



SWG BetaWorks

DB2 9

Technical Education Series

“Migration”

The BetaWorks logo, featuring the word "BetaWorks" in a stylized font. The "B" is black, "e" is black, "t" is black, "a" is black, "W" is purple, "o" is a blue gear, "r" is yellow, "k" is red, and "s" is green.

BetaWorks

Important Disclaimer

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY.

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.

IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE.

IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION.

NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, OR SHALL HAVE THE EFFECT OF:

- CREATING ANY WARRANTY OR REPRESENTATION FROM IBM (OR ITS AFFILIATES OR ITS OR THEIR SUPPLIERS AND/OR LICENSORS); OR
- ALTERING THE TERMS AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENT GOVERNING THE USE OF IBM SOFTWARE.

Agenda

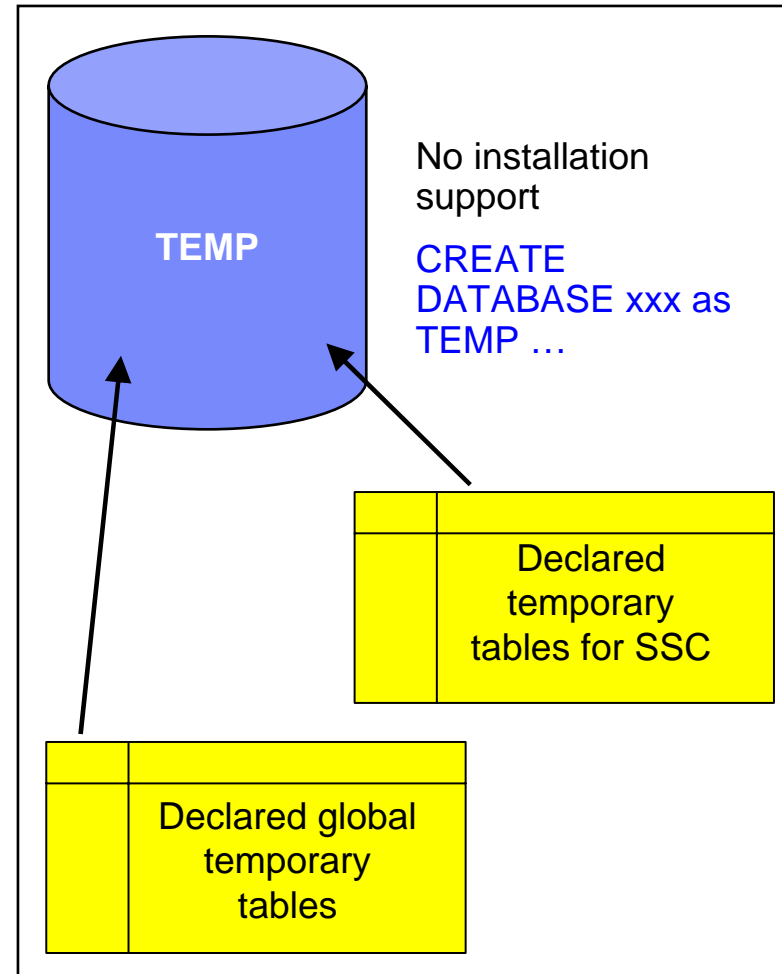
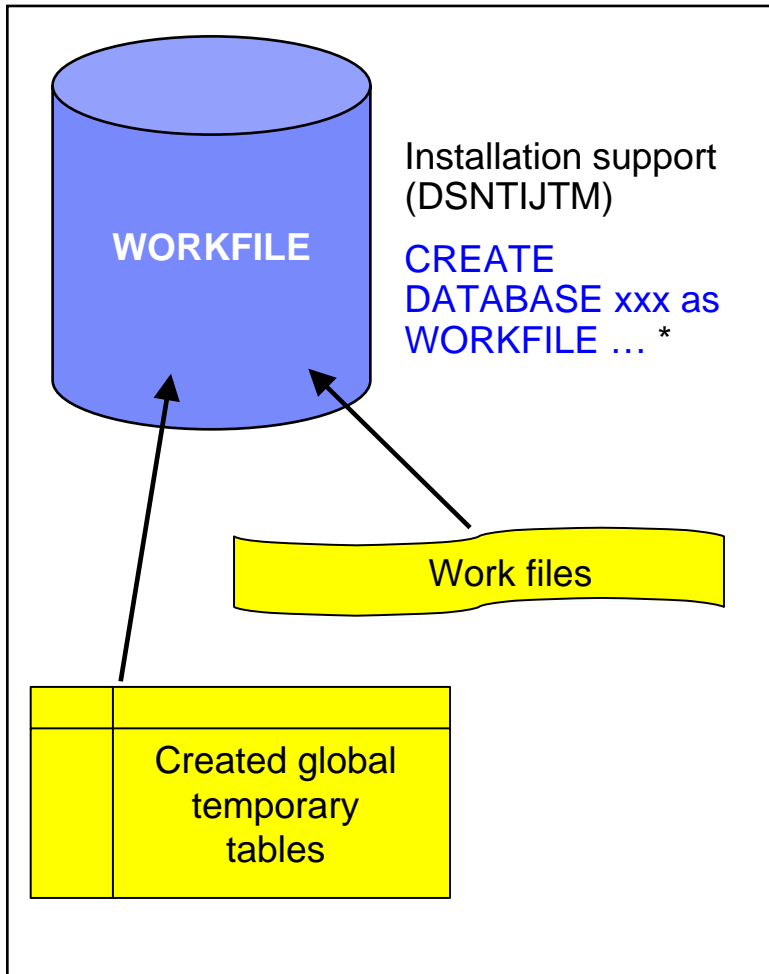
- **Converge Temp Space**
- **Migration, Coexistence and Fallback**
 - Migration Prerequisites – a Quick Glance
 - The Migration Process
 - Migration Overview
 - New Release Migration
 - Conversion to New Function Mode
 - Fallback
 - Installation and Migration Process Changes
 - New Features and DSNZPARM-related changes
 - Process Enhancements
 - Sample Applications



SWG BetaWorks

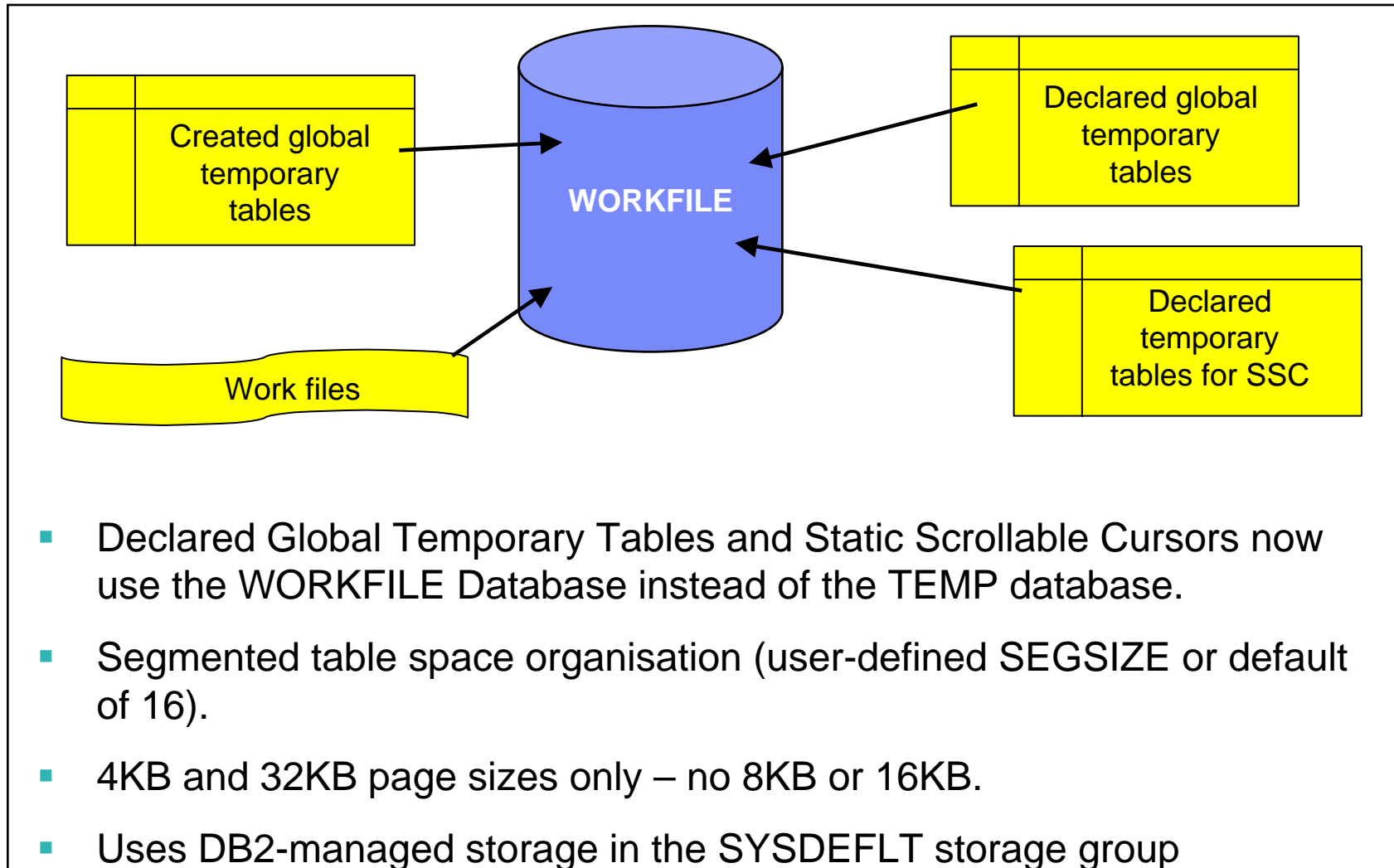
Converge Temporary Space

Temporary Space – The Current Picture



* Only in a data sharing environment – in non-data sharing syntax is **CREATE DATABASE DSNDDB07**







Temporary Space – The New Picture



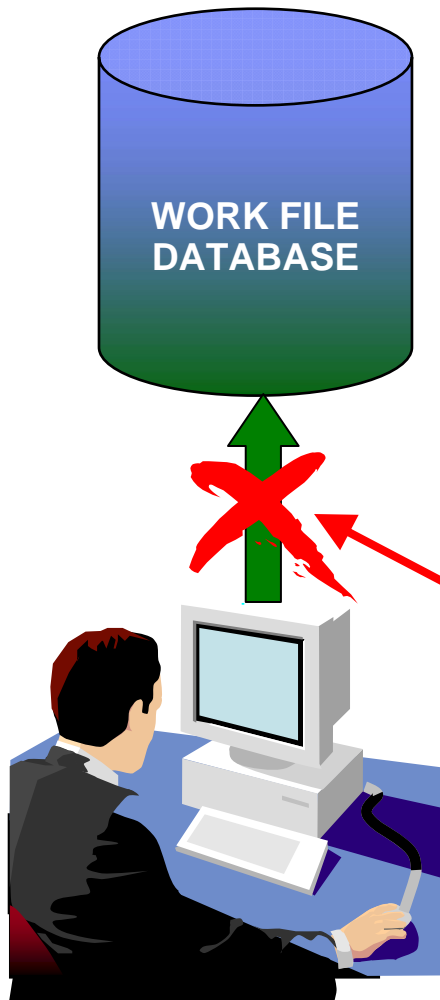
- Declared Global Temporary Tables and Static Scrollable Cursors now use the WORKFILE Database instead of the TEMP database.
- Segmented table space organisation (user-defined SEGSIZE or default of 16).
- 4KB and 32KB page sizes only – no 8KB or 16KB.
- Uses DB2-managed storage in the SYSDEFLT storage group

Converge Temporary Space – The Details

■ Changes to **CREATE/ALTER DATABASE** and **CREATE/ALTER TABLESPACE**:

- CREATE DATABASE ... AS TEMP 
- CREATE TABLESPACE in an existing TEMP database. 
- DROP table space from an existing TEMP database 
- DROP DATABASE for an existing TEMP database 
- Specify INDEXBP on CREATE DATABASE ... AS WORKFILE 
- Specify SEGSIZE for WORKFILE database table spaces 

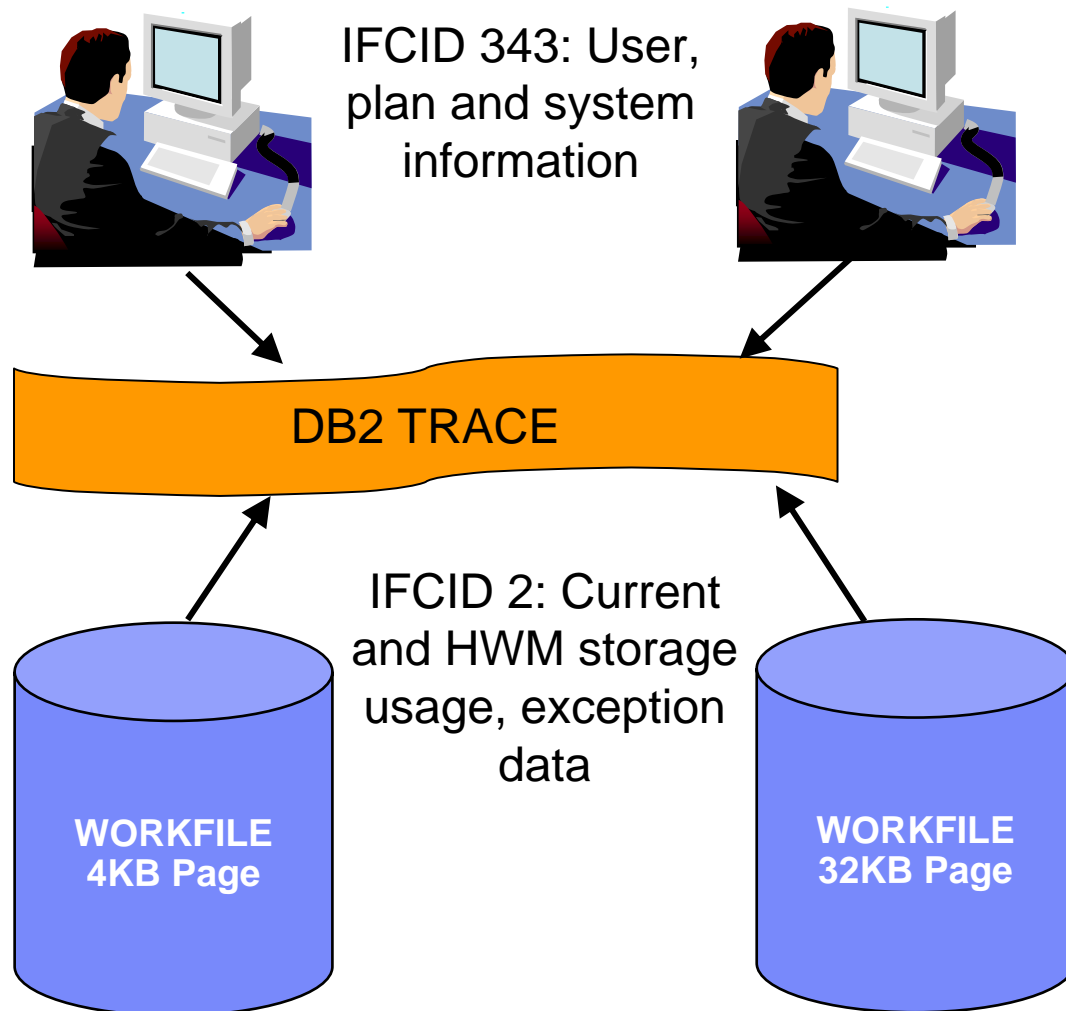
Controlling Temporary Space Utilization



- Control of temporary space utilization at the agent level
- New ZPARM, MAXTEMPS (macro DSN6SYSP, panel DSNTIP9)
- If MAXTEMPS is exceeded for any given agent:

```
SQLCODE = -904, ERROR:  
UNSUCCESSFUL EXECUTION CAUSED BY AN  
UNAVAILABLE RESOURCE.  
REASON 00C90305,  
TYPE OF RESOURCE 100, AND  
RESOURCE NAME = 'WORKFILE DATABASE'  
SQLSTATE = 57011
```


Monitoring Temporary Space Utilization



- To allow monitoring of temp space usage at the DB2 subsystem level, including the number of times MAXTEMPS has been exceeded, IFCID 2 is updated with new counters.
- New IFCID 343 (Performance Trace, Class 3) is written when the MAXTEMPS limit for an agent is exceeded.

Planning For Converged TEMP Space

- Migration from DB2 V8: to reclaim TEMP database storage, *YOU* must drop the TEMP database and reallocate the storage.
 - Recommendation: do not drop the TEMP database until you are sure that you will not return to V8.
- DB2 9 installation generates DDL to create DB2-managed, instead of user-managed, WORKFILE table spaces in the SYSDEFLT storage group.
- Find grants to TEMP databases, grant to WORKFILE database.
- Recommendation: set the 'DSVCI' ZPARM to YES to allow DB2 to match VSAM CI size to table space page size.
- Ensure you have 32KB WORKFILE table spaces for Declared Global Temporary Tables and Static Scrollable Cursors.
- Plan for changed error message text for DGTT and SSC.



SWG BetaWorks

Migration, Coexistence and Fallback

Prerequisites

- **Processor requirements:**
 - Any processor that supports z/Architecture, including IBM System z9 and z10 and zSeries z800, z890, z900, z990 families of processors .
 - DB2 9 will probably require increased real storage for a workload compared to DB2 V8
- **Software Requirements:**
 - z/OS V1.7
 - DFSMS Version 1 Release 7
 - Language Environment Base Services
 - z/OS Version 1 Release 7 Security Server (RACF)
 - IRLM V2.2
 - z/OS Unicode Services and appropriate conversion definitions are required.

Primary DB2 9 New Changes & Incompatibilities

- Info APAR II12423, toleration APARs PK11129 (UK90008) PK33664, PK39850, PK38799, PK44884, V8 APAR PK31841 (DSNTIJP9)
- Migrate only from DB2 V8 NFM with expanded BSDS – DSNJCNVB
- Removed from DB2 9
 - DB2-managed stored procedures → WLM
 - Extenders for AIV, Text, Net Search
 - Net.Data DB2 Estimator Old JDBC / SQLJ driver
 - Ability to create simple table space → segmented structure
- Deprecated in DB2 9: XML Extender
- Process:
 - Rebind required for V3 and prior plans & packages
 - PDSE is required for SDSNLOAD
- Previously, some **DSNWZP** output was formatted differently to the DSN6* macros. E.g. in **DSN6SYSP**, the EDMPOOL size is specified in KB; **DSNWZP** reports it in bytes. **DSNWZP** output is now formatted like the DSN6* macros.

Requirements for Dependent Functions

- Volume based utilities require:
 - z/OS 1.8
 - DFSMSHsm v1.8
 - DFSMSdss
 - FlashCopy Version 2
 - Coming soon: DB2 support for incremental FlashCopy (z/OS APAR OA17314)
- Group Bufferpool Batching:
 - CF level 12
 - CF level 13 recommended for castout and lock table cleanup enhancements
 - CF level 15 or 14 recommended when using System Managed Duplexing for lock or SCA structures
- Trusted Contexts and Roles
 - z/OS 1.8
- DB2 requires Java 1.4.2, but XML Schema functionality requires Java 1.5

Automatic Creation of Objects Premigration Jobs

- **Before commencing migration to DB2 9, make sure there are no incomplete object definitions in the DB2 Version 8 catalog.**
 - For example, if a table has a primary key/unique key defined but the required index does not exist, the table definition is considered incomplete.
- **Complete or drop all such objects *before* beginning migration because their behaviour will be different in DB2 9.**
 - For example, if you attempt to create an enforcing primary key index to complete a table definition in DB2 9, if the residing table space is implicitly created, then the index is treated as a regular index instead of an enforcing index.
- **A query and a report are added to the DB2 9 premigration readiness jobs, DSNTIJPM and DSNTIJP9, to identify incomplete object definitions:**
 - Job DSNTIJPM ships with each release of DB2 and should be run on the previous release to identify premigration catalog cleanup that needs to be performed before new release migration.
 - In some but not all cases, DSNTIJPM may provide DDL or utility statements for performing the cleanup.
 - Job DSNTIJP9 is the same as the DB2 9 DSNTIJPM but will be shipped with V8 in order to maximize the time available to prepare the DB2 Version 8 catalog for migration to DB2 9.

Other z/OS Planning

- **Current CICS, IMS, WebSphere, COBOL, PL/I, Omegamon ...**
- **z/OS APARs:**
 - OA16303 -- z/OS XML System Services (XML parser)
 - OA16037 -- z/OS Unicode Services (COLLATION_KEY)
 - Maintenance needed for DDF 64-bit storage:
 - OA14623 z/OS UNIX System Services
 - OA15634 z/OS UNIX System Services
 - PK27083 TCP/IP
 - PK11661 TCP/IP
 - PK15034 TCP/IP
 - PK16990 TCP/IP
 - PK18527 TCP/IP
 - PK14920 TCP/IP
 - PK15444 TCP/IP

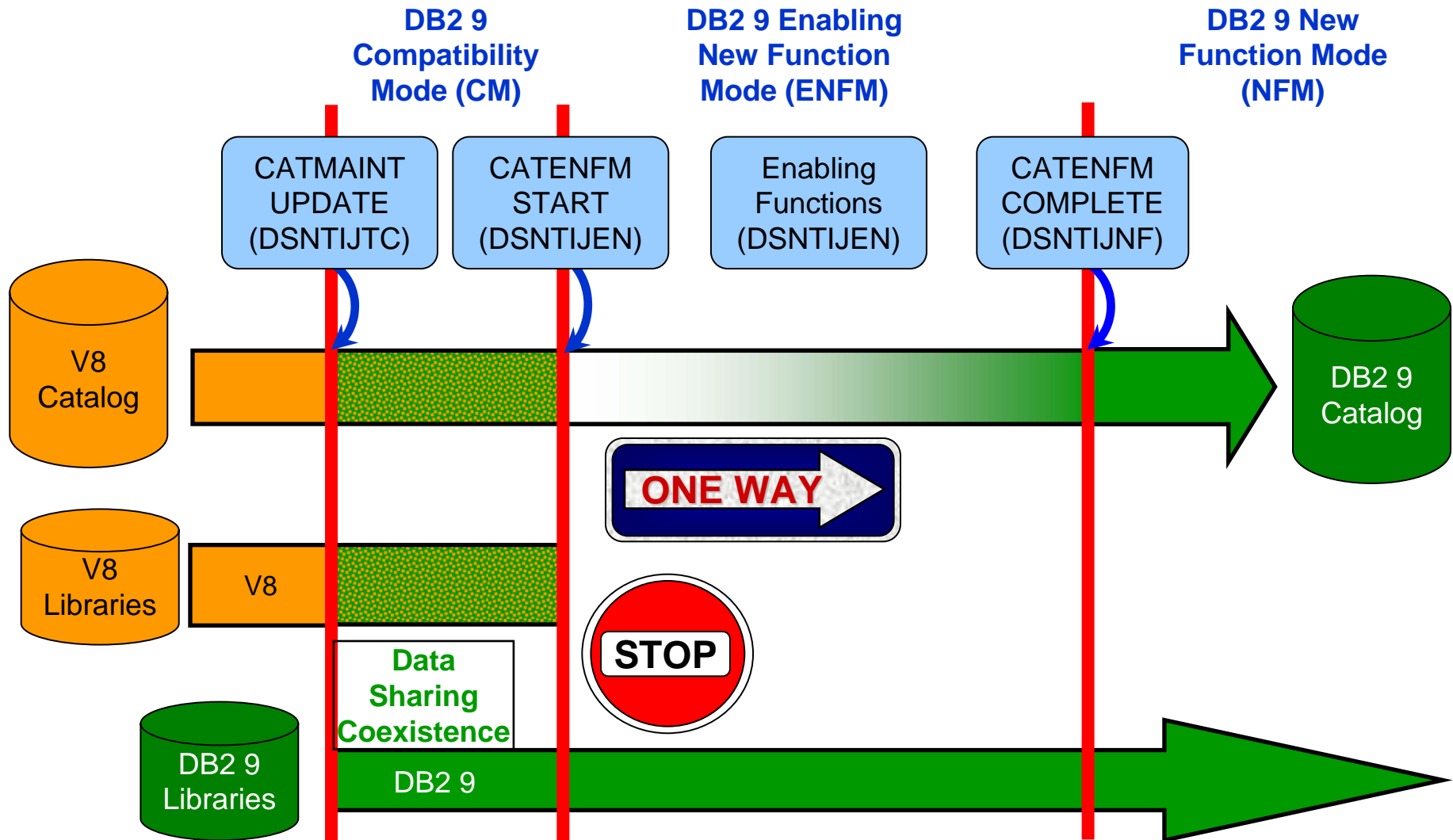


SWG BetaWorks

The Migration Process

Migration Overview

Migration Overview



The Modes – An Overview

CM Compatibility Mode - This is the mode DB2 is in when DB2 9 is started for the first time after V8. It will still be in CM when migration job DSNTIJTC has completed. No new function can be executed in CM. Data sharing systems can have V8 and DB2 9 members in this mode. DB2 can only migrate to CM from V8 NFM.

ENFM Enabling New Function Mode - This mode is entered when CATENFM START is executed (the first step of job DSNTIJEN). DB2 remains in this mode until all the enabling functions are completed. Data sharing systems can only have DB2 9 members in this mode.

NFM New Function Mode - This mode is entered when CATENFM COMPLETE is executed (the only step of job DSNTIJNF). This mode indicates that all catalog changes are complete and new function can be used.

ENFM* This is the same as ENFM but the * indicates that at one time DB2 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 ENFM* it can not fallback to V8 or coexist with a V8 system.

CM* This is the same as CM but the * indicates that at one time DB2 was at a higher level. Objects that were created at the higher level can still be accessed. When DB2 is in CM* it can not fallback to V8 or coexist with a V8 system.

The Migration Jobs

DSNTIJTC is used to migrate from V8 to DB2 9.

DSNTIJEN is used to move DB2 from CM to ENFM, or to complete the changes that are made in ENFM. If the job does not complete it can simply be resubmitted, and it will continue from where it was stopped.

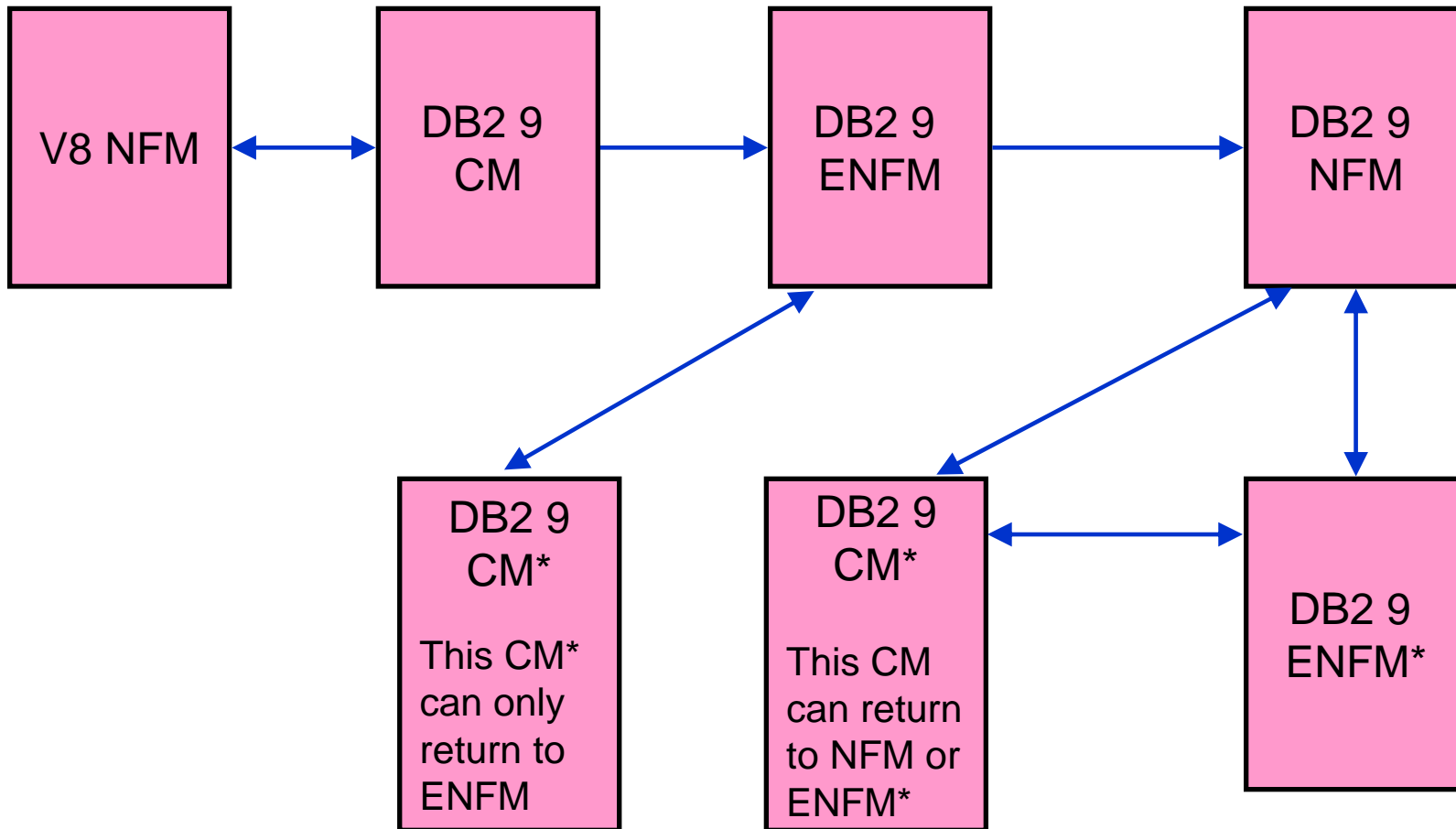
DSNTIJES takes DB2 to ENFM* from CM* or NFM. Once run, exploiting new function is no longer allowed (if DB2 went from NFM to ENFM*) but objects created while in NFM can still be accessed.

DSNTIJCS returns DB2 to CM* from ENFM, ENFM* or NFM. Once run, exploiting new function is no longer allowed (if DB2 went from NFM to CM*) but objects created in NFM can still be accessed. In a data sharing system all members must remain at DB2 9.

DSNTIJNF takes DB2 to NFM.

Mode Flows

The allowed flows from one mode to another





SWG BetaWorks

The Migration Process

New Release Migration

Preparing DB2 V8 for Migration

- Apply the SPE APAR, PK11129 and fixes PK33664, PK39850, PK38799, PK44884 to DB2 V8.
- **Non-Data Sharing and Data Sharing**
 - DB2 V8 must be started with the SPE applied, or migration to DB2 9 is terminated.
- **Data Sharing**
 - Before migrating a data sharing member to DB2 9, all other started DB2 V8 members must have the fallback SPE applied.
 - The fallback SPE must be on all active V8 group members for DB2 9 to start.
 - Coexistence between DB2 V8 NFM and DB2 9 CM only.



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

Planning: Release Incompatibilities

■ Incompatibilities: Check the book

- The DBPROTCL ZPARM is removed - the default will always be DRDA.
- All stored procedures must be WLM enabled.
- The legacy JDBC driver will not work. This happens in CM as well as NFM.
 - Data sharing groups with V8 in coexistence with DB2 9 CM will fail if they reference the legacy JDBC driver.
 - Recommendation: modify your WLM-SPAS JCL to use the Universal JDBC Driver while still on V8, prior to migration to DB2 9.
- The RELCURHL=NO ZPARM option is removed.


Planning: Migration Considerations

■ Considerations:

- The new stored procedures for XML Schema Support cannot be used in CM.
- Prepare to allow for 'J' instances for data sets backing LOB table spaces.
- The output of various –DISPLAY commands has changed.
- Some messages have also changed.
- There are new IFCIDs, and more changed ones.
- Message DSN7101I (-DISPLAY GROUP) shows the new modes.

Planning: Migration Considerations (cont.)

■ Considerations (cont.):

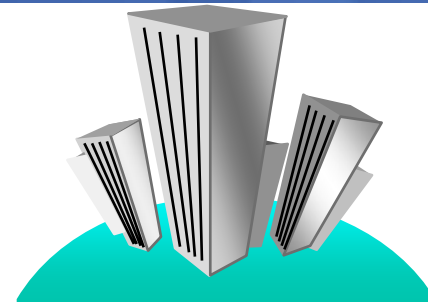
- The default setting for MGEXTSZ is YES.
- It is no longer possible to create a simple table space. SEGSIZE, if not specified, defaults to 4.
- The values reported by DSNWZP are in the same format as for the DSNZPARM macros.
- If you delete your TEMP database in CM, you will need to recreate it if you return to V8.
- The page-range screening improvement is available in CM
 - You must rebind on V8 if falling back from DB2 9 compatibility mode to V8. 

Migration to Compatibility Mode

- Migration to CM is achieved by running the CATMAINT UPDATE job, DSNTIJTC:
 - Adds new catalog table spaces and tables.
 - Adds new columns to existing catalog tables.
 - Adds new meanings to existing catalog columns.
 - Adds new indexes for new and existing catalog tables.
 - Adds new check constraints for catalog columns.
 - Modifies existing check constraints for catalog columns.

DB2 Catalog Evolution

The DB2 catalog continues to grow with every DB2 release.



DB2 Version	Table Spaces	Tables	Indexes	Columns	Table Check Constraints
V1	11	25	27	269	N/A
V3	11	43	44	584	N/A
V5	12	54	62	731	46
V6	15	65	93	987	59
V7	20	84	117	1212	105
V8	21	87	128	1286	105
DB2 9	28	106	151	1668	119

Doesn't include objects for XML Schema Repository!

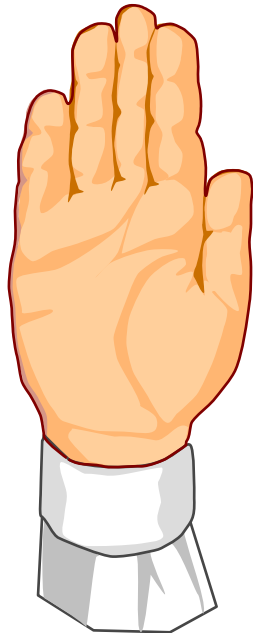


SWG BetaWorks

The Migration Process

Conversion to New Function Mode

ENFM and NFM Considerations



STOP!
THINK!

**Attention: You cannot return to DB2 V8
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 9 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

- **Enabling New Function Mode – Job DSNTIJEN**
 - CATENFM START places DB2 in Enabling New Function Mode.
 - **Data sharing groups must only have DB2 9 members.**
 - Add BIGINT and VARCHAR columns to SYSTABLESPACESTATS and SYSINDEXSPACESTATS.
 - Create DSNRTX03 index on SYSINDEXSPACESTATS.
 - The first time DSNTIJEN runs, DB2 saves the RBA or LRSN of the system log in the BSDS.
 - Convert SYSOBJ to 16k page size.
 - SWITCH RTS TO CATALOG TABLES

Moving to New Function Mode

■ New Function Mode – Job DSNTIJNF

- Enables New Function

■ Considerations:

- System dump titles now indicate the DB2 mode.

■ Incompatibilities:

- Extensions for SQL Procedures

- Name resolution differs in CM and NFM.

- DSNHxxxx messages are issued for external SQL procedures (as before): **SQLCODEs are issued for native SQL procedures (new).**

- Up to 500 user-defined indexes can be created on the catalog in NFM. Not supported in CM.

- The WLMENV name for stored procedures can now be 32 characters long. Not supported in CM.

Moving to New Function Mode (cont.)

■ Incompatibilities:

- For an implicitly created table space, explicitly drop required objects
- Create a LOB table space in an implicitly created database.
- Drop an auxiliary table for a base table in an implicitly created table space.
- DROP TABLESPACE for an implicitly created LOB table space.
- Invocation of unqualified UDFs which share the name of new built-in functions will result in the new built-in function being invoked instead



DSNX@XAC Considerations

- Changes to the Access Control Authorization Exit (DSNX@XAC):
 - CURRENT SCHEMA makes the qualifier of tables, views and aliases the schema name. This may or may not identify the owner, so field XAPLOWNR is introduced to identify the owner.
 - Trusted Contexts introduces support for roles, and therefore new field XAPLROLE is introduced to accommodate roles.
 - Again for roles, new flag XAPLRAOO is introduced that indicates whether the role is the owner of any objects created. XAPLUCHK identifies the role that will be the owner, and XAPLUCKT identifies if XAPLUCHK is a role.
 - If an application owned by a role creates an object, XAPLROLE contains the runner's role, whereas XAPLUCHK contains the application's owning role.
 - XAPLUCKT identifies the authorization id type of XAPLUCHK. XAPLONRT (also new) identifies the authorization id type of the object owner. These are single byte indicators – blank (' ') for standard authorization ids, 'L' for roles.

DSNX@XAC Considerations (cont.)

- Macro DSNXAPRV contains new values for new privileges (field XAPLPRIV):
 - DEBUGSESSION system privilege
 - The rename index privilege.
 - OSC START PROFILE and STOP PROFILE privileges.
- To prevent use of an incompatible version of the access control authorization exit routine (ACAER) with DB2 9 the ACAER interface includes a means to verify that the ACAER in use is compatible with all modes of DB2 9.
- At DB2 startup, as part of the ACAER initialization process, the ACAER will indicate whether it is compatible with DB2 9.
- If it indicates that it is compatible then the exit will be activated.
- If it does not indicate that it is compatible, DB2 issues message DSNX210I, ACAER abends (reason code 00E70016) and DB2 terminates.
- A flag bit is defined in the EXPL to indicate compatibility with DB2 9.
 - ON indicates compatible and OFF indicates incompatible. DB2 sets it to OFF on invoking the ACAER for initialization during DB2 startup.



SWG BetaWorks

Migration and Fallback

Returning to a prior release or mode

Returning to DB2 V8

- **Fallback to DB2 V8 NFM with the fallback SPE (PK11129) applied is only supported from DB2 9 CM.**
- Plans and packages bound (or rebound) in DB2 9 CM are automatically rebound in V8 NFM.
- Objects with functional dependencies are indicated by an 'M' in the IBMREQD catalog columns and are frozen on return to DB2 V8.
- In a distributed environment, DB2 9 can communicate with DB2 V8 only.

Return to prior DB2 9 Modes

- It is possible to return to CM* from ENFM.
- It is possible to return to ENFM* or CM* from NFM.
- It is possible to return to CM* from ENFM*.
- If you return from NFM (or ENFM*):
 - Objects created and new SQL features exploited will still function in ENFM* and CM*
 - No new objects can be created and no new functions can be exploited until you return to NFM.



SWG BetaWorks

Migration Process Changes

New Features and DSNZPARM-related changes

Installation CLIST and Install/Migrate Job Changes

- **This section covers the changes to the installation CLIST and the installation and migration jobs.**
 - XML Schema Support
 - Enhancements for Optimistic Locking and Update Detection
 - Converge Temp Space
 - Volume Based Utility Enhancements
 - Relief for Sequential Inserts
 - Decimal Floating Point data type
 - LOB File Reference
 - Autonomic reopt based on parameter marker value change
 - Automatic Creation of Objects
 - Removal of RELCURHL ZPARM
 - Improve RLF usability for middleware servers
 - Removal of Private Protocol
 - Increase of WLM Environment name to 32 characters
 - Support of built-in procedure SQLJ.ALTER_JAVA_PATH
 - Removal of support for the DB2-managed stored procedure address space

XML Schema Support and Optimistic Locking Support

- **For XML Schema Support, the installation process is changed to:**
 - Create the XSR stored procedures.
 - Create the validation functions.
 - Create the decomposition stored procedure.
 - Create the XSR tables, database and table space.
 - Bind the SQL packages.
 - Run the JAVA XSR bind utility.
 - If this is not run, the XSR_COMPLETE stored procedure will fail with SQLCODE -443 and additional error information will indicate a -805 SQLCODE.
- **Optimistic Locking Support:**
 - Installation job DSNTJIN is changed to increase the primary BSDS allocation by 12 blocks (1 3390 track) to support optimistic locking.

Converge Temp Space

- **New installation panel, DSNTIP9 allows you to specify:**
 - The number of 4K and 32K table spaces to create.
 - The total amount of storage for each page size.
 - The segment size for each page size (cannot be updated in migrate mode).
 - The setting for MAXTEMPS.
- **The installation process creates the work file database and table spaces.**
- **The migration process reuses the work file database and table spaces from V8; extra work file table spaces can be added**
- **DSNTIJSG no longer creates STOGROUP SYSDEFLT – now done by DSNTIJTM.**
- **In migration mode, REXX EXEC DSNTWFG generates the DDL for additional table spaces (DSNTIJTM), and ensures there are no naming conflicts.**
 - Sample job DSNTIST allows you to create additional workspaces after migration/installation.
- **DSNTIJDE deletes the DB2-manged 4K and 32K work files – DB2 must be down for DSNTIJDE to run, so DROP TABLESPACE is not possible.**

Volume-based Utility Enhancements

- **The volume-based utility enhancement introduces 4 new ZPARMs (see panel DSNTIP6):**
 - **DSN6SPRM SYSTEM_LEVEL_BACKUPS**
 - Should RECOVERY use system-level backups as a recovery base.
 - Default value: YES
 - **DSN6SPRM STORE_RECOVER_FROM_DUMP**
 - The default behaviour when using a system-level backup as a recovery base for RESTORE SYSTEM or RECOVER: use the disk copy (NO); the tape copy (YES).
 - Default value: NO
 - **DSN6SPRM UTILS_DUMP_CLASS_NAME**
 - The DFSMSHsm dump class to be used by RESTORE SYSTEM and by RECOVER when restoring from a system-level backup that has been dumped to tape.
 - Only valid if STORE_RECOVER_FROM_DUMP is YES
 - **DSN6SPRM RESTORE_TAPEUNITS**
 - Specifies the default value for the maximum number of tape units or drives that can be used by RESTORE SYSTEM.
 - Default value NOLIMIT means that RESTORE SYSTEM will use as many tape units or drives as necessary to restore the system-level backup.

Decimal Floating Point

- New data type Decimal Floating Point introduces a new DSNHDECP parameter, DEF_DECFLOAT_ROUND_MODE which specifies the default decimal floating point rounding mode:
 - ROUND_CEILING:
 - ROUND_FLOOR
 - ROUND_UP
 - ROUND_DOWN
 - ROUND_HALF_UP
 - ROUND_HALF_DOWN
 - ROUND_HALF_EVEN

Automatic Creation of Objects new ZPARAMs

- **Automatic Creation of Objects introduces 5 new ZPARAMs, and changes an existing ZPARAM default:**
 - **DSN6SYSP IMPTSCMP**
 - Specifies whether compression for table spaces created in an implicitly created database is enabled or not. The default value is NO.
 - **DSN6SYSP IMPDSDEF**
 - Specifies whether the data set for a table space created in an implicitly created database should be defined (corresponds to the DEFINE YES/NO clause in CREATE TABLESPACE). The default value is NO.
 - **DSN6SYSP TBSBP8K, TBSBP16K and TBSBP32K**
 - These specify the default 8K, 16K and 32K buffer pools to use for table spaces created in an implicitly created database. Existing ZPARAM TBSBPOOL is used for 4K table spaces.
 - The default for existing ZPARAM **DSN6SYSP MGEXTSZ** (Specify whether secondary extent allocations for DB2-managed data sets are to be sized according to a sliding scale) is changed to YES.
- **New/updated panels: DSNTIP2; DSNTIP6; DSNTIP7; DSNTIPD.**

Other Changes

- Removal of RELCURHL ZPARM – updated panel DSNTIP8 – possible release incompatibility
- Improved RLF usability
- Positioning for the removal of Private Protocol – updated panel DSNTIP5
- Increase the maximum size of the WLM environment to 32 characters – updated panel DSNTIPX
- New built-in procedure SQLJ.ALTER_JAVA_PATH
- Removal of support for DB2-managed stored procedures.
- DSNTINST update mode panel DSNTIPB changed.



SWG BetaWorks

Installation and Migration Process Changes

Enhancements

Enhancements to the Installation Process

- **DB2 9 introduces the following installation enhancements:**
 - These enhancements do not affect ZPARMS or objects produced by the installation or migration processes.
 - DB2 Installation support for SMS-managed storage
 - Separate prefix / suffix fields for installation-defined data sets
 - Optional tracking of hidden keyword parameter settings
 - DSNTIJRX is now customized by the CLIST, and becomes a required job
 - Retain migration input value for DSNTIPX field 1 WLM PROC NAME
 - Return all DSNWZP output in same format as DB2 ZPARMs

DB2 Installation Support for SMS-managed Storage

```
DSNTIPA2      INSTALL DB2 - DATA PARAMETERS PANEL 1
===>
Check parameters and reenter to change:
 1 CATALOG ALIAS      ===> DSNCAT      Alias of VSAM catalog for DB2 subsystem
                                     data sets
 2 DEFINE CATALOG     ===> YES        YES or NO
 3 USE SMS             ===> NO        Use SMS for installation-defined objects

PRESS:  ENTER to select  RETURN to exit  HELP for more information
```

SMS-managed Storage – New Panel DSNTIPA3

```

DSNTIPA3          INSTALL DB2 - DATA PARAMETERS PANEL 2
===>
Check parameters and reenter to change:
 1 PERMANENT UNIT NAME      ==> 3390          Device type for MVS catalog
                                   and partitioned data sets
 2 TEMPORARY UNIT NAME     ==> SYSDA        Device type for
                                   temporary data sets
                                   ----- SMS -----
                                   VOL/SER      DATA CLASS      MGMT CLASS      STOR CLASS
                                   -----
 3 CLIST ALLOCATION         ==> DSNV01      ==>              ==>
 4 NON-VSAM DATA         ==> DSNV01      ==>              ==>
 5 VSAM CATALOG, DEFAULT, ==> DSNV02      ==>              ==>
  AND WORK FILE DATABASE
 6 DIRECTORY AND CATALOG  ==>              ==>              ==>
  DATA
 7 DIRECTORY AND CATALOG  ==>              ==>              ==>
  INDEXES
 8 LOG COPY 1, BSDS 2     ==>              ==>              ==>
 9 LOG COPY 2, BSDS 1     ==>              ==>              ==>

PRESS:  ENTER to select  RETURN to exit  HELP for more information
    
```

Separate SMP/E and Installation Prefixes/Suffixes

```
DSNTIPA1  DB2 9 INSTALL, UPDATE, MIGRATE, AND NFM - MAIN PANEL
```

```
===>
```

```
Check parameters and reenter to change:
```

```
1 INSTALL TYPE          ===> INSTALL      Install, Migrate, NFM, or Update
2 DATA SHARING         ===> NO           Yes or No (blank for NFM or Update)
```

```
Enter the data set and member name for migration only. This is the name used
from a previous Installation/Migration from field 9 below:
```

```
3 DATA SET(MEMBER) NAME ===>
```

```
For DB2 SMP/E libraries (SDSNLOAD, SDSNMACS, SDSNSAMP, SDSNCLST, etc.), enter:
```

```
4 LIBRARY NAME PREFIX   ===> DSN910
5 LIBRARY NAME SUFFIX    ===>
```

```
For install data sets (NEW.SDSNSAMP, NEW.SDSNCLST, RUNLIB.LOAD, etc.), enter:
```

```
6 DATA SET NAME PREFIX ===> DSN910
7 DATA SET NAME SUFFIX ===>
```

```
Enter to set or save panel values (by reading or writing the named members):
```

```
8 INPUT MEMBER NAME     ===> DSNTIDXA   Default parameter values
9 OUTPUT MEMBER NAME    ===>           Save new values entered on panels
```

```
PRESS:  ENTER to select  RETURN to exit  HELP for more information
```

Tracking Opaque System Parameter Settings

- Two conventions for documented system parameters:
 - Externalized
 - Described in the Installation Guide, exposed in DSNTIJUZ, supported by the installation panels.
 - Opaque
 - Described in the Installation Guide, exposed in DSNTIJUZ, not supported by the installation panels.
- The problem: opaque system parameters are susceptible to regression, especially at migration time.

Tracking Opaque System Parameter Settings (cont.)

- **The solution:**

- The Installation CLIST scans the input member(s) for any opaque DB2 9 parameters.
- It validates these and writes them out to DSNTIJUZ (if validation is passed).
- Any invalid entries result in the following message being displayed on DSNTIPA1:
 - **DSNTxxxx ENTRY FOR <parameter-name> IGNORED BECAUSE THE SETTING IS INVALID**
- You must (initially) enter any entries in the input member

- **This enhancement provides a central place for maintaining the current settings, and helps to ensure that they are used when the system is updated or migrated.**

DSNTIJRX, WLM PROC NAME and DSNWZP

- Installation/migration job **DSNTIJRX**, which binds the packages for DB2 REXX support, is no longer optional but required (it is needed by **DSNTWFG** to create table spaces in the work file database).
- In previous releases, the value for **WLM PROC NAME** on panel **DSNTIPX** was not retained for update or migration processing by the installation CLIST. This is now corrected.
- In previous releases, some **DSNWZP** output was in a different format to that used by the DSN6xxxx macros. For example, in **DSN6SYSP**, the size of the EDMPOOL is specified in KB; **DSNWZP** reports it in bytes. **DSNWZP** output is now in the same format as for the DSNZPARM macros.
 - **WARNING**: This is a release incompatibility for any application which processes DSNWZP output and which reformats it to compensate for any inconsistencies.



SWG BetaWorks

Migration and Fallback

Sample Applications

Sample Applications

- New and changed sample applications are provided to test the functionality of new features:
 - XML (DRDA and Host Interfaces)
 - Extensions for SQL Procedures
 - BIGINT, VARBINARY and BINARY data types
 - Decimal Floating Point data type
 - LOB File Reference Variables
 - Automatic Creation of Objects
- In CM you should use the V8 sample applications. The DB2 9 samples are intended to verify entry to NFM.
- An isolation level of UR for SPUFI is allowed. The default isolation level for SPUFI is now CS.

Extensions for SQL Procedures

- New SQL pre-processing options introduced in DSNTEP2 and DSNTEP4.
 - **SQL** specifies that SQL statements are to be pre-processed as in earlier versions of DB2: DSNTEP2 and DSNTEP4 copy multi-line SQL statements into a buffer for PREPARE, removing SQL comments. This is the default behaviour.
 - **SQLCOMNT** specifies that SQL-style comments are to be passed in the PREPARE buffer instead of being discarded. A line formatting character is appended after any detected SQL comment, preventing loss of SQL comments, to assist with diagnostics and debugging, and to retain the format of the SQL procedure.
 - **SQLPL** specifies that SQL-style comments are to be preserved, but a line formatting character is appended to each input line that ends without a token split. This preserves the SQL comments and multi-line format of an SQL procedure, for better diagnostics and debugging. It also helps retain the formatting of the SQL procedure when stored in the DB2 catalog.
 - SPUFI, DSNTEP2, and DSNTEP4 are updated to handle C-style comments, which are valid in SQL processed by DB2 9. When parsing of the SQL input by these applications encounters an un-delimited occurrence of the sequence '/*', SPUFI, DSNTEP2, and DSNTEP4 will simply copy all subsequent text, unparsed, into the PREPARE buffer for PREPARE, until the sequence '*/' is found.

DSNTIAUL and LOB File Reference Variables

- Use LOB file reference variables to extract full LOB values to external data sets.
- Data set naming convention: `<prefix>.Q<i>.C<j>.R<k>`:
 - `<prefix>` is the data set name prefix (cannot exceed 26 characters).
 - `Q<i>` is the (`<i>-1`)th query, in the range from 00 to 99.
 - `C<j>` is the (`<j>-1`)th column, in the range from 000 to 999.
 - `R<k>` is the (`<k>-1`)th row fetched, in the range from 0000000 to 9999999.

`<prefix>`

```
DSN910.DSNTIAUL.Q00.C005.R0000001
```

- New LOBFILE option:

```
RUN PROGRAM(DSNTIAUL) PLAN(DSNTIB91) -  
  PARS('LOBFILE(DSN910.DSNTIAUL)') -  
  LIB('DSN910.RUNLIB.LOAD')
```
- If LOBFILE is not specified, LOB data is handled as in previous releases.

“Thank You for listening”

If you have any questions on this DB2 9 for z/OS session, then please send them to the BetaWorks team at:

Ian_Cook@uk.ibm.com
FLETCHPL@uk.ibm.com

