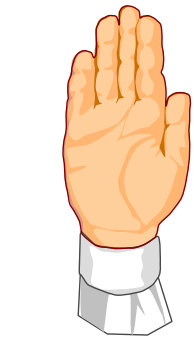
4 new tables

11 new indexes

7 old table spaces dropped


DB2 10 for z/OS
The Migration Process
Enabling New-Function-Mode
35
InformationOnDemand2011

ENFM and NFM Considerations



Attention: You cannot return to the previous release (DB2 V8 or DB2 9) once you enter ENFM.

Do NOT move to ENFM until you are certain that you will not need to return.

HOWEVER

The code base for DB2 10 ENFM and NFM is the same as for CM.

You can return to CM* from ENFM or NFM if necessary.

Moving to Enabling New Function Mode

- DSNTIJEN job:
  - CATENFM START places DB2 in Enabling New Function Mode.
  - Data sharing groups must only have DB2 10 CM members.
- ENFM will handle the **"Catalog Restructure"** changes
- Existing catalog and directory table spaces will be dropped and the tables will be moved to the new SMS-controlled table spaces.
  - DB2-defined indexes on these tables in the catalog and directory will also become SMS-controlled.
  - User-defined catalog indexes continue as before.
- New columns will be added.
- A number of new Indexes will also be created

Data sharing groups must only have DB2 10 members - The first time DSNTIJEN runs, DB2 saves the RBA or LRSN of the system log in the BSDS.

The DSNTIJEN job invokes the CATENFM and REORG (with SHRLEVEL REFERENCE) utilities in order to convert a V10 conversion mode catalog to ENFM level and complete the catalog restructure changes, which includes:

• Move the existing catalog tables to the new PBG table spaces. Each table with its own table space. These table spaces use row-level locking and all the tables are in reordered-row format.

• Convert SCTR, SYSLGRNX, SYSUTIL and SYSUTILX from EBCDIC to UNICODE.

• Merge SYSUTIL and SYSUTILX tables.

• Drop the old catalog table spaces.

Catalog and directory data sets under SMS control

During the ENFM process, some of the existing catalog and directory table spaces are dropped and the tables moved to the new SMS-controlled table spaces. The DB2-defined indexes on the tables in these catalog and directory table spaces also become SMS-controlled during the ENFM process while the user-defined catalog indexes stay the same way as before.

During ENFM, we create more objects:

8 additional new columns

11 new LOB table spaces

11 new aux tables

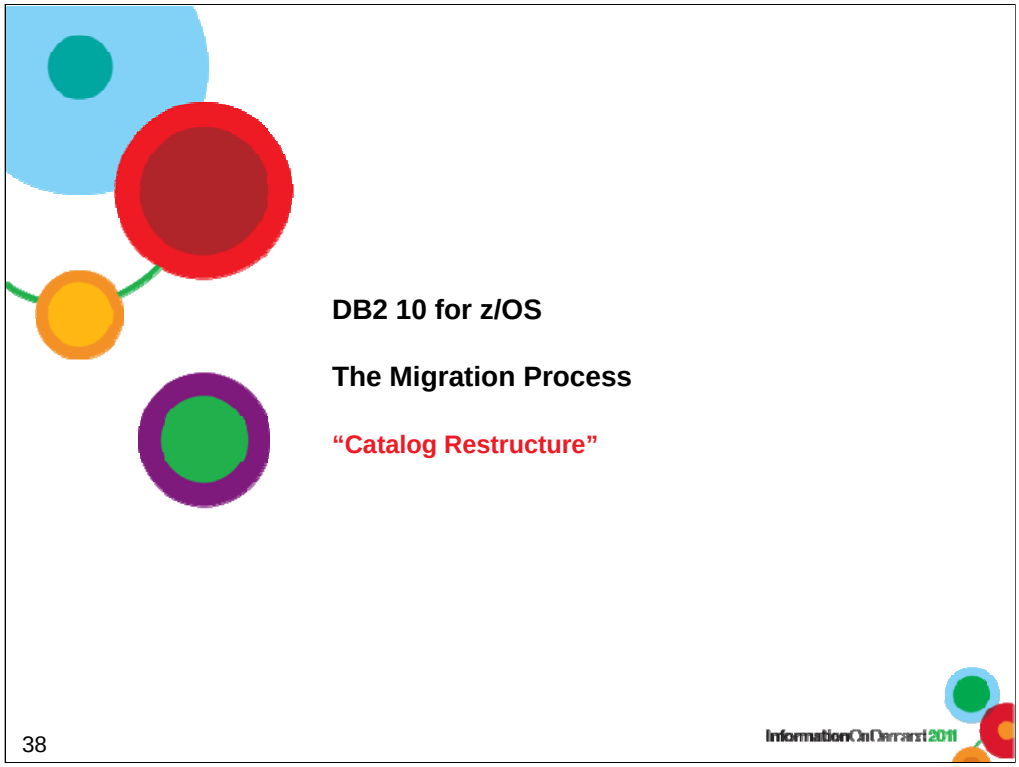
11 new aux indexes

47 new table spaces

4 new tables

11 new indexes

7 old table spaces dropped


DB2 10 for z/OS
The Migration Process
“Catalog Restructure”
38
InformationOnDemand2011

Catalog Restructure enhancement summary

- DB2 (SMS) managed catalog and directory data sets
- New CLOB/BLOB columns to the catalog
  - Merge records that store SQL statements' text
- Reduce catalog contention
  - Removal of links
  - Change to row-level locking
- Convert some catalog and directory table spaces to partition-by-growth (PBG)
  - With MAXPARTS = 1
  - New Row Format

Catalog Restructure is comprised of the following functions:

1. Reduce catalog contention by removing links and change to row-level locking
2. Make management of catalog and directory data sets easier by using z/OS Storage Management Subsystem (SMS).
3. Convert some catalog and directory table spaces to partition-by-growth (PBG), with MAXPARTS = 1
4. Add new LOB columns to the catalog (tables SYSINDEXES, SYSPACKSTMT, SYSVIEWS, SYSTRIGGERS, DBDS, SPT01) to merge records that store SQL statements' text.
6. Combine SYSUTIL and SYSUTILX into a single table SYSUITLX.
7. For documentation purpose, summarize catalog changes from other line items.

Similar to V8 and V9, DB2 V10 will also have a enabling-new-function mode process. This line item primarily focuses on the major catalog restructure changes that take place in this process. This line item will also address all the other catalog changes that take place during migration to V10 conversion mode.

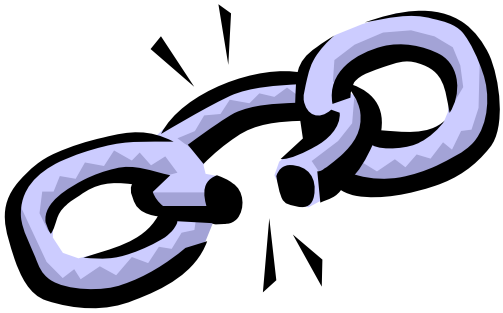
DB2 managed data sets - Benefits

- Minimize user's effort to maintain data sets.
  - No need to allocate data sets and extended pieces (A002, A003 etc…)
  - No need to allocate data sets as part of the migration (next release)
    - We still have DSNTIJIN for new installs to define data sets for the catalog and directory.
  - No need to allocate data sets for the shadow for online REORG.
  - No need to estimate the size for the data sets (had to provide space allocations in the DEFINEs)
    - DB2 will use a default for the primary and a sliding scale for secondary.
    - Minimize the chance of running out of extends before reaching the maximum data set size.
  - SMS will determine which data set goes to which volume.
  - Minimize outage due to improper data set allocations.
- New fields in installation/migration panels (CLIST)
  - SMS information (data class, storage class, management class) stored in ZPARM.

Convert all the new catalog and directory table spaces and DB2-defined indexes to SMS-controlled.

Row level locking and removing links

- SYSDBASE, SYSPLAN, SYSDBAUT, SYSVIEW, SYSGROUP and DBD01 had links
- These table spaces used page level locking because of the links.
- SPT01, SYSOBJ, and SYSPKAGE are also processed in ENFM.
- All of these table spaces will be removed and the tables within each will be moved to new PBG table spaces
  - Row level locking
  - New row format
  - Partition-by-growth
  - One table per table space
  - Referential Integrity in place of links
  - DSSIZE 64G
  - MAXPARTS 1
  - PLAN MANAGEMENT = EXTENDED



- Links removal.
- Add new foreign key to the catalog tables that originally had link to a parent table.

Row level locking and removing links - Benefits

- Row level locking minimizes timeout/deadlock for concurrent catalog access:
  - DDL
  - DML
  - Grant/revoke
  - Commands (DISPLAY, BIND/REBIND/FREE etc)
  - Utilities (e.g. RUNSTATS)
- Less or no SQL timeout when running BIND at the same time therefore increased availability for your applications.
- Reduced catalog contention:
  - Multiple binds in parallel
  - More concurrent Binding and running of DDL

Improve catalog concurrency has benefits for all

- Application Developer/ Architect binding and running applications
- Production DBA responsible for performance and running RUNSTATS.
- Development DBA responsible for database design and running the DDL.
- System Programmer responsible for system performance and running RUNSTATS.

With this enhancement contention on the catalog should be reduced allowing multiple binds to run without deadlocks or timeouts. We also see more concurrency between bind, DDL and utilities.

Eliminate 64Gb limit on Catalog & Directory – e.g. SPT01

- **V10 will relieve SPT01 space problems**
- SPT01 can grow beyond 64G, when system in DB2 10 NFM.
- Hence Application Developer creates packages using:
  - Creates Packages using BIND PACKAGE.
  - Creates Packages using CREATE TRIGGER.
  - Creates Packages using SQL procedures.
  - Binds a package, using plan stability (now the default).
- DB2 extends SPT01 beyond 64G when needed using BLOBs.
  - Primary BLOB inlined by default to preserve compression
    - PM27073 and PM27811

Binds and rebinds don't fail for lack of SPT01 space.

Aspects of a successful solution:

Binds and rebinds don't fail for lack of SPT01 space.

V10 will relieve SPT01 space problems

SPT01 can grow beyond 64G, when system in DB2 10 NFM. Hence:

Application Developer creates packages using BIND PACKAGE.

• Creates packages using CREATE TRIGGER.

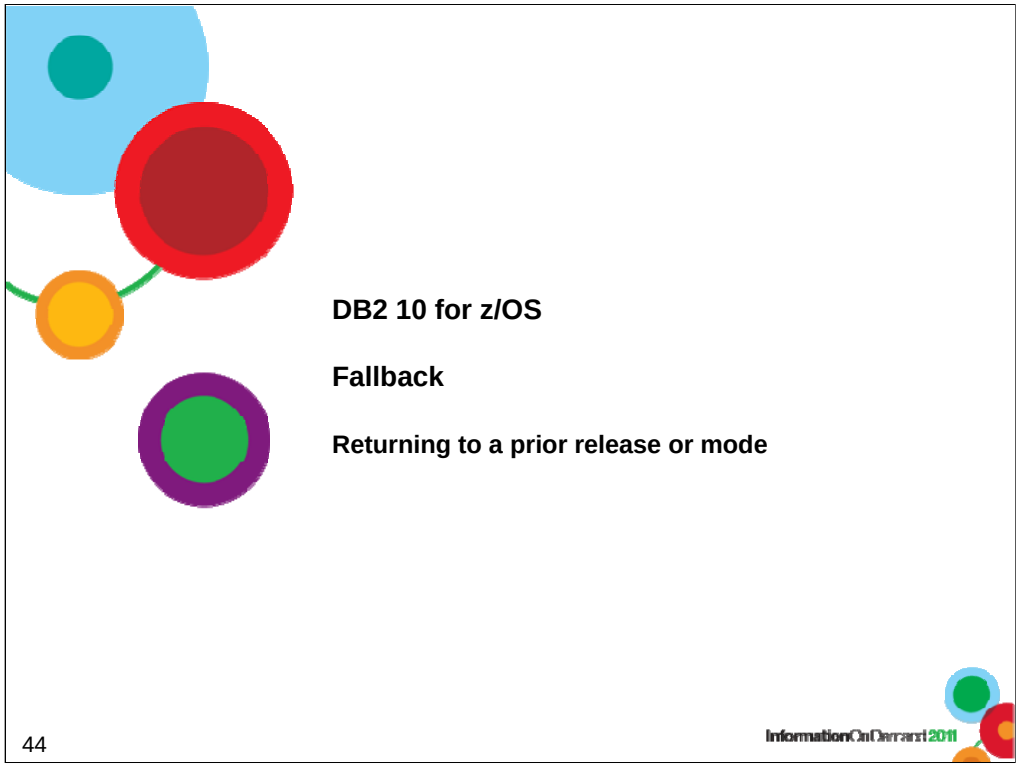
• Creates packages using SQL procedures.

• Binds a package, using plan stability which is now the default

DB2 extends SPT01 beyond 64G when needed.

When the SPT01 table space approaches 64G:

- Turn on compression for SPT01
- Reduce the size of the inlined LOB


DB2 10 for z/OS
Fallback
Returning to a prior release or mode
44
InformationOnDemand2011

Returning to a previous release from DB2 10

- **Fallback is only supported from DB2 10 CM.**
- Migrate to DB2 10 CM9 then you can ONLY fallback to DB2 9 NFM
- Migrate to DB2 10 CM8 then you can ONLY fallback to DB2 8 NFM
- Fallback SPE (PK56922) **must** be applied beforehand.
- Packages bound (or rebound) in DB2 10 CM are automatically rebound in DB2 V8 NFM or DB2 9 NFM.
- DBRMs created by the V10 Precompiler cannot be bound on V8 or V9
- Objects with functional dependencies are indicated by an 'O' in the IBMREQD catalog columns.
  - Frozen on return to DB2 V8 or DB2 9.

Fallback to DB2 V8 NFM or DB2 9 NFM is only supported from DB2 10 CM.

Falling back is the process of returning to a supported previous release of DB2, after a successful migration of the catalog and directory to DB2 10 for z/OS Conversion Mode (CM). Supported releases are DB2 V8 for z/OS from V10 CM8, and DB2 9 for z/OS from V10 CM9.

Prerequisite changes to the fallback release must be installed before migrating to DB2 10 for z/OS; that is, to ensure fallback to the previous release, it is necessary to apply the required maintenance and the fallback SPE to the fallback release (DB2 V8 or DB2 9) before migrating to DB2 for z/OS Version 10. (The changes to the fallback release are necessary to tolerate the migrated catalog, directory, and log records thus allowing fallback after use of the new facilities).

The new facilities of this release require extensions to DB2. The extensions are upward compatible so that those objects which are not dependent on the new facilities are acceptable to the fallback release. Objects that are dependent on the new facilities are not acceptable to the fallback release. To distinguish the two classes of objects, DB2 will record the use of its new facilities. The recording is in the form of a release identifier (also known as the release dependency indicator) so that the scheme can also be used in subsequent releases. The new facilities that force a release dependency are only available in NFM.

There are two types of release dependencies: functional dependencies and design dependencies. Functional dependencies can apply to any type of object including plans and packages and are the result of the use of new external function. Design dependencies apply to packages and are the result of SQL optimization enhancements. For example packages that use sysplex parallel query processing will have design release dependencies but no functional release dependencies.

Some objects in this catalog that have been affected by function in this release might become frozen objects after fallback. Frozen objects are unavailable, and they are marked with a release dependency indicator. To record the release dependency indicators, every catalog table has a CHAR(1) column named IBMREQD. In later releases, the use of the IBMREQD column was extended to record functional release dependencies. The release dependency indicators are:

Y denotes a built-in row of the catalog as in Version 1 Release 1 or 2.

N denotes a row that is not built-in and which describes an object that is not dependent on any new external function of the release.

L denotes a row that is not built-in and which describes an object that is dependent on new external function(s) of DB2 for z/OS Version 8.

M denotes a row that is not built-in and which describes an object that is dependent on new external function(s) of DB2 9 for z/OS.

O denotes a row that is not built-in and which describes an object that is dependent on new external function(s) of DB2 10 for z/OS.

After fallback, objects with functional release dependencies are unavailable until a re-migration occurs. After fallback, packages with design

Fallback Considerations

- **Buffer pool enhancements**
  - Fall back process unchanged
  - Any BPs changed to PGSTEAL=NONE will revert to their previous values
    - i.e. LRU if PGSTEAL was never altered
    - Otherwise the value from the last alter (under either release)
  - On remigration, any BP which was set to PGSTEAL=NONE before fallback will revert to PGSTEAL=NONE.
- **Autonomic diagnosis & tuning for query performance issues**
  - Fall back process unchanged.
  - No special procedure for the explain tables.
    - These tables in V10 explain schema and UNICODE would simply work with the fallback subsystem as it was before after applying the SPE fallback APAR and completing the preparatory steps.

Buffer pool enhancements - Fallback

The fall back process is unchanged from the prior release. Upon fallback, the PGSTEAL attributes of any buffer pools which were changed to PGSTEAL=NONE will revert to their previous values, i.e. LRU if PGSTEAL was never altered to a different value, otherwise the value from the last alter (under either release) before the alter to PGSTEAL=NONE. Upon remigration, any buffer pool which was set to PGSTEAL=NONE before fallback will revert to PGSTEAL=NONE.

Autonomic diagnosis & tuning for query performance issues - Fallback

The fall back process is unchanged from the prior release. In this process, there is no special procedure to be done for the explain tables. These tables in V10 explain schema and UNICODE would simply work with the fallback subsystem as it was before after applying the SPE fallback APAR and completing the preparatory steps.

Acknowledgements and Disclaimers:

Availability. References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates.

The workshops, sessions and materials have been prepared by IBM or the session speakers and reflect their own views. They are provided for informational purposes only, and are neither intended to, nor shall have the effect of being, legal or other guidance or advice to any participant. While efforts were made to verify the completeness and accuracy of the information contained in this presentation, it is provided AS-IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this presentation or any other materials. Nothing contained in this presentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

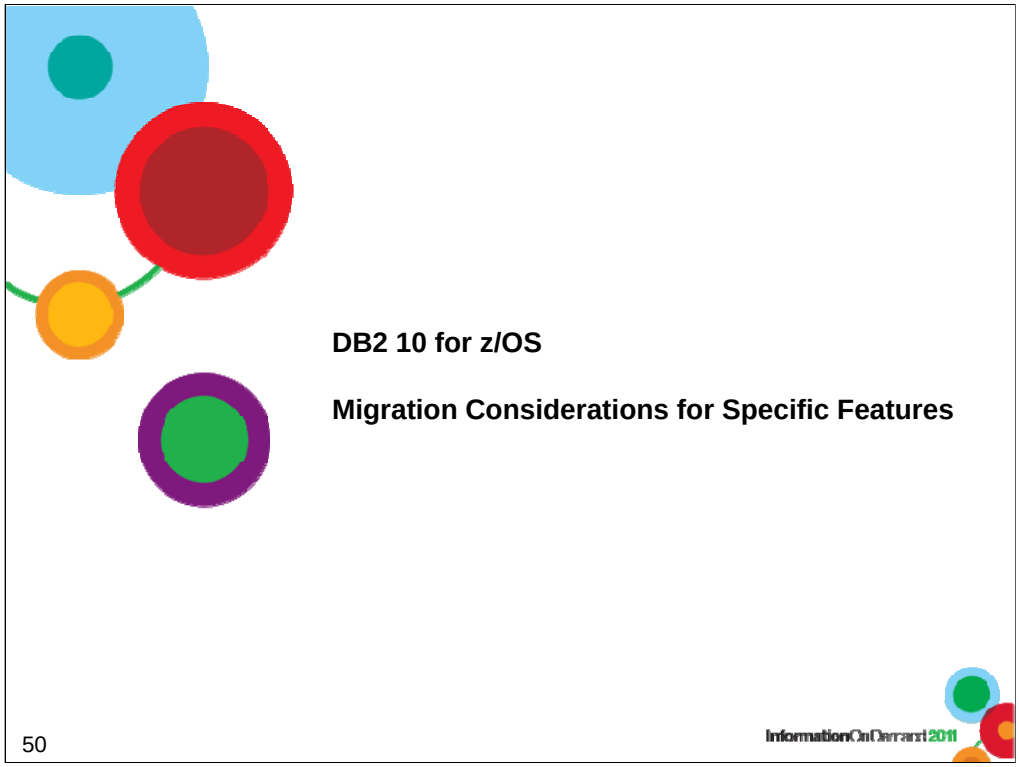
Communities

- **On-line communities, User Groups, Technical Forums, Blogs, Social networks, and more**
  - Find the community that interests you …
    - **Information Management** **ibm.com**/software/data/community
    - **Business Analytics** **ibm.com**/software/analytics/community
    - **Enterprise Content Management** **ibm.com**/software/data/content-management/usernet.html

- **IBM Champions**
  - Recognizing individuals who have made the most outstanding contributions to Information Management, Business Analytics, and Enterprise Content Management communities
    - **ibm.com**/champion

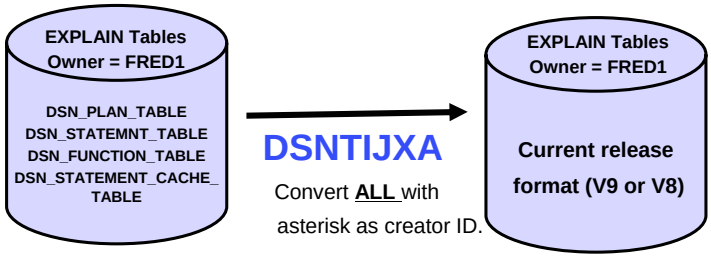
Thank You!
Your Feedback is Important to Us

- Access your personal session survey list and complete via SmartSite
  - Your smart phone or web browser at: iodsmartsite.com
  - Any SmartSite kiosk onsite
  - Each completed session survey increases your chance to win an Apple iPod Touch with daily drawing sponsored by Alliance Tech


DB2 10 for z/OS
Migration Considerations for Specific Features
50
InformationOnDemand2011

Migrate EXPLAIN tables to V10 schema encoded in UNICODE - Prepare

DB2 10 EXPLAIN tables must be in latest schema format and in UNICODE.



- Important - migrating V8 to V10, you will not be able to convert DSN_STATEMENT_CACHE_TABLE to V10 format until DB2 enters V10 new-function mode. (BIGINT not supported in V8).
  - However, DB2 V10 will tolerate DSN_STATEMENT_CACHE_TABLE in V8 format.
  - Once DB2 is in V10 new-function mode, rerun job DSNTIJXA to convert DSN_STATEMENT_CACHE_TABLE to V10 format.

Part of the Autonomic diagnosis & tuning for query performance issues enhancements

Migrating EXPLAIN tables

This process migrates the explain tables to V10 schema encoded in UNICODE. The explain tables migration process described here only covers migration from V8 NFM or V9 NFM to V10 CM, then to V10 NFM. If customers have a DB2 subsystem in V7 or earlier, it is recommended to bring the system up to at least V8 NFM level by following the recommended migration procedure.

The migration process is divided into several phases:

1. Preparatory phase

a. DB2 is assumed at V9 NFM level. (The steps for migrating from V8 NFM is similar, see step 5 below).

b. Customers are recommended to migrate the explain tables to the current release of explain table schema format, that is either V9 or V8, as early as possible to be ready for release to release migration. Customers can skip this step if they have migrated to the latest explain schema already.

c. A sample, customizable job to migrate the explain tables is provided to ease this process. Use job DSNTIJXA to bring all explain tables belonging to a specified creator ID into current release format. To convert all non-compliant explain tables regardless of the creator, specify an asterisk as the creator ID. Note: If you are migrating to DB2 V10 from DB2 V8, you will not be able to convert DSN_STATEMENT_CACHE_TABLE to V10 format until DB2 enters V10 new-function mode. Until that time, DB2 V10 will tolerate DSN_STATEMENT_CACHE_TABLE in V8 format. Once DB2 is in V10 new-function mode, rerun job DSNTIJXA to convert DSN_STATEMENT_CACHE_TABLE to V10 format.

Enhanced Monitoring support – Migration impact on SYSPACKSTMT

- V10 CM, a new STMT_ID column created (but NOT materialized) as INT NOT NULL.
- During ENFM, STMT_ID column altered to BIGINT and a new system sequence object SYSIBM.DSNSEQ_PKS_STMT_ID also created as BIGINT.
- ENFM REORG phase, populates STMT_ID column for any existing rows in SYSPACKSTMT table.
- After ENFM, SYSPACKSTMT will be out of sync with SPT01 until REBIND of the existing package is done.
- During BIND phase, the STMT_ID column is populated via the SYSIBM.DSNSEQ_PKS_STMT_ID sequence object for the new package.
- Rebind of any existing pre-v10 package for static SQL statements is required once the ENFM process is completed successfully.
  - SYSPACKSTMT then in sync with SPT01
  - STMT_ID column contains a unique value for each row in the SYSIBM.SYSPACKSTMT table.
- When the statement identifier shows in the messages and traces for static statements, the user then can correlate the identifier to the STMT_ID column in the SYSIBM.SYSPACKSTMT table to identify the particular SQL statement in question.

The migration impact on the SYSIBM.SYSPACKSTMT table

A new STMT_ID column is created in Conversion Mode(CM) as INT NOT NULL. The column is not materialized in CM. The default value is zero for the column.

During ENFM processing, the STMT_ID column is then altered as BIGINT data type. A new sequence object SYSIBM.DSNSEQ_PKS_STMT_ID is created as BIGINT. This system sequence object is created to be used for the STMT_ID column, any attempted reference on this system-generated sequence object is not allowed and resulted in with SQLCODE -20142.

In ENFM REORG phase, the STMT_ID column is populated for any existing rows in the SYSIBM.SYSPACKSTMT table. After ENFM process is completed successfully, SYSIBM.SYSPACKSTMT catalog table will be out of sync with DB2 directory residing in SPT01 until REBIND of the existing package is done.

During BIND phase after ENFM process, the STMT_ID column is then populated via the SYSIBM.DSNSEQ_PKS_STMT_ID sequence object for the new package, the sequence value starts at n+1 and increments by 1 where n is the number of the existing rows in the SYSIBM.SYSPACKSTMT table. After a FREE package, n is actually the current contents of the sequence.

Rebind of any existing pre-V10 package for static SQL statements is required once the ENFM process is completed successfully for any user who wants to utilize the enhanced messages and traces that provide the statement identifier information. Once the package is rebound or bound, SYSIBM.SYSPACKSTMT table is in sync with DB2 directory residing in SPT01 and the STMT_ID column contains a unique value for each row in the SYSIBM.SYSPACKSTMT table. When the statement identifier shows in the messages and traces for static statements, the user then can correlate the identifier to the STMT_ID column in the SYSIBM.SYSPACKSTMT table to identify the particular SQL statement in question.

Migration impact on SYSIBM.DSN_PROFILE_TABLE_IX2_ALL index

- Index on **SYSIBM.DSN_PROFILE_TABLE**
  - DB2 V10 adds columns, ROLE , PRDID, and GROUP_MEMBER
- No migration considerations for V8 to V10.
- After migrating update format of **SYSIBM.DSN_PROFILE_TABLE**:
  - Use SPUFI to alter/add the new columns to the table and index, so that it includes the new keys in the required sequence
  - Job DSNTIJOM.
  - Job DSNTIJXA to run REXX exec DSNTXTA, specifying schema name SYSIBM. DSNTXTA.
- To utilize the new V10 system profile monitoring functions related to connections and threads, the new index must be defined.
  - If not, a DSNT745 for START PROFILE command issued
  - A user who does not use the new V10 system profile monitoring functions can use profile monitoring properly with the old V9 index. However, any usage of the new V10 system profile monitoring functions in the profile table requires the new DSN_PROFILE_TABLE_IX2_ALL definition.

Profile tables contain information about how DB2 executes or monitors a group of statements. **SYSIBM.DSN_PROFILE_TABLE** is used to specify which SQL statements are monitored or tuned by each monitor profile. Each row defines a particular monitor profile.

DB2 V10 adds three columns, ROLE , PRDID, and GROUP_MEMBER, to the index SYSIBM.

DSN_PROFILE_TABLE_IX2_ALL. There is no migration consideration for customers who migrate from V8 to V10 because these objects are not created in V8. After migrating from V9 to V10, customers who have an existing SYSIBM.DSN_PROFILE_TABLE can use any of the following techniques to bring it into current release format:

• Use SPUFI or a similar tool to process SQL statements to alter/add the new columns to SYSIBM.

DSN_PROFILE_TABLE and to drop and recreate SYSIBM.

DSN_PROFILE_TABLE_IX2_ALL so that it includes the new keys in the required sequence

• Use job DSNTIJOM to alter/add the new columns to SYSIBM.DSN_PROFILE_TABLE and drop and recreate SYSIBM.DSN_PROFILE_TABLE_IX2_ALL so that it includes the new keys in the required sequence.

• Use job DSNTIJXA to run REXX exec DSNTXTA, specifying schema name SYSIBM. DSNTXTA will then alter/add the new columns to SYSIBM.DSN_PROFILE_TABLE and drop and recreate SYSIBM.DSN_PROFILE_TABLE_IX2_ALL so that it includes the new keys in the required sequence.

When the user wants to utilize the new V10 system profile monitoring functions related to connections and threads, the new index definition must be used. A DSNT745 code will be issued for START PROFILE command if the new index definition is not created along with the new DSN_PROFILE_TABLE definition with columns ROLE and PRDID. A user who does not use the new V10 system profile monitoring functions can use profile monitoring properly with the old V9 index. However, any usage of the new V10 system profile monitoring functions in the profile table requires the new DSN_PROFILE_TABLE_IX2_ALL definition.

XML Schema Validation in the engine

- The UDF version of DSN_XMLVALIDATE is deprecated in V10, and will be removed in the release that follows.
- To use the new built in scalar function DSN_XMLVALIDATE, an SQL package rebind is needed.
  - Not rebinding the package, means the old UDF DSN_XMLVALIDATE will be invoked.
- After rebinding the package,
  - if DSN_XMLVALIDATE is unqualified or qualified with SYSIBM in the call, the new built in scalar function will be invoked;
  - if DSN_XMLVALIDATE is qualified with SYSFUN in the call, the old UDF DSN_XMLVALIDATE will be invoked.

XML schema validation in the engine - Migration

The user defined function version of DSN_XMLVALIDATE is deprecated in V10, and will deplete in the release that follows. Deprecation in V10 means we still document it and allow users to use it. To use the new built in scalar function DSN_XMLVALIDATE, an SQL package rebind is needed. If a user does not rebind the package, the old UDF DSN_XMLVALIDATE will be invoked.

After rebinding the package, if DSN_XMLVALIDATE is unqualified or qualified with SYSIBM in the call, the new built in scalar function will be invoked; if DSN_XMLVALIDATE is qualified with SYSFUN in the call, the old UDF DSN_XMLVALIDATE will be invoked.

Dynamic Statement Cache enhancements

- New column, LITERAL_REPL, added to user table DSN_STATEMENT_CACHE_TABLE
  - Install / Migration process will handle adding of column
  - CHAR(1) NOT NULL WITH DEFAULT

'R'	Literals have been replaced with '&'
'D'	Literals have been replaced with '&' and may have an identical value for STMT_TEXT for another cached statement, but literal reusability matching criteria failed.
' '	CONCENTRATE STATEMENTS has a value of OFF (default value).

Dynamic Statement Cache enhancements - Migration

A new column, **LITERAL_REPL**, is added to the **user table DSN_STATEMENT_CACHE_TABLE**.

The DB2 10 Install / Migration process is modified to handle adding the new column to table DSN_STATEMENT_CACHE_TABLE.

The column **LITERAL_REPL** is CHAR(1) NOT NULL WITH DEFAULT.

UTS support for Member Cluster

- Pre V10, values "K" or " I " in the TYPE column of SYSTABLESPACE indicated a MEMBER CLUSTER table space structure.
  - "K" = >64Gb and "I" = <64Gb
- In V10 new column MEMBER_CLUSTER added to SYSTABLESPACE
  - Populated during ENFM process for a MEMBER CLUSTER table space.
- Original values of 'I' or 'K' replaced with blank or L respectively.
  - New column SYSTABLESPACE MEMBER_CLUSTER will have value of 'Y'.

SYSTABLESPACE.TYPE = I

Mode	Value of SYSTABLESPACE.TYPE	Value of SYSTABLESPACE. MEMBER_CLUSTER column
Before ENFM	I	Blank
After ENFM	Blank	Y

SYSTABLESPACE.TYPE = K

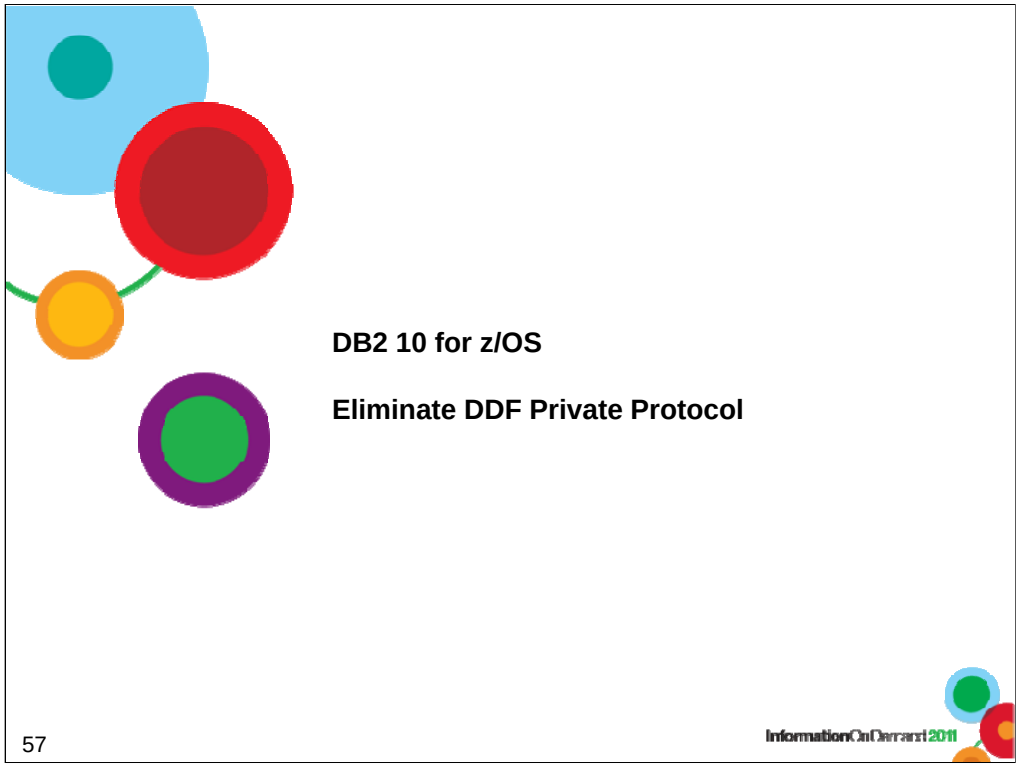
Mode	Value of SYSTABLESPACE.TYPE	Value of SYSTABLESPACE. MEMBER_CLUSTER column
Before ENFM	K	Blank
After ENFM	L	Y

Universal Table Space support for member Cluster - Migration

Prior to this line item, value "K" or " I " in the TYPE column of the SYSTABLESPACE catalog table indicates the table space has MEMBER CLUSTER structure.

A new column MEMBER_CLUSTER on the SYSTABLESPACE catalog table is added, and it will be populated during the ENFM of migration process for existing MEMBER CLUSTER table space.

The original value of 'I' or 'K' SYSTABLESPACE TYPE column will be replaced with blank or L respectively, and the new column SYSTABLESPACE MEMBER_CLUSTER will have value of 'Y'.


DB2 10 for z/OS
Eliminate DDF Private Protocol
57
InformationOnDemand2011

V9 Deliverable

- DB2 9 for z/OS delivered the following functionality
  - Warning when PRIVATE protocol used at BIND time
  - Prevent PRIVATE being the default option
  - IFCID records to determine applications using PRIVATE
  - Analysis Tool to identify and convert plans and packages
    - Additionally identify Aliases required at remote sites
- DB2 10 will no longer support **ANY** Private Protocol
  - Customers should convert to DRDA prior to V10
  - Enhancements to V8 and V9 DRDA support to aid conversion

Enhancements to V8 & V9

- The following is delivered to facilitate conversion
  - Ability to Enable or Disable Private protocol (PK92339)
    - Via ZPARM PRIVATE_PROTOCOL
    - Enables testing to ensure private is eliminated
  - Allow Private protocol to access V8/V9 data sharing members
    - Until all members are at V10 CM
  - DRDA alias resolution processing to be enabled or disabled (PK64045)
  - Provide Enhanced Tools to identify plans and packages requiring conversion (PK64045)
    - DSN1PPTP analyses IFCID157 trace records for input into DSNTP2DP
    - DSNTP2DP reports on applications using private protocol
  - Program to scan SYSSTMT and SYSPACKSTMT for invalid syntax (PK64045) - DSNTPPCK

Provide a DSNTIJTC to switch a PLAN and/or PACKAGE bound with DBPROTOCOL(PRIVATE) to DBPROTOCOL(DRDA) when it is determined that a PLAN or PACKAGE

does not have any remote location dependent statements or dynamic SQL statements

ZPARM MACRO DSN6FAC

DSN1PPTP details

-START TRACE(PERFM) CLASS(30) IFCID(157) DEST(SMF)

will go through the SMF data file looking for IFCID 157 trace entries amongst the SMF 102 records and then put out the information contained in those trace entries to a JCL DD data set, IFCID157. The format of the IFCID157 file will be as follows:

LOAD	PLANNAME	CORRNAME	CORRNMBR	CONNECTTYPE	SECTN	R/W	CALLTYPE	SERVING_LOCATION PROGRAM/PACKAGE_
YES	LI883A	LI883J5	'BLANK'	DB2 CAF	1	W	AUXCALL	STLEC4B LI883A
YES	LI883B	LI883J5	'BLANK'	DB2 CAF	1	W	AUXCALL	STLEC4B LI883B
YES	LI883C	LI883J5	'BLANK'	DB2 CAF	1	W	AUXCALL	STLEC4B LI883C
YES	LI883D	LI883J5	'BLANK'	DB2 CAF	1	R	OPENCALL	STLEC4B LI883D

The actual record length will be about 133 bytes.

The LOAD column in the IFCID157 file is being added so that the data from the IFCID157 trace can be then loaded into a DB2 table which can then be used by the DSNTP2DP via the PPINFOTABLE parameter specification to fill in any target locations for plans and packages.

DSNTP2DP will scan catalog and output from DSN1PPTP creates

Creates 2 part alias names for target systems for 3 part alias names found locally

Creates plan/package commands to convert protocol

Input parm PPINFOTABLE – table containing IFCID 157 trace info

Tools only needed for V8/V9 as PRIVATE cannot be present in V10 systems

V10 Enhancements

- Enhancements to V10 are
  - Remove PRIVATE PROTOCOL from DB2
  - Attempt to load/execute object with DBPROTOCOL column = 'P' will be tolerated, but any reference to non-local data will fail with SQLCODE -904, reason code 00E3001E
  - REBIND will always use DBPROTOCOL(DRDA) regardless of object's current DBPROTOCOL value
  - BIND PACKAGE COPY will create DBPROTOCOL(DRDA) object regardless of source object of copy
  - DRDA alias resolution will always be performed
    - SQL statement text sent to remote system will contain object names substituted for 3-part name alias references
      - Aliases will not be needed at remote locations
    - Occurs during dynamic PREPARE and EXECUTE IMMEDIATE and remote package bind

Currently, when processing an SQL statement from an application under control of a package or plan bound with DBPROTOCOL(DRDA), the SQL statement text is sent to a remote location as it was received from the application. Thus, if the SQL statement referenced an alias, and that alias resolved to a remote 3-part name, then the statement with the alias reference would be sent unchanged on to the remote location (identified by the alias) for processing. This is not the behaviour of private protocol, in that the alias reference in the SQL statement is resolved to the remote object name prior to the statement being sent to the remote location. Also, the private protocol behaviour occurs whether the statement is being processed via embedded dynamic SQL processing or embedded static SQL processing. By not resolving alias references into the SQL statement text during DRDA protocol processing, one might need an alias at the remote location to resolve the object reference. Also, the following alias definition scenario will not currently work with DRDA protocol:

• create an alias named ALIAS1 in location DB2A to point to table USER.TABLE at DB2B

• create an alias named ALIAS1 in location DB2B to point to table USER.TABLE at DB2A

• with private protocol, USER.TABLE is accessed at DB2B when specifying ALIAS1 at DB2A and USER.TABLE is accessed at DB2A when specifying ALIAS1 at DB2B

• with DRDA protocol, the ALIAS1 access on either system would result in an error because the alias reference would lead to a continuous cross-system loopback until the nesting of levels reached 16 levels

In V10, aliases will be resolved with DRDA protocol during:

• a PREPARE of a dynamic SQL statement (if DEFER(PREPARE) is also specified, the SQL statement text will have the alias reference resolved but the statement will not be sent to the remote location for processing until a request to execute the statement is made by the application)

• an EXECUTE IMMEDIATE of a dynamic SQL statement

• a remote package bind of a static SQL statement

Migration Steps

- Activate trace while running Private protocol applications
  - Identify embedded dynamic SQL and remote sites referenced
- Enable Alias Resolution
  - To properly convert applications
- Build Private Protocol Information Table
  - Output from DSN1PPTP (DSNTIJPT installation job)
- Run Protocol REXX Catalog Analysis Program (DSNTP2PD)
  - Provides commands for plan and package management
- Action Plan/Package Management commands
- Test the application
  - Check for errors from DRDA conversion
  - SQLCODE -805 suggests embedded dynamic SQL
    - Fix using BIND PACKAGE COPY of indicated package

To properly convert any application that is currently using private protocol, alias resolution processing must be enabled for DRDA protocol. This is accomplished by either changing and rebuilding an existing DB2 subsystem parameters module or creating a new subsystem parameters module which includes setting the DRDA_RESOLVE_ALIAS parameter of the DSN6SPRM ZPARM macro to YES. This new or changed subsystem parameters module can then be loaded or reloaded via the -SET SYSPARM DB2 command.

Eliminate DDF Private Protocol – Coexistence support

- We will see (V8 + V10) and / or (V9 + V10) scenarios
  - Some members will still be able to support Private Protocol inbound access.
- Enabling an earlier release of DB2 to continue to use private protocol to access the data sharing group (in coexistence mode)
  - Implement member-specific routing to V10 coexistence data sharing group (i.e. LULIST)
  - DB2 requester will attempt connect to LUNAMEs in LULIST until success (at least one member of coexistence group must be either V8 or V9)
  - If more than one member of coexistence group is V8 or V9, then requester will balance requests based on DB2 SNA Sysplex TPN exchange

Eliminate DDF PRIVATE protocol - Coexistence

Coexistence of multiple releases of DB2 is unchanged from the prior release. However, as a V8 or V9 data sharing group undergoes a gradual migration to V10CM, some members will still remain able to support private protocol inbound access while others will not. To enable an earlier release of DB2 to continue to use private protocol to access the data sharing group while some members still permit inbound private protocol and also make sure that the requester only accesses those members which still allow inbound private protocol, the following changes will be made as part of this line item:

• Code will be added in V10 which cause a DB2 to register its version level with the WLM sysplex routing services interface.

• Code will be added in V8 and V9 via the V10 toleration SPE and V10 to have the DB2 utilize the version level information retrieved from WLM sysplex routing services and return that information to any requester via the reply to the DB2 sysplex TP (VTAM Transaction Program) request

• Code will be added to the V8 and V9 requesters that will cause it to issue a DB2 sysplex TP request to the LUNAME(s) of the location to ascertain the levels of the server systems if it had previously received a private protocol VTAM communications connection request failure with VTAM sense code X'10086021' and based on the reply from the DB2 sysplex TP request, the following will occur:

- if at least one of the server systems is not V10 or the server list does not contain any version level indicators (no members are running V10), then the requester will then route the private protocol request to a remaining non-V10 system
- if all server systems are V10, then the requester will return an SQLCODE -904 with reason code X'00D31026', resource type X'1001' (System Conversation), and resource information being of the format LU.MODE.RTNCD.FDBK2.RCPRI.RCSEC.SENSE where the LU will be the LUNAME to which a conversation was being established, MODE will be the VTAM logmode that was used in the connection attempt, and RTNCD.FDBK2.RCPRI.RCSEC.SENSE will 00.0B.0004.0008.10086021