



DB2 11 for z/OS Application Compatibility – What you Need to Know

Christopher J. Crone
IBM

Session Code: A15
Thursday October 17th – 8:30 – 9:30 AM | Platform: DB2 for z/OS





Disclaimer

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.



Agenda

- Overview of the Problem
 - Why is DB2 Introducing this new function
- Solution Approach
- Solution Detail
- How to Identify Potential Issues
- Summary and Questions



Abstract

DB2 11 for z/OS introduces the concept of Application Compatibility to simplify migration.

This session will introduce the concept and provide the information you need to know to exploit this new capability to simply your migration from DB2 10 to DB2 11.

We will also cover how to exploit this new capability to identify changes you need to make to applications that may be affected by incompatible changes in DB2 11.

Overview of the Problem





Problem

- Changes to SQL and XML behavior are problematic to adopt
 - It is difficult to understand the impact of change to thousands of applications
 - Change needs to be controlled at the application level
- Service stream incompatibilities cause maintenance issues, especially when key PTFs pre-req the PTFs that cause the changed behavior
- Changes at a release boundary allow for some planning, but it is difficult to sync application changes with release migration plans



Problem - Standards

- There are IBM and industry standards for SQL that DB2 for z/OS must be compliant with.
- DB2 for z/OS may be out of compliance because of defect or incomplete implementation
 - Fixing the compliance issue introduces an incompatible change that may break existing applications
 - These incompatible changes are saved up and introduced on DB2 release boundary
- Example of incompatibility introduced in DB2 10
 - CHAR function results (also for VARCHAR and CAST of these data types)
 - Leading zeroes no longer returned when there is a decimal point
 - Originally PMR raised to report as error
 - Working as designed to conform to SQL Standards
 - Advice in manuals was to change application code to accommodate new behavior



Problem – New Function

- New SQL Function can create incompatibilities
- Example - DB2 10 introduced implicit casting
 - Implicit casting enables characters to be implicitly converted to numbers and numbers to be implicitly to characters
CREATE TABLE T1 C1 INTEGER;
INSERT INTO T1 VALUES ('123'); -- Fails in DB2 9, works in DB2 10
 - Function resolution rules, and the functions themselves have been changed to allow both character and number interchangeably
SELECT 123 || 'abc' FROM T1; -- Fails in DB2 9, works in DB2 10
- By Definition – most new function is incompatible in some way with existing behavior in some way
 - Most common case is removal of an SQLCODE



Solution Requirements

- Do not force application changes to address incompatible SQL changes on a release boundary
- Allow changes to be introduced at the application (package) level
- Provide more warning and time to customers for incompatible changes to be addressed
- Provide a mechanism to identify applications that need to be analyzed for incompatible behavior
- Provide a 'fence' so DB2 can address errors vs. new incompatible change

Solution Approach





Proposed Solution – Separate Release Migration and Application Migration to new release function

- Limit SQL DML and XML incompatibilities when possible
- Provide a mechanism to identify applications affected by DML and XML changes
- Provide a mechanism to make changes at an application (package) level
 - DB2 11 will be the initial deployment of this capability
 - DB2 10 will be the lowest level of compatibility supported
 - This mechanism will enable support for up to two back level releases (N-2).
For Example:
 - DB2 11 supports DB2 10 and DB2 11
 - DB2_2* supports DB2 10 and DB2 11 and DB2_2*
 - DB2_3* supports DB2_1* and DB2_2* and DB2_3*
 - DB2_4* supports DB2_2* and DB2_3* and DB2_4*

*DB2_2, DB2_3, and DB2_4 are theoretical releases used in the above example strictly for demonstrative purposes. This is not a commitment by IBM to actually produce any of these releases.

Solution Detail





Solution Details

- New ZPARM APPLCOMPAT for DEFAULT of bind option
 - Will be set to V10R1 on Migration to V11
 - Since this is a new bind option for V11, previously bound packages will not have specified this parm. If a package is bound in V10 or below, APPLCOMPAT(V10R1) will be assumed.
 - The first time a package is bound in V11, it will pick up the DEFAULT.
 - In CM only V10R1 may be specified
 - V11R1 for new install
 - Default on subsequent migrations will be the down level release - e.g. On migration to V_2, default will be V11R1
- REBIND picks up the current value or default
 - Customers can override rebind with any “valid” value



Solution Details

- New PACKAGE bind (BIND/REBIND) option (APPLCOMPAT) and special register
 - Applies to anything that has a package (including things like SPs, UDFs, Triggers...)
 - Trigger Packages are tied to an OBJECT, but compatibility is tied to the package bind options
 - BIND option applies to static SQL, and the default for the special register
 - The CURRENT APPLICATION COMPATIBILITY special register applies to dynamic SQL
 - The default behavior for this special register in a SP or UDF is to not inherit this special register.
 - Special Register need to be set in properties/ini file for (ODBC/JDBC/.NET) drivers
 - DSN_PROFILE tables instead of, or in addition to, a SPECIAL REGISTER may be used to control remote applications

BIND/REBIND

- BIND/REBIND PACKAGE

Syntax

BIND/REBIND

```
>-----+-----  
>      '----- APPLCOMPAT---(--V10R1---)-----'  
      (---V11R1---)
```

- REBIND TRIGGER

Syntax

REBIND TRIGGER

```
>-----+-----  
>      '----- APPLCOMPAT---(--V10R1---)-----'  
      (---V11R1---)
```

DSNT298I

csect-name ATTEMPT TO USE COMMAND OR OPTION *command* WHEN THE APPLICATION COMPATIBILITY IS SET FOR A PREVIOUS LEVEL

Explanation

A new command or command option was issued when application compatibility bind option is set to a prior DB2 release. Another possibility is when a command or command option introduced in release n was used but the application compatibility bind option is lower than n.

command The command or command option that can be used only when the application compatibility bind option is set to the compatible release.

System action

The command or command option is not processed.

System programmer response

To use the command or command option, specify the application compatibility bind option to at least the release that introduces this bind option or command.

CREATE/ALTER Routines

- CREATE/ALTER PROCEDURE (SQL native):

```
Syntax
>>-CREATE/ALTER PROCEDURE ----->
option-list:
>-----+-----
>< |---APPLCOMPAT V10R1--|
+---APPLCOMPAT V11R1--+
```

- CREATE/ALTER FUNCTION (SQL native):

```
Syntax
>>-CREATE/ALTER FUNCTION ----->
option-list:
>-----+-----
>< |---APPLCOMPAT V10R1--|
+---APPLCOMPAT V11R1--+
```

Updated Catalog Columns

Table Name	Column Name	Data Type	Description
SYSPACKAGE	APPLCOMPAT	VARCHAR(10)	The DB2 release level that the SQL statements in this package is compatible with.
SYSPACKCOPY	APPLCOMPAT	VARCHAR(10)	The DB2 release level that the SQL statements in this package is compatible with.



Solution Details (cont)

- ALL SQL DML and XML changes are fenced by these options. This includes:
 - **New SQL function**
 - Incompatible SQL changes such as:
 - Function Resolution rules
 - Result type changes
 - SQLCODE changes
 - BIF changes
 - ...

SQLCODE -4743

-4743

ATTEMPT TO USE A FUNCTION WHEN THE APPLICATION COMPATIBILITY IS SET FOR A PREVIOUS LEVEL

Explanation

Functions that this release of DB2® introduces cannot be used when application compatibility setting is for a prior DB2 release. An attempt was made to execute one of these functions. Another possibility is when a function introduced in release n was used but the application compatibility setting is lower than n.

To use the new functions that have been introduced in this release of DB2, the application compatibility bind option or special register CURRENT APPLICATION COMPATIBILITY must be set to the current release. An attempt to use functions introduced in release n requires the application compatibility bind option or special register CURRENT APPLICATION COMPATIBILITY value to be at least n.

System action

The statement cannot be processed.

Programmer response

Either delay running your program until the application compatibility bind option (for static SQL) or special register CURRENT APPLICATION COMPATIBILITY (for dynamic SQL) has been set to at least the release that introduces this function, or restrict your program to functions that are allowed in a lower release only.

SQLSTATE

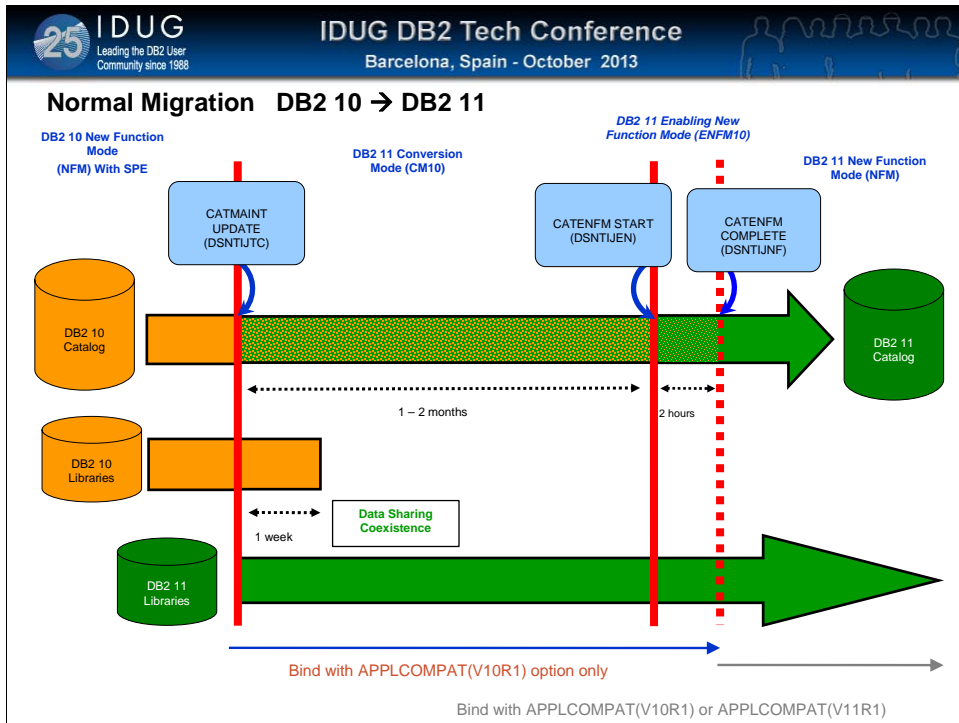
56038



Interaction with NFM

- NFM is intended to fence application **and** system behavior in a migration/co-existence environment
 - Ensure that the “n-m”* level of code can process objects created in the “n” level of the code
 - Ensure that applications bound on level “n” can run (after autobind) on level “n-m”*
- The APPLCOMPAT bind option is intended to ensure application behavior is consistent across releases of DB2
 - As such, APPLCOMPAT for the current release “n” may not be specified until NFM mode
 - Once in NFM MODE, either APPLCOMPAT(n) or APPLCOMPAT(n-1) may be specified (see the following charts). In a future release APPLCOMPAT(n-2) may be specified

* n-m where m = 1 or 2. DB2 10 is the lowest level that may be specified



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.

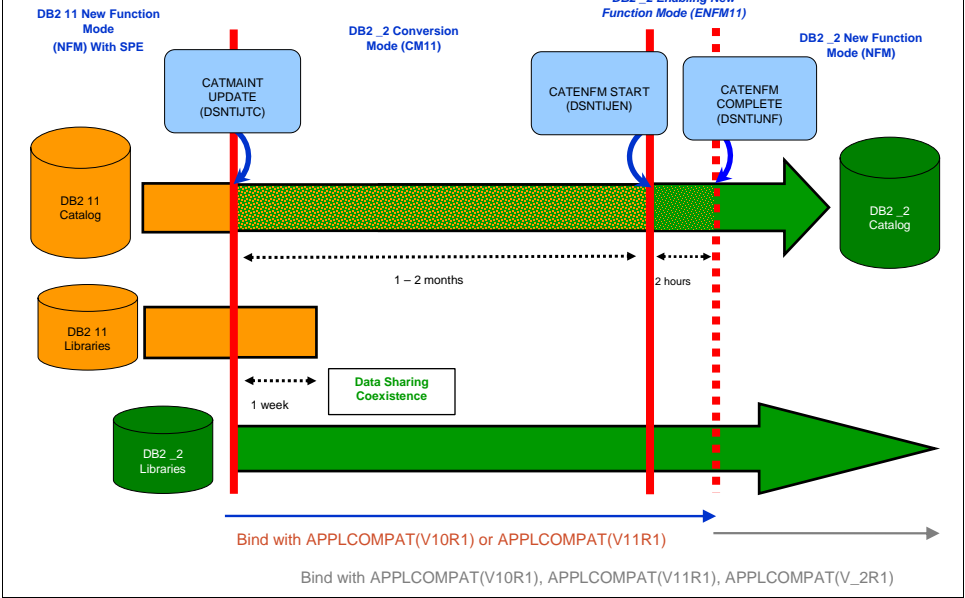
Some migration considerations are:

- 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.

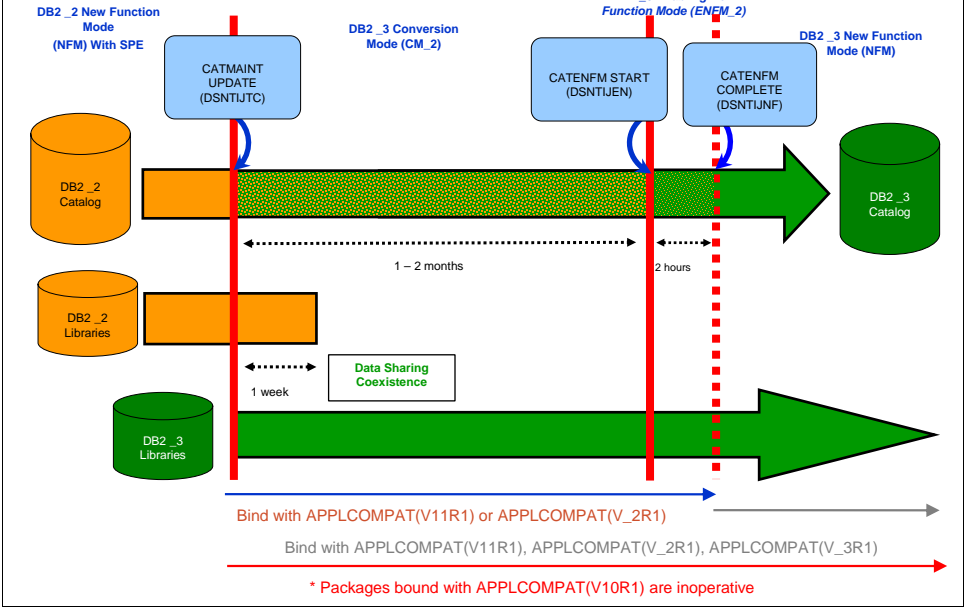
Note – this is not necessarily to scale!

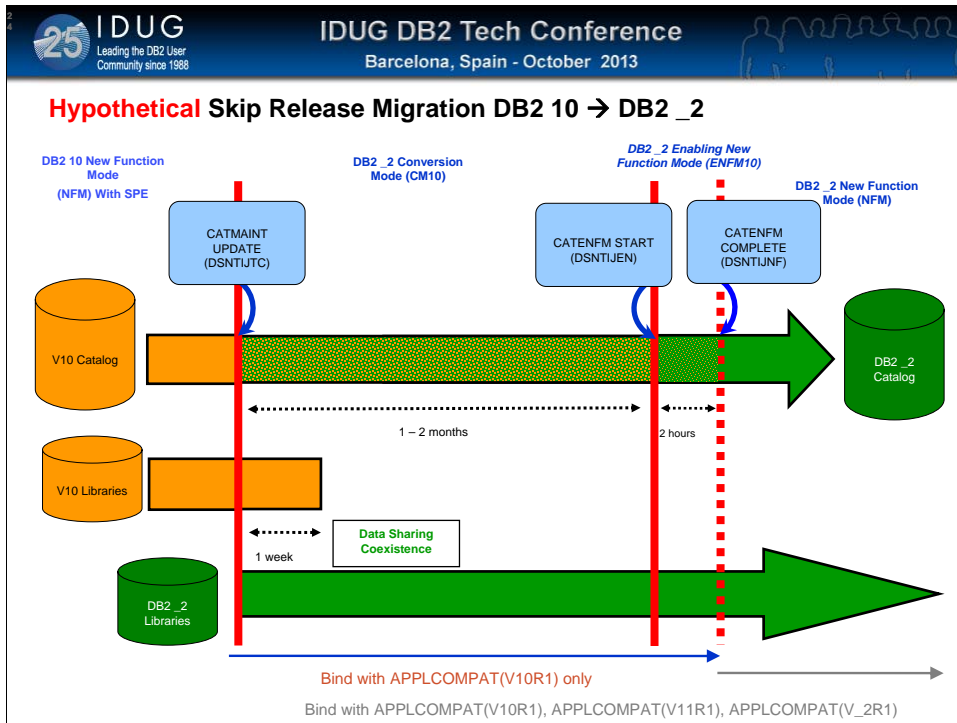
Note – ONE WAY – fallback to CM9* (covered later) is possible but not to DB2 9

Normal Migration DB2 11 → DB2_2



Normal Migration DB2_2 → DB2_3





Migration from V8 direct 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* (covered later) is possible, but not to DB2 V8



Interaction with NFM (cont)

- What about New DDL and Authorization syntax and semantics in an application bound with APPLCOMPAT(n-1)?
 - DDL and Authorization are fenced by NFM and are not affected by the APPLCOMPAT level
 - For example – You can use SPUFI - bound with APPLCOMPAT(V10R1)
 - “CREATE” a GLOBAL VARIABLE (new DB2 11 function) – “gv”
 - “GRANT” authority to a user to “SET” the GLOBAL VARIABLE – “gv”
 - “SET” “gv” fails because “SET” is DML



New Reserved Words

- Every release, DB2 adds new reserved words.
For example: DB2 11 added ARRAY_EXISTS (a new reserved word for a predicate).
- DB2 has made great strides in enhancing the parser to make new reserved words only be reserved in context. This reduces but does not eliminate impact to customers as they migrate from release to release.



New Reserved Words Example

In V10 and V11, these statements work fine because ARRAY_EXISTS is softened to identifier:

```
CREATE TABLE T1 (ARRAY_EXISTS INT); -- no syntax error
INSERT INTO T1 VALUES (11);

CREATE FUNCTION UDF2 () RETURNS INT
BEGIN
  DECLARE X INT;
  SELECT ARRAY_EXISTS INTO X FROM T1; -- no syntax error
  RETURN X * 2;
END!

SELECT UDF2() FROM SYSIBM.SYSDUMMY1! -- returns 22
```



New Reserved Words Example (cont)

However, the following works in V10 but the syntax in red fails in V11

```
CREATE FUNCTION ARRAY_EXISTS (PARM1 CHAR(3), PARM2 INT) -- no error in V10 or V11
  RETURNS CHAR(1)
  BEGIN
    RETURN SUBSTR(PARM1,PARM2,1);
  END!

COMMIT!

CREATE FUNCTION UDF1 () RETURNS INT
  BEGIN
    DECLARE X CHAR(3) DEFAULT 'ABC';
    IF ARRAY_EXISTS(X,1) = 'A' THEN -- OK in V10, -104 in V11
      RETURN 1;
    ELSE
      RETURN 2;
    END IF;
  END!

SELECT UDF1() FROM SYSIBM.SYSDUMMY1;
```



New Reserved Word Solution

- Each DB2 release has only one parser grammar
 - Using the APPLCOMPAT capability doesn't work in this case because to dual path the incompatible change would require two parsers (essentially the DB2 10 and DB2 11 grammar)
- To avoid the cost of shipping two parsers, we will add code (in an APAR) to DB2 10 to identify where DB2 11 keywords would cause a failure and issue IFCID 366 to alert customers of this issue.
 - Since new reserved words are only reserved in context, we expect customers will identify few or no issues, but the function allows customers to move forward with confidence.
- APAR PM84769/UK94459 closed in 2Q2013
 - Three words are checked, in context ARRAY_EXISTS, CUBE, and ROLLUP
- Other directions possible in the future

How to Identify Potential Issues





Solution Detail (cont)

- Accounting Summary Field (WPKTINCOMPAT)
- Trace records IFCID 366/376 will be provided for all incompatible code paths
 - The traces will be in the down-level code paths to illuminate the path until the down-level code is removed (in release N+3)
 - For example if we change the SQLCODE -123 to SQLCODE -456 in DB2 V11
 - New trace with function code yyyy will be cut in the paths where the old SQLCODE is being returned for packages bound with APPLCOMPAT(V10)
 - The trace will not be cut if APPLCOMPAT(V11R1), or above is specified
 - The trace will be cut in DB2 11 and DB2_2 in the down level path (APPLCOMPAT(V10R1)), but will be removed in DB2_3 when the compatibility code is removed



IFCID 366/376

IFCID 366/376 identify the SQL statements with potential incompatible changes when switching to the new V11R1 application behavior.

IFCID 366 is enhanced to identify any new type (TBD) of incompatible application changes so that customers can identify subject packages prior to switching to the new V11R1 application behavior.

IFCID 376 This trace record is written once for each unique dynamic cached statement and static statement if it was bound on V10 NFM or later. For static statements that are bound before V10 NFM, this trace record will still be externalized once per unique combo of plan, pkg ID, statement number.

IFCID 376

QW0376 DSECT

QW0376FN DS F The value '1' indicates that

- * the DB2 for z/OS Version 9
- * SYSIBM.CHAR(decimal-expr)
- * function has been executed.

*

- * The value '3' indicates
- * unsupported character string
- * representation of a timestamp

*

- * The value of '1101' indicates inserting into an XML column
- * without XMLDOCUMENT function

*

- * The value of '1102' indicates V10 Xpath evaluation was in effect

*

- * The value of '1103' indicates V10 RLF
- * reactive governing behavior for dynamic SQL was in effect

...

IFCID 376

```
QW0376SN DS F Statement number of the query
QW0376PL DS CL8 Plan name for this query
QW0376TS DS CL8 Timestamp for this query
QW0376SI DS CL8 Statement Identifier
QW0376TY DS XL2 Statement information
QW0376DY EQU X'8000' Statement is dynamic
QW0376SC EQU X'4000' Statement is static
QW0376SE DS H Section number
QW0376PC_Off DS H Offset from QW0366 to Package
* Collection ID
QW0376PN_Off DS H Offset from QW0366 to
* Program name
QW0376VL DS H Version length
QW0376VN DS CL64 Version
*
QW0376PC_D DSECT
QW0376PC_Len DS H Length of Package Collection ID
QW0376PC_Var DS 0CL128 %U Package Collection ID
*
QW0376PN_D DSECT
QW0376PN_Len DS H Length of Program Name
QW0376PN_Var DS 0CL128 %U Program Name
```

Updated IFCID 106

IFCID 106 is updated to trace the new ZPARAM

QWP4RMDB DS CL8 REORG_MAPPING_DATABASE N0239R3

QWP4DM1636 DS CL8 (s) DM1636

QWP4MIMTS DS F MAXSORT_IN_MEMORY N4504r5

QWP4MUSE DS XL2 (s) N4504r5

QWP4IXCU DS H INDEXCLEANUP_THREADS n0010r5

QWP4DEGD DS F PARAMDEG_DPSI n231r5

QWP4APCO DS 0H APPLCOMPAT n8195r5

QWP4APCO_Len DS H length of APPLCOMPAT setting

QWP4APCO_Var DS CL10 APPLCOMPAT setting

DS CL32 UNUSED LI872

The IFCID 106 formatter stored procedures, SYSPROC.DSNWZP and SYSPROC.ADMIN_INFO_SYSPARM are updated to report the APPL_COMPAT setting.



On Migration

- Premigration jobs will flag as a warning any packages bound with APPLCOMPAT(V10R1) or APPLCOMPAT() on migration to DB2 _2
- Premigration jobs will flag as a serious-warning (error) any packages bound with APPLCOMPAT(V10R1) or APPLCOMPAT() on migration to DB2 _3
 - These packages will be **NOT** be autobound in DB2 _3
 - These packages will be inoperative in DB2 _3



What isn't addressed

- Changes to SQL function because of customer reported issues PMRs/APARs
 - Customer initiated APARs can result in differences in SQL behavior. While these changes almost always correct a defect in the product, the changed behavior can affect results that other customers see.

Summary and Questions



Summary

- With the APPLCOMPAT bind option, we are striving to separate release migration from application changes that may be necessary to migrate to a new release.
- DB2 is providing two releases of backward compatible behavior to enable customers to migrate faster to a new release with confidence that applications will be have the same in the new release as they have in the previous releases.
- Customers can use the tracing capabilities that have been provided to identify applications that potentially need to change.

Chris Crone

Session A15
cjc@us.ibm.com





Disclaimer

© Copyright IBM Corporation 2011. All rights reserved.
U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule
Contract with IBM Corp.

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, 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 ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS AND/OR SOFTWARE.

IBM, the IBM logo, ibm.com, and DB2 for z/OS are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml