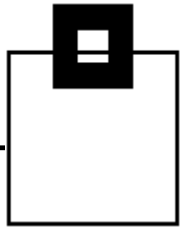
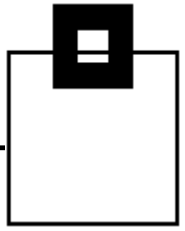


Don't Let ICIs…
put your DB2 applications in the ICU!

Ulf Heinrich

SEGUS Inc

AGENDA



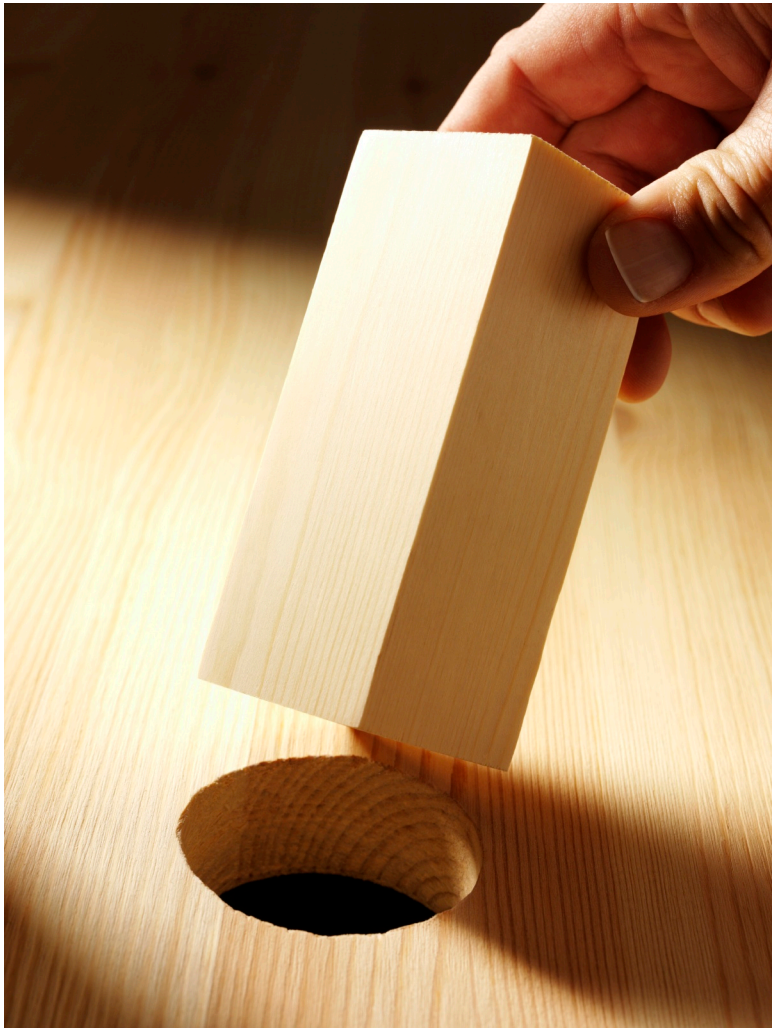
1. The what and why of ICIs
2. BiF problems
3. Ways to deal with incompatibilities
4. Some guidance
5. Freeware to the rescue
6. Customer experiences and quotes

DB2 Version Incompatibilities

- In recent versions, IBM has modified the behavior of certain DB2 functionality
  - Built-in Functions (BiFs)
  - Reserved Words
  - SQL Return Codes
  - Deprecated Functionality

- It is important that you are aware of, and track, the incompatibilities that may cause issues/problems… why?
  - Applications no longer function
  - Applications function differently
  - The results of your SQL SELECT statements can change

DB2 Version Incompatibilities

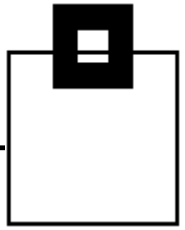


ICI =

Incompatibility Indicator

- DB2 can track when your applications use incompatible functionality
  - If you start the right traces/IFCIDs

How Many Incompatibilities Are There?



- 1: V9 version of CHAR(DEC) executed
- 2: V9 version of VARCHAR(DEC) executed
- 3: Unsupported character string representation of a TIMESTAMP
- 4: V10 default SQL path used instead of V11
- 7: SQLCODE -301 from a DB2 11 server
- 8: Stored procedure data types
- 9: TIMESTAMP TIMEZONE from DRDA
- 10: V9 version of LTRIM, RTRIM, STRIP executed
- 1101: INSERT into XML column w/o XMLDOCUMENT function
- 1102: XPATH evaluation resulted in error
- 1103: Dynamic SQL ASUTIME limit RLF issue
- 1104: CLIENT_ACCTNG longer than supported length pre-V11
- 1105: CLIENT_APPLNAME longer than supported length pre-V11
- 1106: CLIENT_USERID longer than supported length pre-V11
- 1107: CLIENT_WORKSTNNAME longer than supported length pre-V11
- 1108: CLIENT register longer than supported used for RLF
- 1109: CAST(STRING AS TIMESTAMP) using invalid string lengths as of V11
- 1110: Argument for SPACE function greater than 32764
- 1111: Optional integer argument of VARCHAR greater than 32764
- 1112: Empty XML element

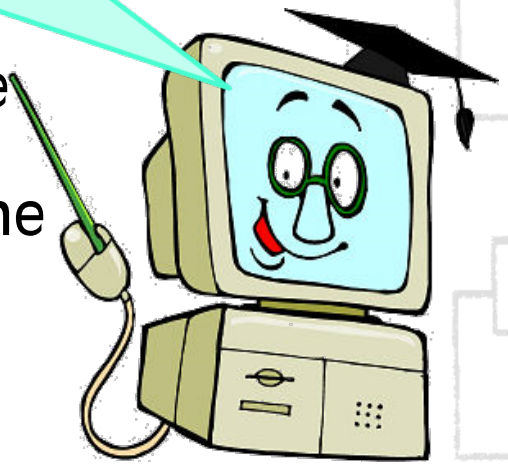
The turquoise ones are related to BIFs…

Why are incompatibilities being introduced?

- The first reaction from many is to question why IBM would change the way that DB2 works?
  - This is a reasonable question, but of course, there is a reasonable answer:

To be compatible with industry standards

- The ANSI SQL Standard
- Makes porting applications from Oracle and SQL Server to DB2 easier if they all work the same way and return the same results



Of Course…

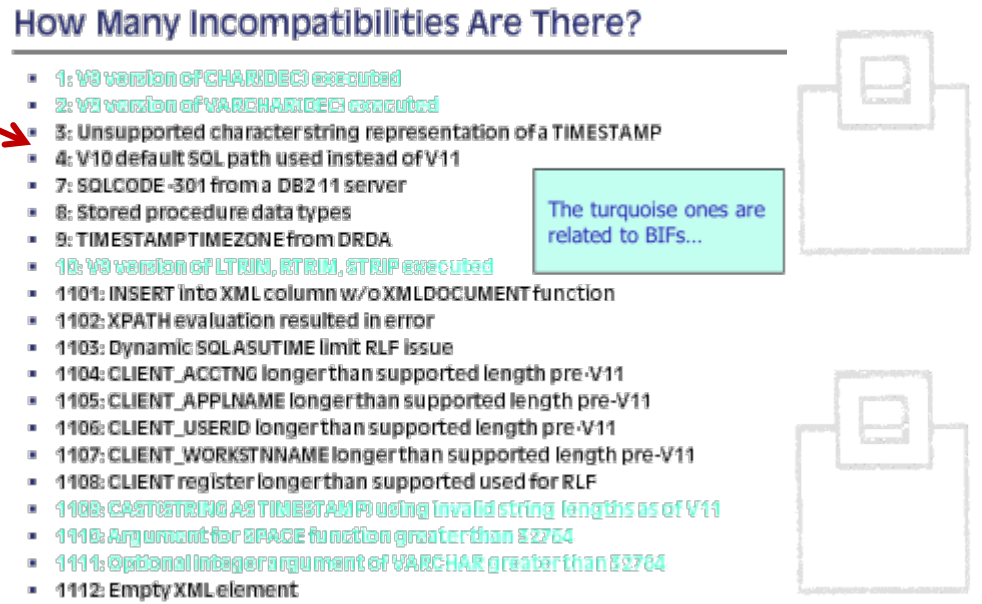
- By changing sometimes long-standing functionality, users are forced to modify their applications or live with changed functionality
  - And nobody wants to change code, no matter how simple, when the code is working…
    - Should say ***was working***

- Fortunately, there is help
  - We'll look at ways to identify changed functionality

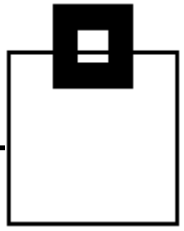
First, Let's Look at an Example: BiFs

- Built-in Functions, or BiFs, are one area that will impact many shops
  - The output of several BiFs have changed… we'll take a look at how
  - Remember the blue ones from the long list of ICIs
    - slide 6

- Before looking at the incompatibilities, a quick BiF review…

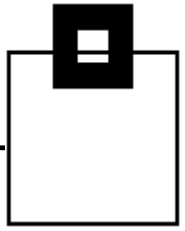


What are BiFs?



- There are 4 types of BiFs but the two most common are:
  - Aggregate Functions
    - Compute, from a group of rows, a single value for a designated column or expression.
    - This provides the capability to aggregate data, thereby enabling you to perform statistical calculations across many rows with one SQL statement.
  - Scalar Functions
    - Are applied to a column or expression and operate on a single value.
    - Contrast this with the column functions, which are applied to a set of data and return only a single result.
    - There are well over a hundred scalar functions.
      - SQL Reference Guide, Chapter 3

Aggregate vs. Scalar Functions



AGGREGATE

- ARRAY_AVG
- AVG
- CORRELATION
- COUNT / COUNT_BIG
- COVARIANCE / COVARIANCE_SAMP
- GROUPING
- MAX
- MEDIAN
- MIN
- STDDEV / STDEV_SAMP
- SUM
- VARIANCE / VARIANCE_SAMP
- XMLAGG

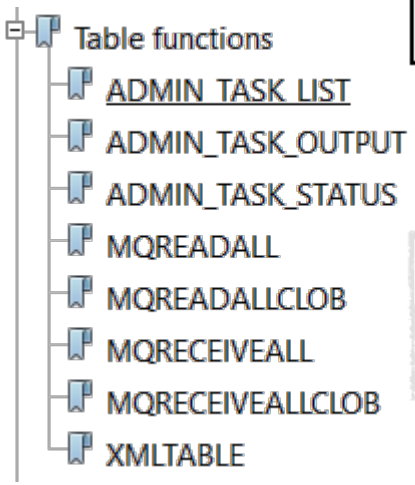
SCALAR

- ABS
- ACOS
- ASCII
- CEILING
- CHAR
- COALESCE
- LOCATE
- LOWER / UPPER
- LPAD / RPAD
- LTRIM / RTRIM
- OVERLAY
- RAND
- TRUNCATE
- VARCHAR
- ARRAY_AVG

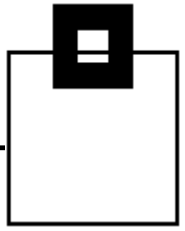
Example scalar functions; there are almost 200 of them.

OK, What are the other two types of BiFs?

- Table Functions
  - Table functions are used only in the FROM clause of an SQL statement.
  - Table functions return columns just like your normal created tables.
- Row Functions
  - Used in specific contexts
  - There is only one row function as of DB2 11
    - The UNPACK function returns a row of values that are derived from unpacking a binary string.
    - Unpacks a string that was encoded according to the PACK function.
      - The PACK scalar function returns a binary string value that contains a data type array and a packed representation of each non-null *expression* argument.



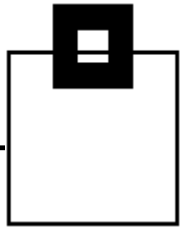
User Defined Functions



- Of course, you can also create your own UDFs, or user-defined functions
  - UDFs can be EXTERNAL or SOURCED
  - EXTERNAL
    - You write the function program
  - SOURCED
    - The UDF is based on an existing BiF

- So, if the UDF is sourced using an existing BiF, and the BiF has incompatible functionality, the UDF will be impacted, too.

Why All the Fuss About BiFs?



- The way some of the most popular scalar BiFs work started changing with DB2 10 for z/OS. For example, CHAR:

```
SELECT CHAR(DEC_COL) FROM SYSIBM.SYSDUMMY1;
```

	CHAR(000.1)	CHAR(1000.)	CHAR(1.1)
V10+	'.1'	'1000'	'1.1'
V9	' 000.1'	' 1000.'	' 1.1'

- CHAR on a DEC no longer returns:
  - Leading zeros.
  - The trailing decimal point character.
  - Leading blanks for positive decimal values. This change applies to the CHAR built-in function only.
- You can see how that might cause your application to work differently, right?

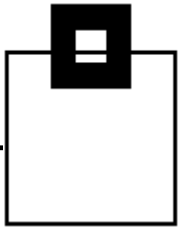
Other BiFs with “Similar” Incompatibilities

- VARCHAR
- TRIM / LTRIM / RTRIM
- STRIP
- TIMESTAMP

Customers Need a Way to Deal With all this!

- And nobody wants to change their code!
- Some help…
  - BIF_COMPATIBILITY
  - New schemas
  - APPLCOMPAT
  - New BiFs with V9 functionality

BIF_COMPATIBILITY



- IBM added a new DSNZPARM to help control the behavior of BiFs with incompatible functionality
  - Set on DSNTIPX panel
- BIF_COMPATIBILITY values
  - CURRENT
    - new functionality
  - V9
    - pre-V10 functionality for CHAR
  - V9_DECIMAL_VARCHAR
    - pre-V10 functionality for VARCHAR (and CHAR)
  - V9_TRIM
    - pre-V10 functionality for TRIM functions (and CHAR and VARCHAR)

```
DSNTIPX         INSTALL DB2 - ROUTINE PARAMETERS
 ===>_

 Enter data below:

  1 WLM PROC NAME     ===> DSN1WLM      WLM-established stored procedure JCL PROC
  2 NUMBER OF TCBS    ===> 8            Number of concurrent TCBs (1-100)
  3 MAX ABEND COUNT   ===> 0            Allowable ABENDs for a routine (0-255)
  4 TIMEOUT VALUE     ===> 180          Seconds to wait before SQL CALL or
                                        function invocation fails (5-1800,NOLIMIT)
  5 WLM ENVIRONMENT   ===>              Default WLM environment name
  6 MAX OPEN CURSORS ===> 500           Maximum open cursors per thread
  7 MAX STORED PROCS ===> 2000          Maximum active stored procs per thread
  8 MAXIMUM LE TOKENS===> 20            Maximum tokens at any time. 0-50
  9 BIF COMPATIBILITY===> CURRENT             Compatibility level for DB2 built-in
                                        functions (V9, V9_DECIMAL_VARCHAR, or
                                        CURRENT)
 10 ADMIN SCHEDULER   ===> DSNADMT      Administrative scheduler task proc name



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

Two New Schemas

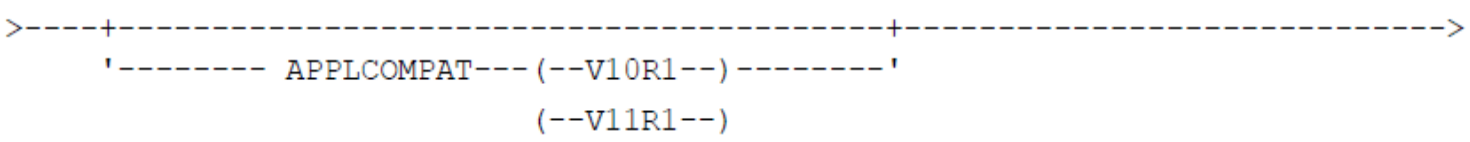
- SYSCOMPAT_V9 and SYSCURRENT
  - For CHAR, VARCHAR, and all the TRIM functions
  - Must be added before SYSIBM in the path
    - Used in the current path special register or PATH bind option

DB2 11- APPLCOMPAT

- APPLCOMPAT DSNZPARM and BIND option
- Provides support for up to two back level releases of DB2
  - DB2 11 supports V10R1 and V11R1
  - DB2 Next will support 10, 11, and Next
  - Next+1… 10 support goes away
- So you can put things off until DB2 13
  - Disclaimer: : IBM has not stated that there will actually be a version 13, nor do I have any knowledge that there will be a DB2 13. But we can all count, right?

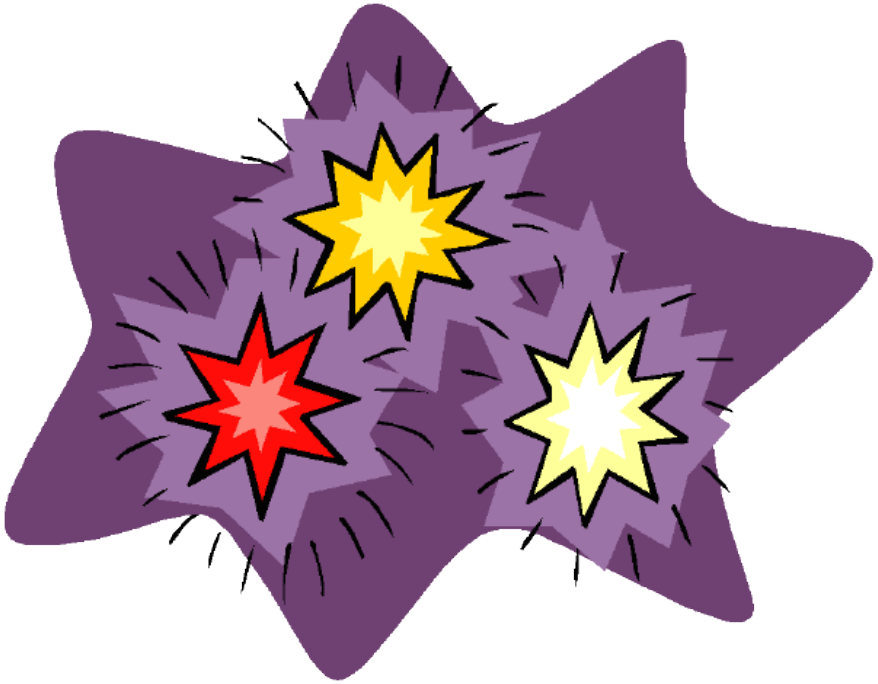
APPLCOMPAT

- The DSNZPARM value is the default value
- You can use the BIND option to control application compatibility on a program by program basis
- APPLCOMPAT does not control new function availability
  - Example: you can CREATE a GLOBAL VARIABLE using SPUFI even if the package was bound for V10R1
    - Note: Global variables are new in DB2 11
- APPLCOMPAT parameter is available on:
  - BIND PACKAGE
  - REBIND PACKAGE
  - REBIND TRIGGER
  - CREATE/ALTER PROCEDURE
  - CREATE/ALTER FUNCTION



DB2 11 – Two New BiFs

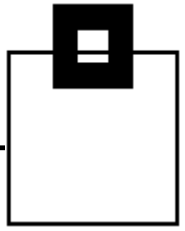
- These behave like they did for DB2 9 and prior releases, with the old functionality
  - CHAR9
  - VARCHAR9



Some Guidance

- Nobody likes to change application code, but really how hard would it be to just change:
  - CHAR → CHAR9
  - VARCHAR → VARCHAR9
- APPLCOMPAT is great because it buys us some time to make changes
  - Don't make the mistake of never moving forward your APPLCOMPAT version until IBM no longer supports it
    - Doing that just moves your mad scramble to some point in the future

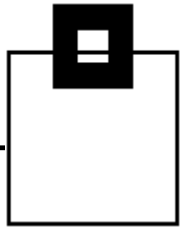
IFCIDs 366 and 376



- IBM makes it easier for us to track when we use incompatible functionality via IFCID 366
- This IFCID is output whenever DB2 detects a *possible* change of behavior from the current release to the next release.
- Remember our list of ICIs from earlier?
- This IFCID tracks them by the numbers shown in that list

- Shops started to get many 366 records being cut
  - So IBM created IFCID 376 in DB2 11
  - This is an aggregated version of the 366.

Detailed versus Aggregated

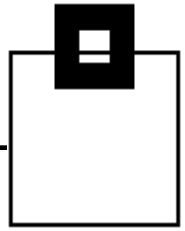


- IFCID 366
  - Identifies any new type of incompatible application change
  - Makes it possible to find packages with incompatible behavior
- IDCID 376
  - Written once for each unique dynamic cached stmt
    - And static stmt if bound V10 NFM or later.
    - For static stmts bound before V10 NFM, once per unique combo of plan, pkg ID, statement number.

How can you be best prepared?

- A lot of new information and options
  - What is your strategy for attacking ICIs?
- Some shops have many, many instances of incompatible changes running
- Are you prepared to trace IFCID 366 / 376?
  - Could be a lot of data.
- Vendors offers help…

Customer Example



Temporary solution for the problem until the affected applications are identified:

```
ZPARM BIF_COMPATIBILITY=V9_DECIMAL_VARCHAR
```


But this just “hides” the problem of course

Let’s have a quick look at a real customer and his problems…

Methodology at customer's site

- To get all relevant applications (daily, weekly etc.) since mid of April 2014 a performance trace with IFCID 366 was run in production

```
-STA TRACE(P) CLASS(32) IFCID(366) DEST(SMF) SCOPE(GROUP)
```

- For packages with static SQL additionally IFCID 63 was activated

- Identified were 18 programs:

1	Cobol Program	(Static SQL)
1	C++ Program	(Dynamic SQL) 3rd party
4	Java Programs	(Dynamic SQL)
12	High Performance Unloads	(Dynamic SQL)

Evaluation of trace records

- Evaluation of SMF records with the help of
  OMEGAMON XE FOR DB2 PERFORMANCE EXPERT

- BMC Mainview also offers reporting capabilities described in the „Performance Reporter User Guide“

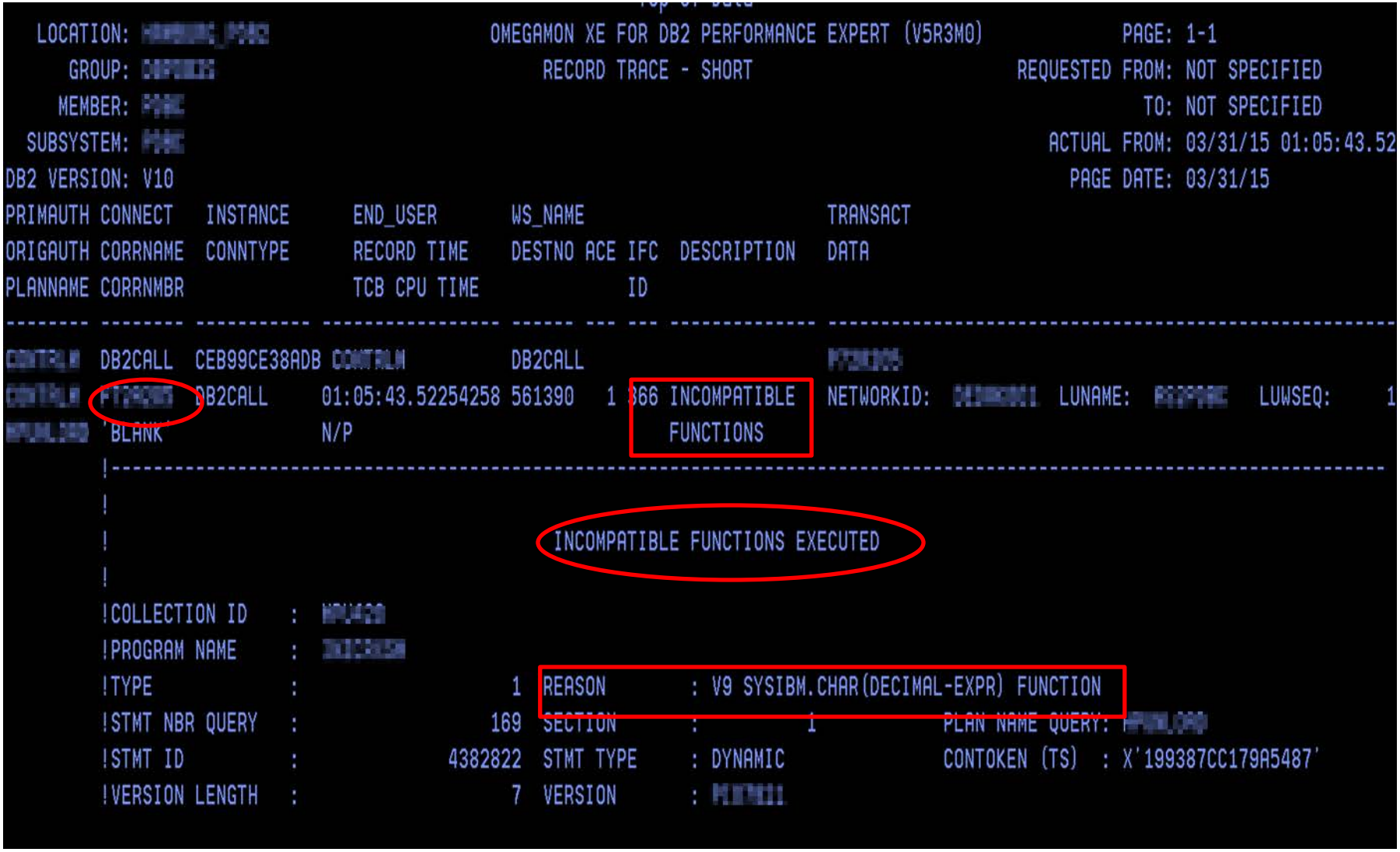
  The BMC Datacollector has to be active.

  Attention:

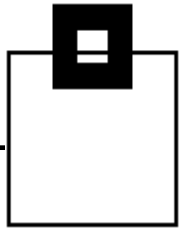
  Depending on the frequency of program calls space requirements for the SMF records will increase drastically!

JCL of the evaluation job (OMEGAMON)

```
//[redacted]   EXEC PGM=DB2PM
//STEPLIB   DD   DISP=SHR,DSN=[redacted]
//          DD   DISP=SHR,DSN=[redacted]
//INPUTDD   DD   DISP=SHR,DSN=[redacted]
//JOBSUMDD  DD   SYSOUT=*
//RTTRCDD1  DD   SYSOUT=*
//SYSIN     DD   *
  GLOBAL
                      TIMEZONE(-2)
                      EXCLUDE(
                              PRIMAUTH   (
                                          [redacted]*))
                      INCLUDE(
                              IFCID  (366)
                              GROUP  ([redacted])
                              )
  RECTRACE
              TRACE
                      LEVEL(SHORT)
                      DDNAME(RTTRCDD1)
  EXEC
```

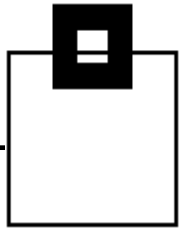

LOCATION:
OMEGAMON XE FOR DB2 PERFORMANCE EXPERT (V5R3M0)
PAGE: 1-1
GROUP:
RECORD TRACE - SHORT
REQUESTED FROM: NOT SPECIFIED
MEMBER:
TO: NOT SPECIFIED
SUBSYSTEM:
ACTUAL FROM: 03/31/15 01:05:43.52
DB2 VERSION: V10
PAGE DATE: 03/31/15
PRIMAUTH CONNECT INSTANCE END_USER WS_NAME TRANSACT
ORIGAUTH CORRNAME CONNTYPE RECORD TIME DESTNO ACE IFC DESCRIPTION DATA
PLANNAME CORRNMBR TCB CPU TIME ID
DB2CALL CEB99CE38ADB DB2CALL
DB2CALL 01:05:43.52254258 561390 1 366 INCOMPATIBLE FUNCTIONS
NETWORKID: LUNAME: LUWSEQ: 1
'BLANK' N/P
INCOMPATIBLE FUNCTIONS EXECUTED
!COLLECTION ID :
!PROGRAM NAME :
!TYPE : 1
REASON : V9 SYSIBM.CHAR(DECIMAL-EXPR) FUNCTION
!STMT NBR QUERY : 169
SECTION : 1
PLAN NAME QUERY:
!STMT ID : 4382822
STMT TYPE : DYNAMIC
CONTOKEN (TS) : X'199387CC179A5487'
!VERSION LENGTH : 7
VERSION :

SET CURRENT PATH = SYSCOMPAT_V9 BIF_COMPATIBILITY=V9_DECIMAL_VARCHAR



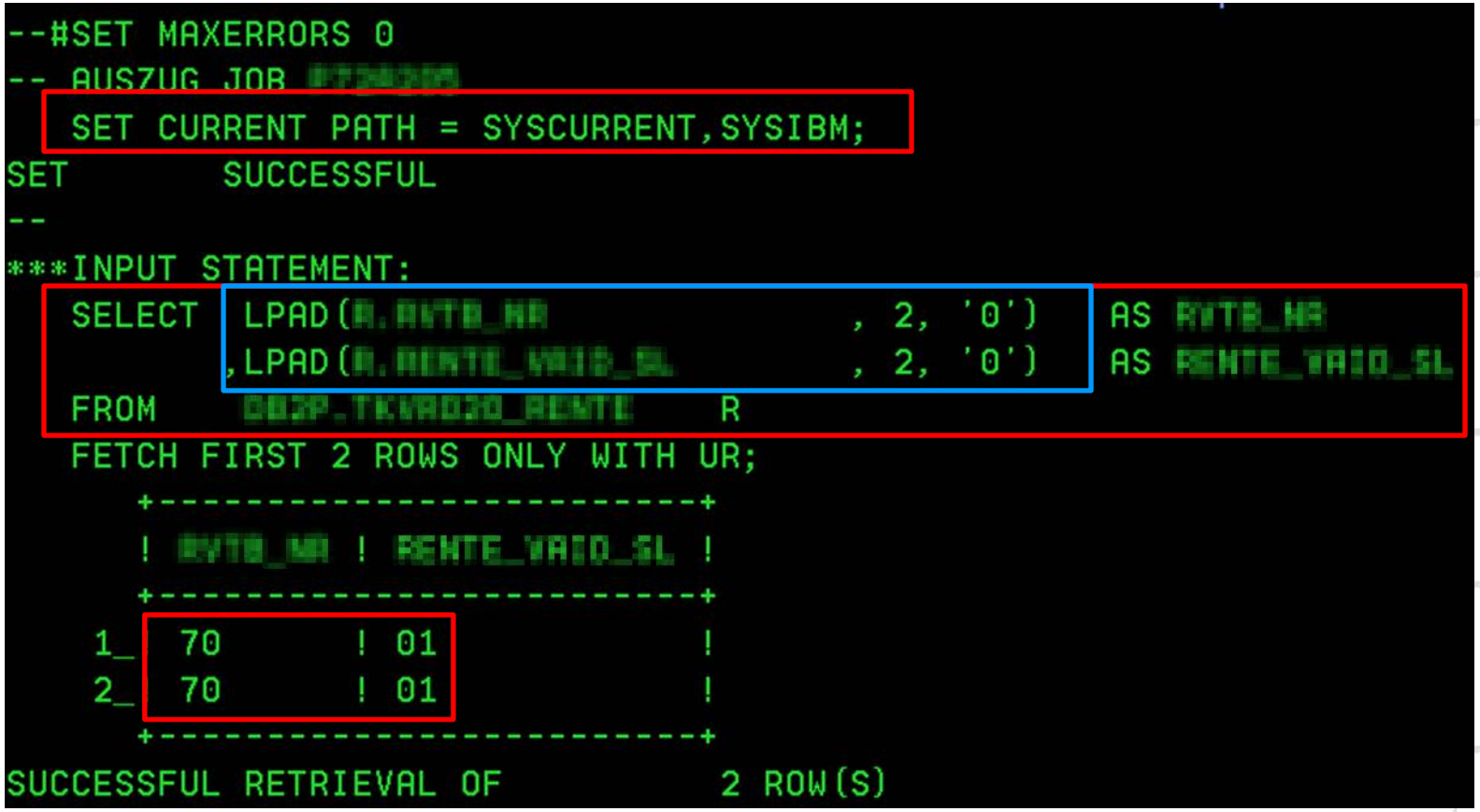
```
-- AUSZUG JOB PT2A2D5
   SET CURRENT PATH = SYSCOMPAT_V9;
RESULT OF SQL STATEMENT:
DSNT400I SQLCODE = 000,  SUCCESSFUL EXECUTION
SET       SUCCESSFUL
--
***INPUT STATEMENT:
   SELECT  SUBSTR(CHAR(R.AVTB_NR)       , 2, 2)     AS AVTB_NR
          ,SUBSTR(CHAR(R.RENTE_VAID_SL), 2, 2)     AS RENTE_VAID_SL
   FROM    DB3P.TEVA02D_RENTE    R
   FETCH FIRST 2 ROWS ONLY WITH UR;
      +-------------------------+
      ! AVTB_NR ! RENTE_VAID_SL !
      +-------------------------+
    1_! 70       ! 01            !
    2_! 70       ! 01            !
      +-------------------------+
SUCCESSFUL RETRIEVAL OF          2 ROW(S)
```

SET CURRENT PATH = SYSCURRENT,SYSIBM BIF_COMPATIBILITY=V9_DECIMAL_VARCHAR



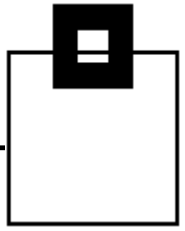
```
-- AUSZUG JOB [illegible]
   SET CURRENT PATH = SYSCURRENT,SYSIBM;
RESULT OF SQL STATEMENT:
DSNT400I SQLCODE = 000,  SUCCESSFUL EXECUTION
SET        SUCCESSFUL
--
***INPUT STATEMENT:
   SELECT   SUBSTR(CHAR(A.[illegible])       , 2, 2)      AS [illegible]
           ,SUBSTR(CHAR(A.[illegible]), 2, 2)      AS [illegible]
   FROM     [illegible]     R
   FETCH FIRST 2 ROWS ONLY WITH UR;
      +-------------------------+
      ! [illegible] ! [illegible] !
      +-------------------------+
    1_! 0         !               !
    2_! 0         !               !
      +-------------------------+
SUCCESSFUL RETRIEVAL OF          2 ROW(S)
```

Modified SQL Statement
BIF_COMPATIBILITY=V9_DECIMAL_VARCHAR



```
--#SET MAXERRORS 0
-- AUSZUG JOB
   SET CURRENT PATH = SYSCURRENT,SYSIBM;
SET        SUCCESSFUL
--
***INPUT STATEMENT:
   SELECT  LPAD(R.RVTB_NR                    , 2, '0')   AS RVTB_NR
          ,LPAD(R.RENTE_VAID_SL              , 2, '0')   AS RENTE_VAID_SL
   FROM     DB2P.TKVA030_RENTE     R
   FETCH FIRST 2 ROWS ONLY WITH UR;
      +--------------------------+
      ! RVTB_NR ! RENTE_VAID_SL !
      +--------------------------+
    1_  70       ! 01            !
    2_  70       ! 01            !
      +--------------------------+
SUCCESSFUL RETRIEVAL OF           2 ROW(S)
```

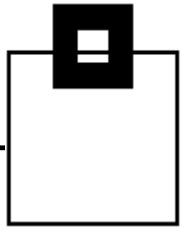
BiF HealthCheck for DB2 z/OS



- The Freeware reports ICIs based on your workload:
  - Mainframe DB2 fast Workload collector using Opx and processing engine
  - Report of detected incompatible statements being executed
- DSC and SSC (EDMPOOL) are selected and the data is reported

Freeware

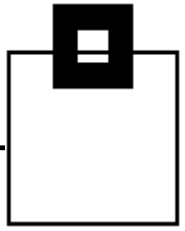
BIF HealthCheck for DB2 z/OS



- A handy batch job, or started task that reports BIF incompatibilities in DB2 10, or DB2 11
- Both, IFCID 366 and 376 can be captured as long as desired
- At the end of the capture process, or during -if desired- a report shows any affected dynamic, or static statement along with it's
  - PLAN
  - COLLECTION
  - PACKAGE
  - Number of occurrences (366)

Freeware

BIF HealthCheck for DB2 z/OS



```
SQL WorkloadExpert for DB2 z/OS Version 2.2   2015-09-24 at 15:15:58  Page    1
(C) 2013-2015 - SOFTWARE ENGINEERING GmbH

BIF Health Report for group GSA12A13 started at 2015-09-24 15:15:58.48
Type Plan     Collection          Package  ICI Count
Dyn  DSNESPUR DSNESPUR            DSNESM68   1    5
Dyn  DSNESPUR DSNESPUR            DSNESM68   5    4
Stat BIFTEST  BIFTEST_TEST        O2TESTB    1    2
```



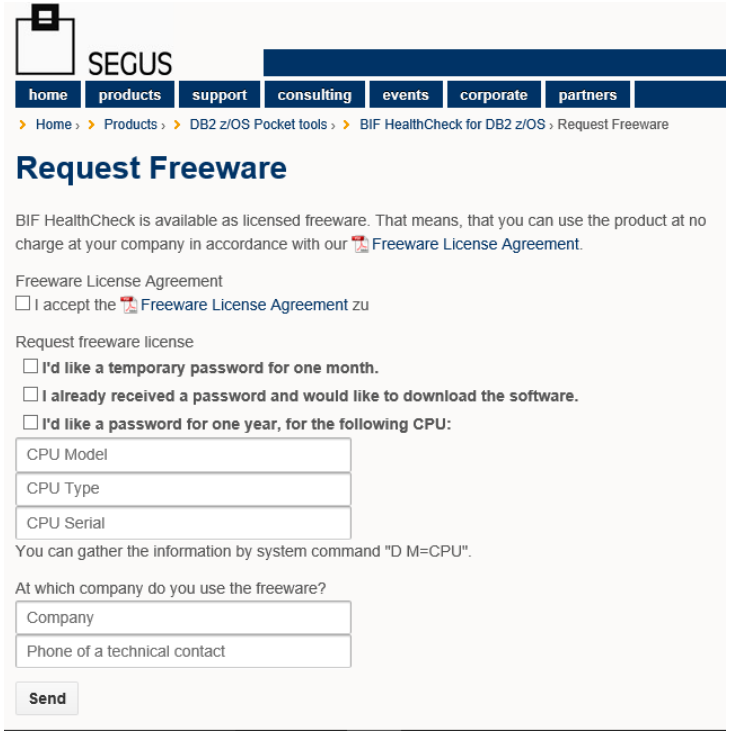
```
SQL WorkloadExpert for DB2 z/OS Version 2.2   2015-09-25 at 10:48:17  Page    1
(C) 2013-2015 - SOFTWARE ENGINEERING GmbH

BIF Health Report for subsystem DB10 started at 2015-09-25 10:48:17.23
Type Plan     Collection          Package  ICI Count
Dyn  DSNESPUR DSNESPUR            DSNESM68   1    2
Stat BIFTEST  BIFTEST_TEST        O2TESTB    1    2
```

Freeware

BIF HealthCheck for DB2 z/OS

- Download* the Freeware at

https://www.segus.com/bif



*requires a free account at segus.com

Customer experiences and quotes

Users have added their own expertise into the mix as well! Here's an example of a REXX reading the BiF/ICI data and then transforming it into a CSV file for use by application development to check if it is a real problem or not!

	A	B	C	D	E	F	G	H
1	WLX_TIMESTAMP	PACKAGE	PROGRAM	ICI_NO	REASON	STMT_GR(	STMT_ID	STMT_OR(
2	2016-04-12-00.01.13.744608	C00	PROGXY	1	DB2 9 CHAR Usage	DB29	938582	S
3	2016-04-12-00.01.13.744608	C00	PROGXY	1	DB2 9 CHAR Usage	DB29	938582	S
4	2016-04-12-00.01.13.744608	C00	PROGXY	1	DB2 9 CHAR Usage	DB29	938582	S

TRANSACTIO	END_USEF	WORKSTA	STMT_LENG	STMT_TEXT_SHORT
			487	DECLARE CR-NAMExyz CURSOR WITH HOLD FOR SELECT Collist
APPL.EXE	AUTHXXX	WS00L787	299	SELECT DISTINCT NR_KUNDEN , char (NR_P
APPL.EXE	AUTHXXX	WS00L787	335	SELECT DISTINCT char (NR_PARTNER) , N

To quote a customer:

„Due to the high number of DB2 systems, we use this through TWS.

Customer experiences and quotes

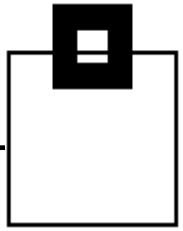
Some quotes from our customers:

The question: “Where’s the BiF?” Is answered immediately, and the SQL is categorized and displayed. The full SQL text is displayed - even for dynamic SQL.

Unlike the analyses we had previously conducted using Omegamon, it was not necessary, when using BiF HealthCheck, to collect SMF data for more than 12 months prior to the migration.

With BiF HealthCheck you don’t even need SMF data. The straight-forward approach saves huge amounts of time, manpower, and storage.

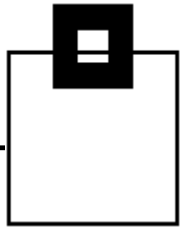
Where's the BiF?



As you have seen BiFs can cause you problems if you are *not* aware of their usage:

- Start a project plan *now* for finding the BiF
- Action and Approve any and all SQL Changes
- Store all results for comparison next time…
- Stay aware and ready!

List of DB2 10 BIF Incompatibilities

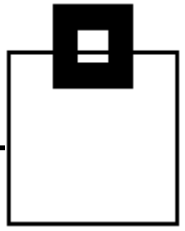


ICI	Description of the incompatibility
1	Execution of the DB2 9 for z/OS version of SYSIBM.CHAR(DECIMAL-EXPR)
2	Execution of the DB2 9 for z/OS version of SYSIBM.VARCHAR(DECIMAL-EXPR), CAST (DECIMAL AS VARCHAR), OR CAST (DECIMAL AS CHAR)
3	Use of an unsupported character string representation of a TIMESTAMP
4	Use of a USER-DEFINED FUNCTION (UDF) that has the unqualified name ARRAY_EXISTS
5	Use of a USER-DEFINED FUNCTION (UDF) that has the unqualified name CUBE
6	Use of a USER-DEFINED FUNCTION (UDF) that has the unqualified name ROLLUP

List of DB2 10 BIF Incompatibilities

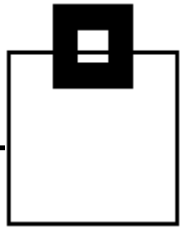
ICI	Description of the incompatibility
7	Execution of a SQL statement by a client non-Java application that included an unsupported conversion from a string type to a numeric type, while the DB2 z/OS Data Server environment was one of the following (the Data Server issues SQLCODE -301) ▪ In version 10 Conversion Mode (CM) ▪ In version 10 New-Function Mode (NFM) and implicit casting was disabled because subsystem parameter DDF_COMPATIBILITY was set to SP_PARMS_NJV, or DISABLE_IMPCAST_NJV

List of DB2 10 BIF Incompatibilities



ICI	Description of the incompatibility
8	Execution of a non-Java client that called a Stored Procedure (SP) that is on the DB2 for z/OS Data Server, while subsystem parameter DDF_COMPATIBILITY was set to SP_PARMS_NJV (the Data Server returned output argument values whose data types matched the data types of the call statement arguments).

List of DB2 11 BIF Incompatibilities

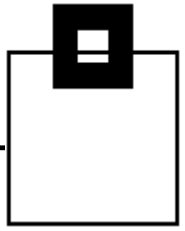


ICI	Description of the incompatibility
1	Execution of the DB2 9 for z/OS version of SYSIBM.CHAR(DECIMAL-EXPR)
2	Execution of the DB2 9 for z/OS version of SYSIBM.VARCHAR(DECIMAL-EXPR), CAST (DECIMAL AS VARCHAR), OR CAST (DECIMAL AS CHAR)
3	Use of an unsupported character string representation of a TIMESTAMP
4	Use of the DB2 10 for z/OS default SQL path instead of the V11 path, which has more implicit schemas

List of DB2 11 BIF Incompatibilities

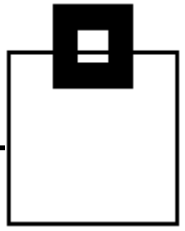
ICI	Description of the incompatibility
7	Execution of a SQL statement by a client non-Java, or Java application that included an unsupported conversion from a string type to a numeric type, or from a numeric type to a string type while the DB2 z/OS Data Server environment was one of the following (the Data Server issues SQLCODE -301) ▪ The Data Server was in version 11 New-Function Mode (NFM) ▪ APPLICATION COMPATIBILITY was set to V10R1 ▪ Implicit casting was disabled because subsystem parameter DDF_COMPATIBILITY was set to SP_PARMS_NJV, or DISABLE_IMPCAST_NJV

List of DB2 11 BIF Incompatibilities



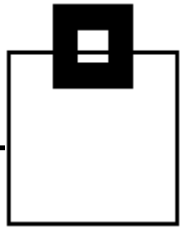
ICI	Description of the incompatibility
8	Execution of a non-Java client that called a Stored Procedure (SP) that is on the DB2 for z/OS Data Server, while subsystem parameter DDF_COMPATIBILITY was set to SP_PARMS_NJV (the Data Server returned output argument values whose data types matched the data types of the call statement arguments).
1101	Execution of an insert statement that inserts into an XML column without the XMLDOCUMENT function, which generates SQLCODE -20345 on a DB2 release prior to V11, but does not generate an error starting in V11

List of DB2 11 BIF Incompatibilities



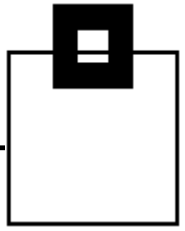
ICI	Description of the incompatibility
1102	V10 XPATH evaluation behavior was in effect, which resulted in an error (e.g. a data type conversion error occurred for a predicate that would otherwise be evaluated to false.). Starting in V11, such errors might be suppressed
1103	Execution of a dyn. SQL stmt that uses the ASUTIME limit that was set for the entire thread for RLF reactive governing (e.g. a dyn. SQL stmt is processed from a pckg A, if the ASUTIME limit was already set during other dyn. SQL processing from pckg B in the same thread, the SQL from pckg A uses the ASUTIME limit that was set during the SQL processing from pckg B). Starting with V11, dyn. SQL from multiple pckgs uses the ASUTIME limit that is set in their own pckg information.

List of DB2 11 BIF Incompatibilities



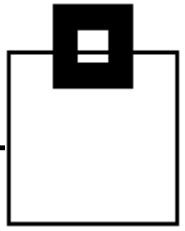
ICI	Description of the incompatibility
1104	Setting the CLIENT_USERID special register to a value that is longer than the supported length prior to V11. The value is truncated.
1105	Setting the CLIENT_WRKSTNNAME special register to a value that is longer than the supported length prior to V11. The value is truncated.
1106	Setting the CLIENT_APPLNAME special register to a value that is longer than the supported length prior to V11. The value is truncated.
1107	Setting the CLIENT_ACCTNG special register to a value that is longer than the supported length prior to V11. The value is truncated.

List of DB2 11 BIF Incompatibilities



ICI	Description of the incompatibility
1108	Setting the CLIENT_USERID, CLIENT_WRKSTNNAME, CLIENT_APPLNAME, or CLIENT_ACCTG special register to a value that is longer than the supported length prior to V11. The truncated value was used for a resource limit facility search.
1109	Execution of CAST(STRING AS TIMESTAMP) with one of the following types of input strings: ▪ A string of length 8, which DB2 treats as a STORE CLOCK value ▪ A string of length 13, which DB2 treats as a GENERATED_UNIQUE Prior to DB2 11, this behavior is invalid for a cast. It is valid for the TIMESTAMP built-in function only. Starting in V11, input to a CAST is not treated as a STORE CLOCK value, or a GENERATED_UNIQUE value.

List of DB2 11 BIF Incompatibilities



ICI	Description of the incompatibility
1110	The value of the argument of the SPACE built-in function was greater than 32784.
1111	The value of the optional integer argument of the VARCHAR built-in function was greater than 32764.

Where's the BiF?