IFCID 366 APPLICATION COMPATIBILITY V11R1

DB2 V11

June 2017

Agenda

- IFCID 366 Introduction
- System and Configuration
  - System State before DB2 v11R1
  - Starting I366 trace
- Externalize I366 and SQL text
  - EXPLAIN_STMT_CACHE/Unload SYSPACKSTMT
  - I366 TRACE FROM SMF
- Reporting and Analysis (with IDAA)
- Application Testing and Implementation
- Be Aware….
- Questions

IFCID 366 Introduction

- IFCID 366 is a record trace that identifies V10 incompatible functions.
- IFCID 366 was first introduced in DB2 V10 as a method to identify (CAST) CHAR,(CAST) VARCHAR and TIMESTAMP application incompatibilities.
- DB2 V11 enhanced I366 with additional tracing capabilities.
- DB2 V11 introduces a second IFCID, IFCID 376. IFCID 376 is a roll up of 366. DB2 writes one record for each unique static or dynamic statement
- I366 Function Code Incompatibilities are explained on IBM Knowledge Center https://www.ibm.com/support/knowledgecenter/SSEPEK_11.0.0/wnew/src/tpc/db2z_whatschanged.html

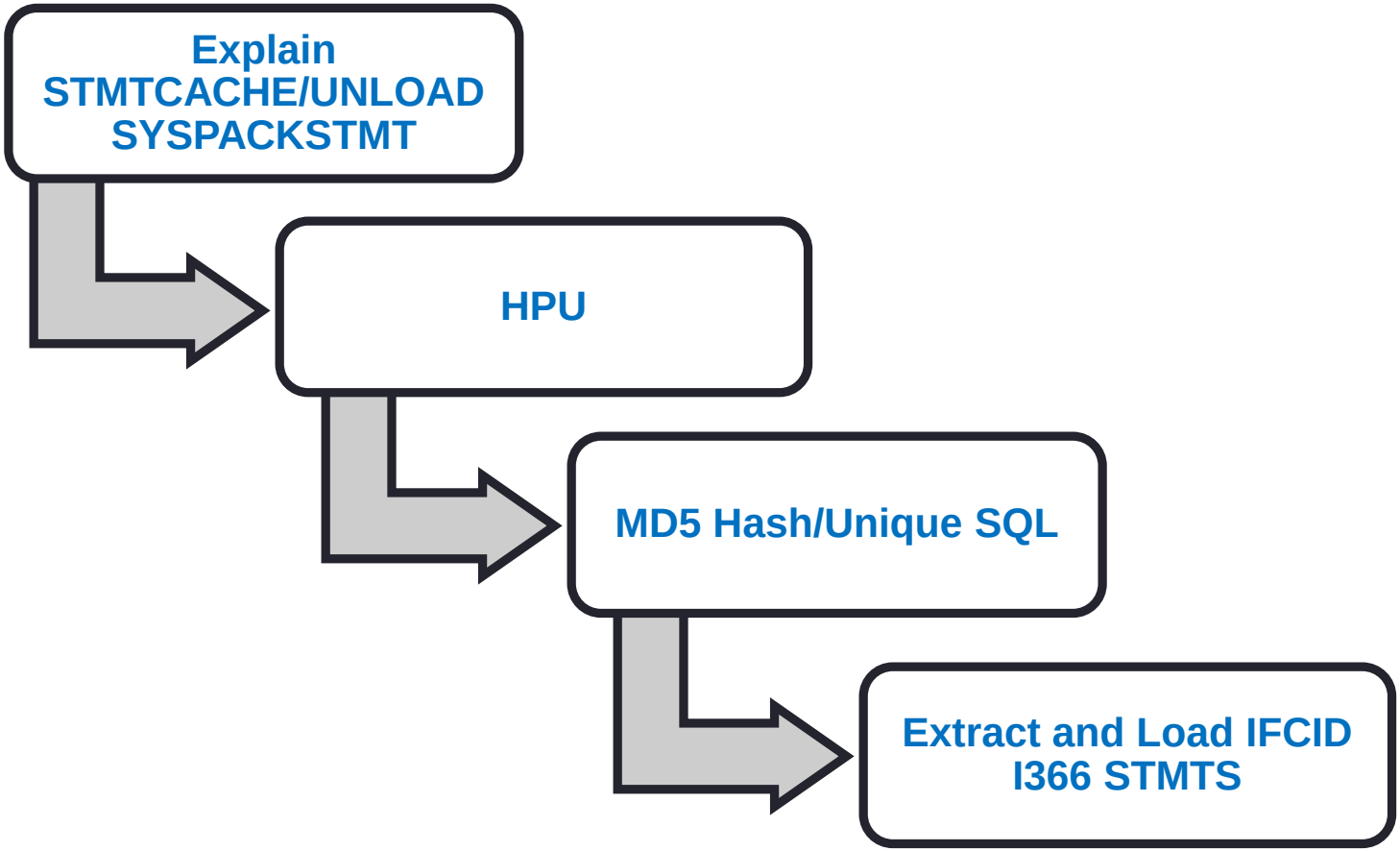
System and Configuration

- System State
  - 2 Production Systems and 4 Non-Production Systems
  - DB2 v10 with BIF_COMPATABILITY = V9_DECIMAL_VARCHAR
  - “The BIF_COMPATIBILITY subsystem parameter specifies whether built-in functions and specifications are to return results in the newer format or revert to the pre-DB2® 10 format.”
  - https://www.ibm.com/support/knowledgecenter/en/SSEPEK_11.0.0/inst/src/tpc/db2z_ipf_bifcompatibility.html

- Start Command IFCID 366
  - -START TRACE(PERFM) CLASS(32) IFCID(366) SCOPE (GROUP) DEST(SMF)
  - Reminder to start trace after system cycle, especially data sharing scope: Group

Overall Process Flow

- Tables with I366 trace and SQL text of Dynamic and Static Statements



EXPLAIN STMTCACHE/Unload SYSPACKSTMT

- "EXPLAIN STMTCACHE ALL" for every DB2 member
- Unload SYSPACKSTMT – Load to STATIC_STG table
- Executed once per day at midnight
- Execute HPU Unload of DSN_STATEMENT_CACHE _TABLE to flat file using SELECT statement
  - CAST(SUBSTR(STMT_TEXT,1,8000) AS VARCHAR(8000))
  - IDAA does not handle LOB fields
- For each STMTCACHE file, LOAD to STMT_CACHE_STG table
- For SYSPACKGE files - Load to STATIC_STG table

HPU

- HPU Unload from STMT_CACHE_STG table where the row doesn't already exist on our STMT_CACHE repository table (identifying new records only):

```
UNLOAD TABLESPACE
SELECT
    STG.STMT_ID
   ,COALESCE(STG.STMT_TOKEN,'NODS')
   ,STG.COLLID
   ,STG.PROGRAM_NAME
   ,STG.INV_DROPALT
  ,STG.STMT_TEXT
.
.
FROM STMT_CACHE_STG  STG
WHERE NOT EXISTS
(
SELECT *
FROM DB2ADMN.TDB_STMT_CACHE A
WHERE STG.PROGRAM_NAME       = A.PROGRAM_NAME
AND   STG.CACHED_TS          = A.CACHED_TS
AND   STG.PRIMAUTH           = A.PRIMAUTH
AND   STG.SCHEMA             = A.SCHEMA
AND   STG.GROUP_MEMBER       = A.GROUP_MEMBER
)
WITH UR;
```

HPU

- **HPU Unload from STATIC_STG table where the row doesn't already exist on our STATIC_SQL repository table (identifying new records only):**

```
WITH PKG_STAT (COLLECTION, PROGRAM, PKG_TIMESTAMP )
AS (
SELECT COLLECTION, PROGRAM, MAX(PKG_TIMESTAMP)
FROM DB2ADMN.CQM31_53_STATIC_STG_SQL
GROUP BY COLLECTION, PROGRAM
MINUS
SELECT SPKG.COLLECTION, SPKG.PROGRAM,
MAX(SPKG.PKG_TIMESTAMP)
FROM DB2ADMN.CQM31_53_STATIC_SQL SPKG
WHERE…

SELECT
.
.
,CAST(SUBSTR(SQLSTMT,1,8000) AS VARCHAR(8000))
FROM DB2ADMN.CQM31_53_STATIC_STG_SQL STGSQL, PKG_STAT
WHERE STGSQL.COLLECTION = PKG_STAT.COLLECTION
AND STGSQL.PROGRAM = PKG_STAT.PROGRAM
AND STGSQL.PKG_TIMESTAMP = PKG_STAT.PKG_TIMESTAMP
```

MD5 Hash/Unique SQL

- Execute Rexx exec that call a stored procedure
- Reads HPU unloaded file
- Calls Stored Procedure to generate an MD5 hash value for each unique SQL Statement
- Inserts data into two repository tables: STMT_CACHE table and STMT_CACHE_TEXT table
- STMT_CACHE table has all columns except STMT_TEXT
- STMT_CACHE_TEXT has only MD5 hash value and STMT_TEXT
- If we receive -803, we know the SQL Statement is already on our table.

Extract and Load IFCID 366 Records

- Dump TYPE(102) Records from SMF
- Execute SAS program to extract relevant IFCID366 data using VMAC102 and _T102366 macro
- Mapping for IFCID366 records can be found in SDSNMACS(DSNDQW05)

Extract and Load IFCID 366 Records

- IF QWHCATYP = 1 THEN F366TYP   = 'TSO ';
- IF QWHCATYP = 2 THEN F366TYP   = 'DB2 CALL ATTACH';
- IF QWHCATYP = B THEN F366TYP   = 'DB2 UTILITIES';
- IF QWHCATYP = C THEN F366TYP   = 'RRSAF';
- IF QWHCATYP = 8 THEN F366TYP   = 'DRDA ';
- IF QW0366FN = 1 THEN
- F366DESC = 'V9 SYSIBM.CHAR(DECIMAL-EXPR) FUNCTION';
- IF QW0366FN = 2 THEN
- F366DESC = 'V9 SYSIBM.VARCHAR(DECIMAL-EXPR)
- FUNCTION CAST (DECIMAL AS CHAR)';
- IF QW0366FN = 3 THEN F366DESC   = 'UNSUPPORTED TIMESTAMP VALUE';
- IF QW0366FN = 4 THEN F366DESC   = 'RESERVED WORD:ARRAY_EXISTS';
- IF QW0366FN = 5 THEN F366DESC   = 'RESERVED WORD:CUBE';
- IF QW0366FN = 6 THEN F366DESC   = 'RESERVED WORD:ROLLUP';
- IF QW0366FN = 7 THEN F366DESC   = 'POSSIBLE SQLCODE -301';

Extract and Load IFCID 366 Records

- IF QW0366FN = 8 THEN F366DESC   = 'DDF_COMPATIBILITY/JAVA EXCP';
- IF QW0366FN = 9 THEN F366DESC   = 'TIMEZONE IGNORE';
- IF QW0366FN = 1101 THEN F366DESC = 'XML NODE';
- IF QW0366FN = 1102 THEN F366DESC = 'XPATH PROCESSING';
- IF QW0366FN = 1103 THEN F366DESC = 'ASUTIME';
- IF QW0366FN = 1104 THEN F366DESC = 'CLIENT_USERID';
- IF QW0366FN = 1105 THEN F366DESC = 'CLIENT WKSTNAME';
- IF QW0366FN = 1106 THEN F366DESC = 'CLIENT_APPLNAME';
- IF QW0366FN = 1107 THEN F366DESC = 'CLIENT_ACCTNG';
- IF QW0366FN = 1108 THEN F366DESC = 'CAST(STRING AS TIMESTAMP)';
- IF QW0366FN = 1109 THEN F366DESC = 'CAST(STRING AS TIMESTAMP)';
- IF QW0366FN = 1110 THEN F366DESC = 'BUILT-IN FUNCTION: SPACE';
- IF QW0366FN = 1111 THEN F366DESC = 'BUILT-IN FUNCTION: VARCHAR';

- IF QW0366TY = '4000'X THEN F366TYPE = 'STATIC';
- IF QW0366TY = '8000'X THEN F366TYPE = 'DYNAMIC';

Extract and Load IFCID 366 Records

Fields Extracted from IFICD 366 and Loaded to IFICD 366 table:

·QWHAMEMN	- DB2 Member Name	·F366TYPE	- ‘STATIC’ or ‘DYNAMIC’ based on QH0366TY
·MYDATE	- SMF Date	·QW0366FN	- FunctionCode
·QWHCAID	- Authid	·F366DESC	- Text description of function code
·QWHCCN	- Connection Name	·QW0366SN	- Statement number of query
·QWHCEUWN	- End user Workstation Name	·QW0366SE	- Section number
·F366TYP	- Type Code	·QW0366SI	- Statement Identifier
·QWHCCV	- Correlation ID Value	·QW0366TS	- Timestamp for query
·QWHSSTCK	- StoreClock	·QW0366VL	- Version Length
·QWHSIID	- Subsystem Name	·QW0366VN	- Version name
·QWHDRQNM	- Requestor location Name		
·QWHCTOKN	- Accounting Token		
·QW0366PC	- Package collection		
·QW0366PN	- Program name		
·QW0366PL	- Plan Name for query		

Reporting and Analysis

- Capturing and externalizing IFCID 366 and the corresponding SQL statements allowed us to build reports by function codes, DB2 member, PRIMAUTH, date and application name easily using IDAA.

MEMBER	PRIMAUTH	CORRELATION_ID	COLLECTION_ID	PROGRAM_NAME	PLAN_NAME	STMT_TYPE	FUNCTION_CODE	FUNCTION_DESC	D_STMT_ID	SMF_DATE	APPLICATIONID
DWP1	P9DWD318	P9DWD318	SYSACCEL	AQTDBCON	AQTSCALL	STATIC	1107	CLIENT_ACCTNG	62310	08/18/16	DW
DWP1	P9DWD011	P9DWD011	SYSACCEL	AQTDBCON	AQTSCALL	STATIC	1104	CLIENT_USERID	62310	08/30/16	DW
DWP1	P9DWDA64	P9DWDA64	DSNTIB10	DSNTIAUL	DSNTIAUL	DYNAMIC	1	V9 SYSIBM.CHAR(DECIMAL-EXPR)...	1143112	08/20/16	DW
DWP1	P9DWDD55	P9DWDD55	SYSACCEL	AQTDBCON	AQTSCALL	STATIC	1106	CLIENT_APPLNAME	62310	08/24/16	DW
DWP1	P9DWD116	P9DWD116	SYSACCEL	AQTDBCON	AQTSCALL	STATIC	1105	CLIENT WKSTNAME	62310	08/13/16	DW
DBP1	E930818	E930818	ADBL	ADB27SP	ADB	STATIC	1	V9 SYSIBM.CHAR(DECIMAL-EXPR)...	361732	08/23/16	Person
DWP1	P9DWD321	P9DWD321	SYSACCEL	AQTDBCON	AQTSCALL	STATIC	1104	CLIENT_USERID	62310	08/13/16	DW
DBP1	DIRECTDB	db2jcc_appli	NULLID	SYSLN200	DISTSERV	DYNAMIC	2	V9 SYSIBM.VARCHAR(DECIMAL-EXPR)	362686	08/16/16	FTD
DWP1	P9DWDA28	P9DWDA28	SYSACCEL	AQTDBCON	AQTSCALL	STATIC	1106	CLIENT_APPLNAME	62310	08/03/16	DW
DWP1	P9DWDY10	P9DWDY10	SYSACCEL	AQTDBCON	AQTSCALL	STATIC	1107	CLIENT_ACCTNG	62310	08/15/16	DW
DBP4	DIRECTDB	db2jcc_appli	NULLID	SYSLN200	DISTSERV	DYNAMIC	2	V9 SYSIBM.VARCHAR(DECIMAL-EXPR)	1518128	08/17/16	FTD
DWP1	P9DWDW32	P9DWDW32	SYSACCEL	AQTDBCON	AQTSCALL	STATIC	1105	CLIENT WKSTNAME	62310	08/21/16	DW
DWP1	P9DWDU75	P9DWDU75	SYSACCEL	AQTDBCON	AQTSCALL	STATIC	1106	CLIENT_APPLNAME	62310	08/23/16	DW
DWP1	P9DWDXA6	P9DWDXA6	SYSACCEL	AQTDBCON	AQTSCALL	STATIC	1107	CLIENT_ACCTNG	62310	08/23/16	DW
DBD2	FCTUWAS9	db2jcc_appli	NULLID	SYSLH200	DISTSERV	DYNAMIC	1106	CLIENT_APPLNAME	199890	08/09/16	CT

Reporting and Analysis: SQL

```
SELECT I366.MEMBER,
        I366.PRIMAUTH,
        I366.CORRELATION_ID,
        I366.COLLECTION_ID,
        I366.PROGRAM_NAME,
        I366.PLAN_NAME,
        I366.STMT_TYPE,
        I366.FUNCTION_CODE,
        I366.FUNCTION_DESC,
        I366.D_STMT_ID,
        i366.smf_date,
        CASE
            WHEN PRIMAUTH LIKE 'DIRECT%' THEN 'FTD'
            WHEN SUBSTR(primauth,1,1) = 'E' THEN 'Person'
            WHEN SUBSTR(primauth,1,1) = 'T' THEN 'Person'
            WHEN SUBSTR(primauth,1,1) = 'S' THEN 'Person'
            WHEN SUBSTR(primauth,1,1) = 'P' THEN SUBSTR(PRIMAUTH,3,2)
            ELSE SUBSTR(PRIMAUTH,2,2)
        END AS ApplicationID
FROM     DB2ADMN.TDB_SMF_IFCID366 I366
WHERE
--I366.STMT_TYPE = 'DYNAMIC'
--AND
year(timestamp_format(I366.SMF_DATE, 'mm/dd/yy')) = '2016'
AND      month(timestamp_format(I366.SMF_DATE, 'mm/dd/yy')) = '$month'
--AND I366.MEMBER LIKE 'DBT%'
--AND I366.FUNCTION_CODE IN (1,2,3,4,5,6,7,8)
--AND I366.PRIMAUTH LIKE 'FDWP%'
GROUP BY I366.MEMBER, I366.PRIMAUTH, I366.CORRELATION_ID, I366.COLLECTION_ID,
   I366.PROGRAM_NAME, I366.PLAN_NAME, I366.STMT_TYPE, I366.FUNCTION_CODE,
   I366.FUNCTION_DESC, I366.D_STMT_ID, I366.SMF_DATE;
```

Reporting and Analysis: STMT_TEXT

- Analysis required reviewing STMT_TEXT and searching for key items as CASTs (using regular expressions search)

TEXT_HASH	STMT_TYPE	FUNCTION_CODE	FUNCTION_DESC	STMT_TEXT	APPLICATIONID
', ·ç0t7SçJÝ	DYNAMIC	2	V9 SYSIBM.VARCHAR(DECIMAL-EXPR)	SELECT STG.I_REC_KEY, STG.I_ORG_UNIT_KEY, STG.I_GL_ACCT,...	Person
ûÁŒD~|çPò@Œæ...	DYNAMIC	1107	CLIENT_ACCTNG	select CURRENT CLIENT_ACCTNG, CURRENT CLIENT_APPLNAME, CURRENT...	RT
6¡[gBf\üòM	DYNAMIC	1104	CLIENT_USERID	select SESSION TIME ZONE, CURRENT CLIENT_ACCTNG, CURRENT...	RT
6¡[gBf\üòM	DYNAMIC	1104	CLIENT_USERID	select SESSION TIME ZONE, CURRENT CLIENT_ACCTNG, CURRENT...	UP
¡iÀL§ì}ÎAbõäâä	DYNAMIC	2	V9 SYSIBM.VARCHAR(DECIMAL-EXPR)	SELECT I_CHK_ITEM_ID, CAST (I_CHK_ITEM_SUPER_MICR_IMG_KEY...	PA
6¡[gBf\üòM	DYNAMIC	1104	CLIENT_USERID	select SESSION TIME ZONE, CURRENT CLIENT_ACCTNG, CURRENT...	MV
Ðih¬ã§RÎÎ.à	DYNAMIC	2	V9 SYSIBM.VARCHAR(DECIMAL-EXPR)	select this_.i_approval as i2_25_2_, this_.i_company as i3_25_2_, this_.z_expiry as...	FTD
!ah:Fµa¡Éþðlç	DYNAMIC	1	V9 SYSIBM.CHAR(DECIMAL-EXPR) FUNCTION	SELECT SUBSTR('DTL-FINACCT ',1,15)..	DW
ÔÏU=l9çdÁïn·	DYNAMIC	2	V9 SYSIBM.VARCHAR(DECIMAL-EXPR)	SELECT FN.TFN_OV_FCT_DISPSTN.OV_FCT_DISPSTN_ID,...	VA
6¡[gBf\üòM	DYNAMIC	1104	CLIENT_USERID	select SESSION TIME ZONE, CURRENT CLIENT_ACCTNG, CURRENT...	TP
>:[gòMÐ¼ãŠ¯J	DYNAMIC	2	V9 SYSIBM.VARCHAR(DECIMAL-EXPR)	Select frstr ,tostr ,TIMESTAMPDIFF(16, CAST((TIMESTAMP(TOSTR||'-11:59:00') -...	Person
³ƒ\²>{}	DYNAMIC	2	V9 SYSIBM.VARCHAR(DECIMAL-EXPR)	SELECT FPV.C_STK, FPV.C_VIN, FPV.I_CL_ID, FPV.I_UN_ID , FPV.C_MDL,...	FTD
‡òšúl, ïë9=w	DYNAMIC	3	V9 UNSUPPORTED CHARACTER STRING REPRE	WITH STAGE001 AS(SELECT...	Person
6¡[gBf\üòM	DYNAMIC	1107	CLIENT_ACCTNG	select SESSION TIME ZONE, CURRENT CLIENT_ACCTNG, CURRENT...	BP
6¡[gBf\üòM	DYNAMIC	1104	CLIENT_USERID	select SESSION TIME ZONE, CURRENT CLIENT_ACCTNG, CURRENT...	PA
A\mvÖ]3öÿÆ3®6	DYNAMIC	2	V9 SYSIBM.VARCHAR(DECIMAL-EXPR)	SELECT I_USER_ID, I_BRAND , VARCHAR_FORMAT...	Person
sF°[;êgöl¶l÷'	DYNAMIC	1	V9 SYSIBM.CHAR(DECIMAL-EXPR) FUNCTION	select company0_.i_company as i1_42_5_, company0_.x_active as x2_42_5_,...	FTD
ê3Åàn¡œ«wÖ›²þë	DYNAMIC	2	V9 SYSIBM.VARCHAR(DECIMAL-EXPR)	SELECT BT.LOLOC00.C00_GROUP,...	BT
³ŸÅtnŠ“š<b¿½	DYNAMIC	3	V9 UNSUPPORTED CHARACTER STRING REPRE	Select TIMESTAMPDIFF(16, CAST((TIMESTAMP(TOSTR||'-11:59:00') -...	Person

Reporting and Analysis: SQLSTMT

- Static Statements required manual review of SQLs in Packages

STMT_TYPE	FUNCTION_	FUNCTION_DESC	SQLSTMT	QUERY_STMT_NBR	COLLECTION_ID	PROGRAM_NAME	PLAN_NAME	PRIMA_	APPLI_
STATIC	1104	CLIENT_USERID	SELECT CURRENT CLIENT_ACCTNG , CURRENT CLIENT_APPLNAME , CURRENT…	312	SYSACCEL	AQTDBCON	AQTSCALL	P9DWD…	DW
STATIC	1107	CLIENT_ACCTNG	SELECT CURRENT CLIENT_ACCTNG , CURRENT CLIENT_APPLNAME , CURRENT…	312	SYSACCEL	AQTDBCON	AQTSCALL	P9DWD…	DW
STATIC	1106	CLIENT_APPLNAME	INSERT INTO ACP1.TAC_FUNDTSF3_SECUR_AUD …	2	ACP1	RAC023D1	DISTSERV	FACPA…	AC
STATIC	1	V9 SYSIBM.CHAR(DECIMAL-EXPR)…	SELECT SUBSTR (CHAR (XL40_CRR_AS_OF_DAT) , 9 , 2) INTO : H FROM…	3214	CBT001B	BTPREDBT	CBT001B	P0BTD…	BT
STATIC	1107	CLIENT_ACCTNG	SELECT CURRENT CLIENT_ACCTNG , CURRENT CLIENT_APPLNAME , CURRENT…	308	SYSACCEL	AQTDBCON	AQTSCALL	P9DWM…	DW
STATIC	1104	CLIENT_USERID	SELECT CURRENT CLIENT_ACCTNG , CURRENT CLIENT_APPLNAME , CURRENT…	308	SYSACCEL	AQTDBCON	AQTSCALL	P9DWD…	DW
STATIC	1105	CLIENT WKSTNAME	SELECT CURRENT CLIENT_ACCTNG , CURRENT CLIENT_APPLNAME , CURRENT…	312	SYSACCEL	AQTDBCON	AQTSCALL	T94993A	Person
STATIC	2	V9 SYSIBM.VARCHAR(DECIMAL-EXPR)	OPEN CURSOR1	1657	CBT001B	BTCUSLIB	CBT001B	P0BTD…	BT
STATIC	2	V9 SYSIBM.VARCHAR(DECIMAL-EXPR)	INSERT INTO DB . TDB_DB2_UPTIME VALUES (: H : H , 'OUTAGE' , TIMESTAMP (:…	3746	CDB001D	PDBDB2UT	DSNREXX	P0DBW…	DB
STATIC	1106	CLIENT_APPLNAME	SELECT CURRENT CLIENT_ACCTNG , CURRENT CLIENT_APPLNAME , CURRENT…	308	SYSACCEL	AQTDBCON	DISTSERV	T76531A	Person
STATIC	1	V9 SYSIBM.CHAR(DECIMAL-EXPR)…	SELECT SEQ . SCHEMA , SEQ . OWNER , SEQ . NAME , SEQ . SEQTYPE , SEQ .…	12931	ADBL	ADB2REM	ADB	E930818	Person

```
------------------------------------------------------------
--*. DETAILS DYNAMIC;
------------------------------------------------------------

WITH QM_CTE (
MEMBER,PRIMAUTH, CORRELATION_ID, COLLECTION_ID, PROGRAM_NAME, PLAN_NAME
,STMT_TYPE, FUNCTION_CODE, FUNCTION_DESC, D_STMT_ID,d_I366)
AS
(SELECT  I366.MEMBER, I366.PRIMAUTH, I366.CORRELATION_ID, I366.COLLECTION_ID, I366.PROGRAM_NAME, I366.PLAN_NAME
,I366.STMT_TYPE, I366.FUNCTION_CODE, I366.FUNCTION_DESC, I366.D_STMT_ID, i366.smf_date
FROM DB2ADMN.TDB_SMF_IFCID366 I366
WHERE
I366.STMT_TYPE = 'DYNAMIC'

and year(timestamp_format(I366.SMF_DATE, 'mm/dd/yy')) = '2016'
and month(timestamp_format(I366.SMF_DATE, 'mm/dd/yy')) = '8'
--and (I366.member like 'DRP%' OR I366.member like 'DRU%' )
GROUP BY
 I366.MEMBER
,I366.PRIMAUTH,I366.CORRELATION_ID,I366.COLLECTION_ID,I366.PROGRAM_NAME
,I366.PLAN_NAME,I366.STMT_TYPE,I366.FUNCTION_CODE,I366.FUNCTION_DESC
,I366.D_STMT_ID,I366.SMF_DATE)
, STMT as (SELECT
 QM_CTE.MEMBER
,QM_CTE.PRIMAUTH ,QM_CTE.CORRELATION_ID ,QM_CTE.COLLECTION_ID ,QM_CTE.PROGRAM_NAME ,QM_CTE.PLAN_NAME
,QM_CTE.STMT_TYPE ,QM_CTE.FUNCTION_CODE ,QM_CTE.FUNCTION_DESC ,QM_CTE.D_STMT_ID
, TEXT.STMT_TOKEN, TEXT.TEXT_HASH
, TEXT1.STMT_TEXT
FROM QM_CTE
LEFT JOIN db2admn.TDB_STMT_CACHE  TEXT
ON
QM_CTE.program_name = text.program_NAME
AND QM_CTE.PRIMAUTH = TEXT.PRIMAUTH
AND QM_CTE.MEMBER = TEXT.GROUP_MEMBER
AND QM_CTE.D_STMT_ID  = text.stmt_ID
JOIN
db2admn.TDB_STMT_CACHE_TEXT TEXT1
ON
text.TEXT_HASH = text1.TEXT_HASH
WHERE
TEXT.CACHED_TS = (SELECT MAX(TEXT2.CACHED_TS) FROM
db2admn.TDB_STMT_CACHE  TEXT2
WHERE
QM_CTE.program_name = text2.program_NAME
AND QM_CTE.PRIMAUTH = TEXT2.PRIMAUTH
AND QM_CTE.MEMBER = TEXT2.GROUP_MEMBER
AND QM_CTE.D_STMT_ID  = text2.stmt_ID
AND TEXT2.CACHED_TS <= timestamp_format(d_i366,'mm/dd/yy') + 1 day
))
select
stmt.text_hash,
stmt.STMT_TYPE ,stmt.FUNCTION_CODE ,stmt.FUNCTION_DESC ,
stmt.stmt_text,


CASE
WHEN PRIMAUTH LIKE 'DIRECT%' THEN 'FTD'
WHEN SUBSTR(stmt.primauth,1,1) = 'E' THEN 'Person'
WHEN SUBSTR(stmt.primauth,1,1) = 'T' THEN 'Person'
WHEN SUBSTR(stmt.primauth,1,1) = 'S' THEN 'Person'
WHEN SUBSTR(stmt.primauth,1,1) = 'P' THEN SUBSTR(STMT.PRIMAUTH,3,2)
ELSE SUBSTR(STMT.PRIMAUTH,2,2)
END AS ApplicationID
from stmt
GROUP BY

stmt.text_hash,
stmt.STMT_TYPE ,stmt.FUNCTION_CODE ,stmt.FUNCTION_DESC ,
stmt.stmt_text,
CASE
WHEN PRIMAUTH LIKE 'DIRECT%' THEN 'FTD'
```

```
---------------------------------------------------------------
--*. DETAILS STATIC;
---------------------------------------------------------------


WITH QM_CTE (
MEMBER,PRIMAUTH, CORRELATION_ID, COLLECTION_ID, PROGRAM_NAME, PLAN_NAME
,STMT_TYPE, FUNCTION_CODE, FUNCTION_DESC, D_STMT_ID,d_I366,QUERY_STMT_NBR)
AS
(SELECT  I366.MEMBER, I366.PRIMAUTH, I366.CORRELATION_ID, I366.COLLECTION_ID, I366.PROGRAM_NAME, I366.PLAN_NAME
,I366.STMT_TYPE, I366.FUNCTION_CODE, I366.FUNCTION_DESC, I366.D_STMT_ID, i366.smf_date, I366.QUERY_STMT_NBR
FROM DB2ADMN.TDB_SMF_IFCID366 I366
WHERE
I366.STMT_TYPE = 'STATIC'

and year(timestamp_format(I366.SMF_DATE, 'mm/dd/yy')) = '2016'
and month(timestamp_format(I366.SMF_DATE, 'mm/dd/yy')) = '8'
--and (I366.member like 'DRP%' OR I366.member like 'DRU%' OR I366.member like 'DDT%' OR I366.member like 'DDD%' )
GROUP BY
 I366.MEMBER
,I366.PRIMAUTH,I366.CORRELATION_ID,I366.COLLECTION_ID,I366.PROGRAM_NAME
,I366.PLAN_NAME,I366.STMT_TYPE,I366.FUNCTION_CODE,I366.FUNCTION_DESC
,I366.D_STMT_ID,I366.SMF_DATE, I366.QUERY_STMT_NBR)
, STMT as (SELECT
 QM_CTE.MEMBER
,QM_CTE.PRIMAUTH ,QM_CTE.CORRELATION_ID ,QM_CTE.COLLECTION_ID ,QM_CTE.PROGRAM_NAME ,QM_CTE.PLAN_NAME
,QM_CTE.STMT_TYPE ,QM_CTE.FUNCTION_CODE ,QM_CTE.FUNCTION_DESC ,QM_CTE.D_STMT_ID
, TEXT.TEXT_HASH
, TEXT1.SQLSTMT
FROM QM_CTE
LEFT JOIN DB2ADMN.CQM31_53_STATIC_SQL  TEXT
ON
QM_CTE.program_name = text.program
AND QM_CTE.QUERY_STMT_NBR = TEXT.STMT
AND case
when QM_CTE.MEMBER like 'DBP%' THEN 'PDG1'
when QM_CTE.MEMBER like 'DRP%' THEN 'PRG1'
when QM_CTE.MEMBER like 'DWP%' THEN 'PWG1'
ELSE '' END
= TEXT.DB2_SUBSYSTEM
AND QM_CTE.COLLECTION_ID  = text.COLLECTION
JOIN
DB2ADMN.CQM31_53_STATIC_SQL_STMT TEXT1
ON
text.TEXT_HASH = text1.TEXT_HASH
WHERE
TEXT.PKG_TIMESTAMP = (SELECT MAX(TEXT2.PKG_TIMESTAMP) FROM
DB2ADMN.CQM31_53_STATIC_SQL  TEXT2
WHERE
QM_CTE.program_name = text2.program
AND QM_CTE.QUERY_STMT_NBR = TEXT2.STMT
AND case
when QM_CTE.MEMBER like 'DBP%' THEN 'PDG1'
when QM_CTE.MEMBER like 'DRP%' THEN 'PRG1'
when QM_CTE.MEMBER like 'DWP%' THEN 'PWG1'
ELSE '' END
= TEXT.DB2_SUBSYSTEM
AND QM_CTE.COLLECTION_ID  = text2.COLLECTION
AND TEXT2.PKG_TIMESTAMP <= timestamp_format(d_i366,'mm/dd/yy') + 1 day
))
select
 stmt.text_hash,
stmt.STMT_TYPE ,stmt.FUNCTION_CODE ,stmt.FUNCTION_DESC ,
STMT.SQLSTMT,


CASE
WHEN PRIMAUTH LIKE 'DIRECT%' THEN 'FTD'
WHEN SUBSTR(stmt.primauth,1,1) = 'E' THEN 'Person'
WHEN SUBSTR(stmt.primauth,1,1) = 'T' THEN 'Person'
WHEN SUBSTR(stmt.primauth,1,1) = 'S' THEN 'Person'
```

Reporting and Analysis: Communication

- The report allowed us to analyze and test functions to determine impact.
- For example: we found that these functions would not impact us:

FC	Description
1104	Long CLIENT_ACCTNG Special Reg
1105	Long CLIENT_APPLNAME Special Reg
1106	Long CLIENT_USERID Special Reg
1107	Long CLIENT_WRKSTNNAME Special Reg

- Focus on functions that will be impacted
- For example: impactful incompatibilities

FC	Description
1	V9 SYSIBM.CHAR(decimal-expr) function
2	V9 SYSIBM.VARCHAR(decimal-expr) function. CAST (decimal as VARCHAR or CHAR)

Reporting and Analysis: Communication

- Informed the Application Teams of incompatibilities
- Informed them of required code changes and, their testing and implementation options.

- Currently in DB2v11 conversion mode(CM)
- Certain functions/application behavior is different in DB2 v11 New Function Mode(NFM) – incompatible functions
- Application team options –
  - Depending on the type of incompatibility
    - Option #1 - Application code change not required
    - Option #2 - Change code ahead of NFM
    - Option #3 - Change code after migrating to NFM

Behavior Change

- Changed CHAR/VARCHAR function behavior
- Incompatibility Function Code 2: V9 SYSIBM.VARCHAR(DECIMAL-EXPR)
  - DB2 V11 New Function Mode
    - VARCHAR (decimal column) CAST(decimal as VARCHAR)) does not return
      - leading zeroes
      - Trailing decimal point character
      - Leading blanks for positive decimal values

	VARCHAR(000.1)	VARCHAR(1000.)	VARCHAR(1.1)
DB2 11 NFM	'.1'	'1000'	'1.1'
DB2 11 CM	' 000.1'	' 1000.'	' 1.1'

Testing and Implementation

- We informed them of other functions (as VARCHAR_FORMAT) that would provide the same result.

- Application teams are able to test New Function Mode and go back using SET PATH statements.
  - This proved to be very helpful to show application teams how the results of CHAR and VARCHAR were different in SYSCOMPAT_V9 and CURRENT mode:
  - SET PATH statements:
    - SET PATH =syscompat_v9,SYSFUN,SYSPROC,SYSIBMADM,SYSIBM;
    - SET PATH =syscurrent,SYSFUN,SYSPROC,SYSIBMADM,SYSIBM;

Testing and Implementation: Fence IDs

- As a DB2 environment is set to CURRENT mode, it may be necessary to ‘fence’ IDs to provide ample time for application/SQL changes to be tested and implemented.
- Fence IDs in Profile tables of (non-batch) authorization ids
- SQLs containing CAST statements will not be ‘fenced’ even though authorization id is ‘fenced’
- Once application/SQLs are tested, delete rows in the profile tables.

Fence ID SQLs

- Fence ID SQLs:
  - Insert into sysibm.dsn_profile(authid, profileid) values (‘AUTHID', 5001);
    - 5001 is a random number
  - Insert into sysibm.dsn_profile_attributes(profileid, keywords, attribute1) values (5001,'SPECIAL_REGISTER','SET PATH = "SYSCOMPAT_V9","SYSIBM","SYSFUN","SYSPROC","SYSIBMADM"')

Be Aware…

- IFCID 366 vs IFCID 376 (Details vs Summary)
- Make sure trace is started automatically when DB2 starts
- Ensure that DB2 environments at the same maintenance levels and same module versions, otherwise I366 may not catch all incompatibilities
- I366 only reports on executed SQLs that return results
- Depending of level activity, you may find that old statements are flushed out of the STMTCACHE
- SQL TEXT sometimes not there “stmt id = 0”
  - Get the latest PI71903: ENHANCEMENTS FOR IFCID 366 AND IFCID 376 (http://www-01.ibm.com/support/docview.wss?uid=swg1PI71903&myns=swgimgmt&mynp=OCSSEPEK&mync=R&cm_sp=swgimgmt-_-OCSSEPEK-_-R)
- “Depends on Data”

Questions

- Contact Information:
  - Lillian Russell: lillian.russell@53.com
  - Lori Alexander: lori.alexander@53.com
  - Billy Sundarrajan: billy.sundarragan@53.com