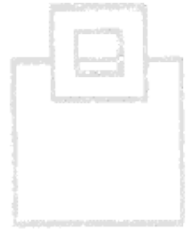
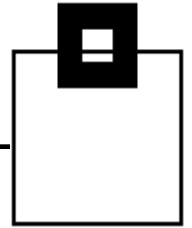


Where's the BiF??

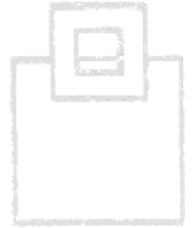
Roy Boxwell,
Senior Software Architect,
Software Engineering GmbH



Agenda

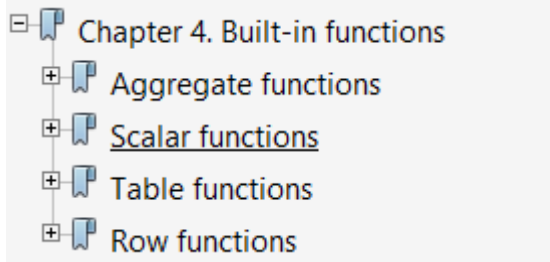


- What's a BiF?
- Where's the BiF?
- Customer Story
- Where's the BiF? – Wrap up
- Q&A

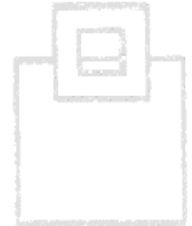
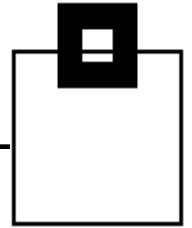


What's a BiF?

A BiF is a Built-In Function such as CHAR, DECIMAL, etc.
They exist in four different flavors:

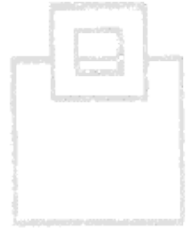
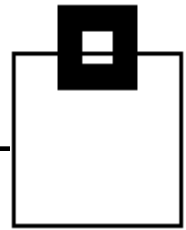
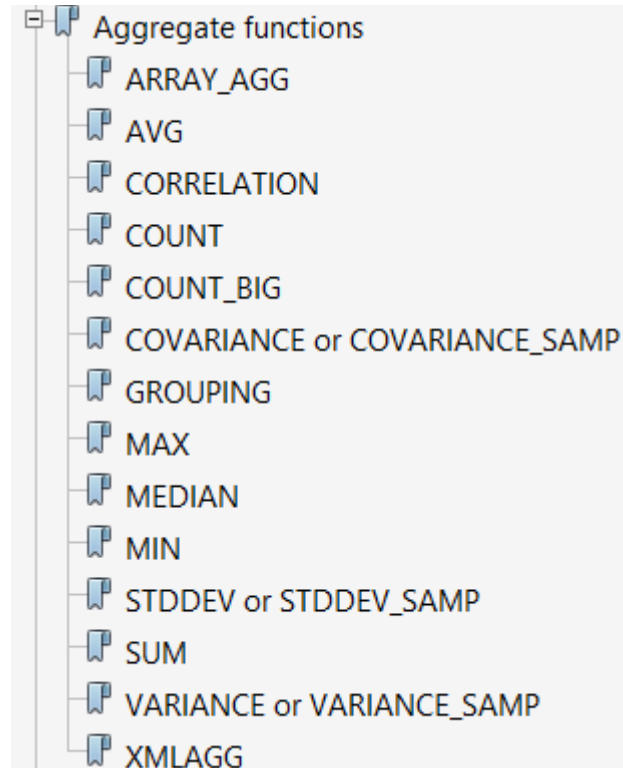


There are hundreds of them these days and every new release of DB2 brings more. In the last few DB2 releases, IBM has changed a few to make DB2 more compatible with SQL standards.



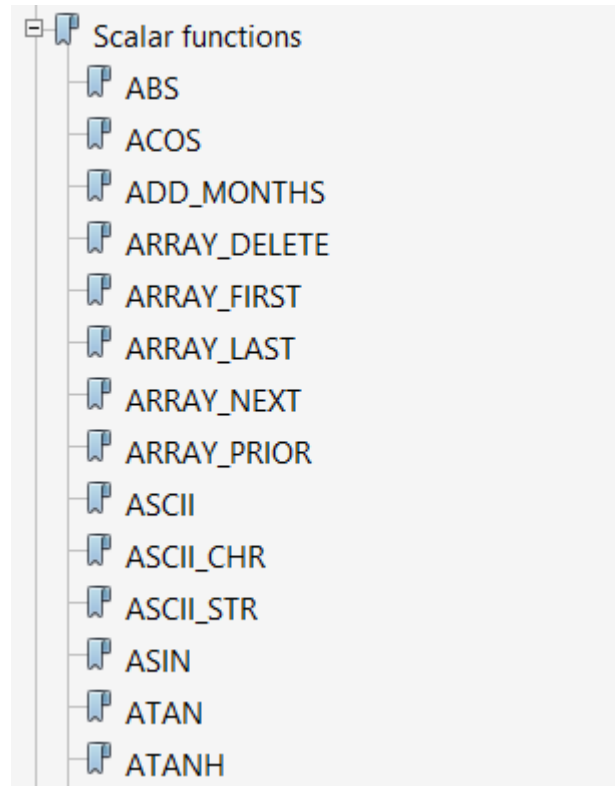
What's a BiF?

Aggregate functions we all know - but mostly we use just AVG, COUNT, MAX, MIN, and SUM:



What's a BiF?

Scalar functions are the ones we really use all the time. Here are the first few in DB2 11:



There are 184 of these...

What's a BiF?

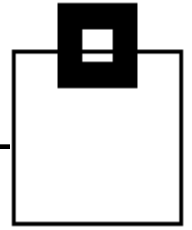
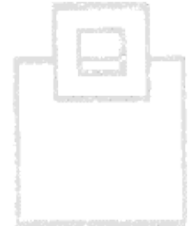
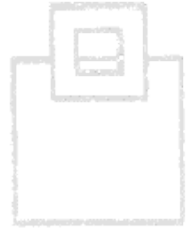
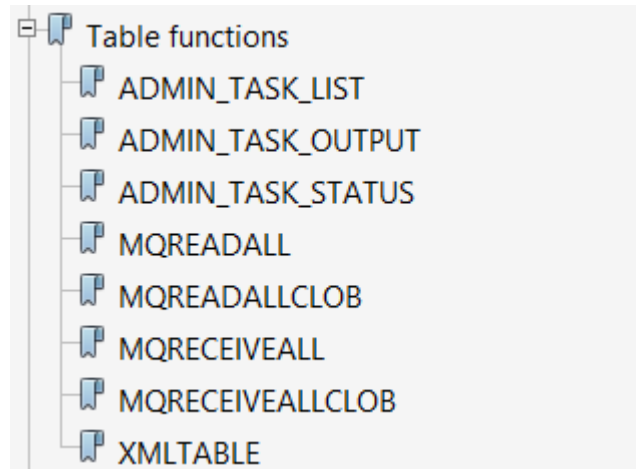
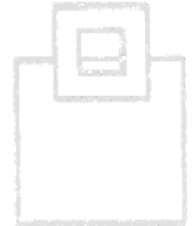
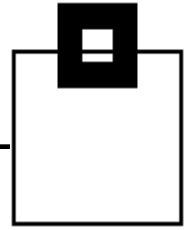
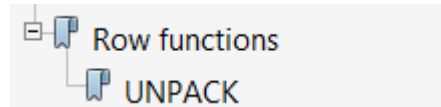


Table functions are a bit exotic and not used that often:



What's a BiF?

Row functions are brand-new, in fact there is only one at this time! Quite why it gets the plural is a puzzle!

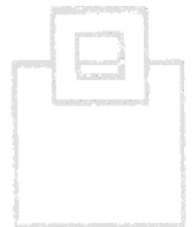
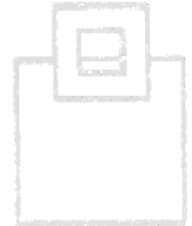
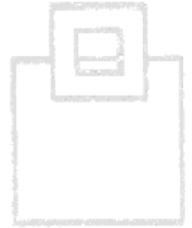
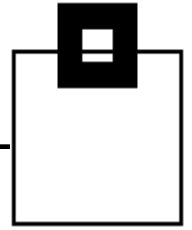


What's a BiF?

There are lots of examples of these in the DB2 Documentation and also in my newsletters, at least all new functions from DB2 V8 and up, so you can have a play in SPUFI to see what all these things can do.

So, now we know what a BiF is, Where's the BiF?

All these functions are very, very handy and useful tools that we use all the time, the problem is that the output of these BiFs sometimes changes with a DB2 release...



Where's the BiF?

The one that started the ball rolling was the change of decimal data within a CHAR function.

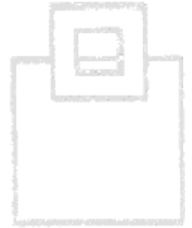
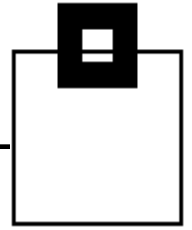
Your SQL looked like:

```
SELECT CHAR( DECCOL1 ) FROM SYSIBM.SYSDUMMY1;
```



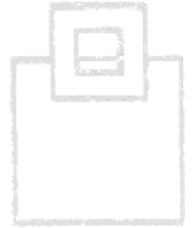
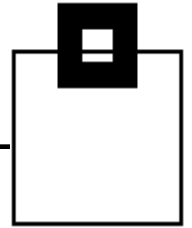
You used to get ` 000.1` and suddenly you got `.1`

This caused a few German people to jump up and down so quick-as-a-flash IBM came up with the infamous BIF_COMPATIBILITY flag in ZPARMs...

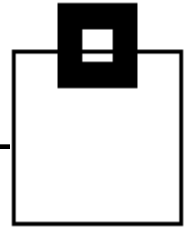


Where's the BiF? BIF_COMPATIBILITY

IBM introduced BIF_COMPATIBILITY with values V9 for old format CHAR and CURRENT for new format.



Where's the BiF? BIF_COMPATIBILITY

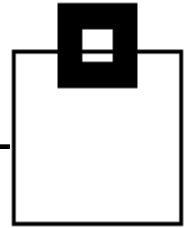


IBM introduced BIF_COMPATIBILITY with values V9 for old format CHAR and CURRENT for new format.

Then they discovered VARCHAR...so BIF_COMPATIBILITY got a new value: V9_DECIMAL_VARCHAR which included CHAR.



Where's the BiF? BIF_COMPATIBILITY

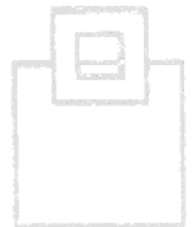


IBM introduced BIF_COMPATIBILITY with values V9 for old format CHAR and CURRENT for new format.

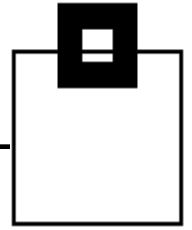
Then they discovered VARCHAR...so BIF_COMPATIBILITY got a new value: V9_DECIMAL_VARCHAR which included CHAR.



Then they discovered TIMESTAMPS...so now if BIF_COMPATIBILITY was set to V9 or V9_DECIMAL_VARCHAR illegal TIMESTAMP formats were "accepted".



Where's the BiF? BIF_COMPATIBILITY

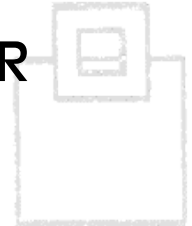


IBM introduced BIF_COMPATIBILITY with values V9 for old format CHAR and CURRENT for new format.

Then they discovered VARCHAR...so BIF_COMPATIBILITY got a new value: V9_DECIMAL_VARCHAR which included CHAR.



Then they discovered TIMESTAMPS...so now if BIF_COMPATIBILITY was set to V9 or V9_DECIMAL_VARCHAR illegal TIMESTAMP formats were "accepted".

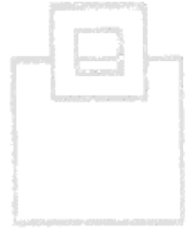
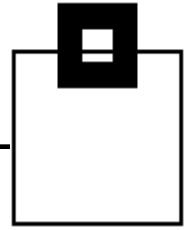


Then they discovered TRIM...so BIF_COMPATIBILITY got a new value: V9_TRIM for LTRIM, RTRIM, and STRIP which also included all of the above...



Where's the BiF? Further Fixes

Two new SCHEMAS were then created to help sort out the mess: SYSCOMPAT_V9 and SYSCURRENT (Just for CHAR, VARCHAR, and all the TRIM stuff. They must be added before SYSIBM in the path of course!)



Where's the BiF? Further Fixes

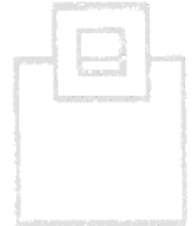
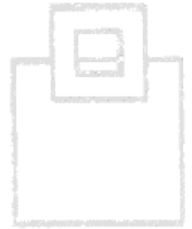
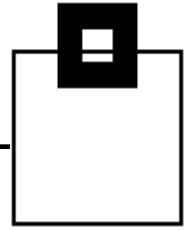
Two new SCHEMAS were then created to help sort out the mess: SYSCOMPAT_V9 and SYSCURRENT (Just for CHAR, VARCHAR, and all the TRIM stuff. They must be added before SYSIBM in the path of course!)

DB2 11: APPLCOMPAT – New ZPARM introduced.

V10R1 after migration

V11R1 after install

If set to V10R1 then DB2 10 NFM behavior is guaranteed for the next two releases. So in DB2 13 it will all go horribly wrong...That's about seven years from now!



Where's the BiF? Further Fixes

Two new SCHEMAS were then created to help sort out the mess: SYSCOMPAT_V9 and SYSCURRENT (Just for CHAR, VARCHAR, and all the TRIM stuff. They must be added before SYSIBM in the path of course!)

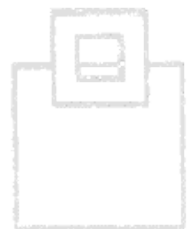
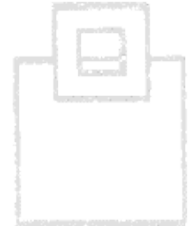
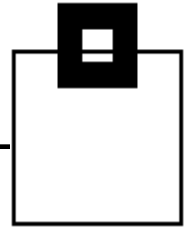
DB2 11: APPLCOMPAT – New ZPARM introduced.

V10R1 after migration

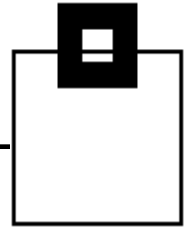
V11R1 after install

If set to V10R1 then DB2 10 NFM behavior is guaranteed for the next two releases. So in DB2 13 it will all go horribly wrong...That's about seven years from now!

DB2 11: Two new BiF's - CHAR9 and VARCHAR9.

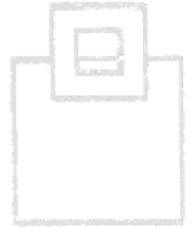


Where's the BiF? Monitoring IFCIDs



This was, to quote Monty Python, “Getting Silly”

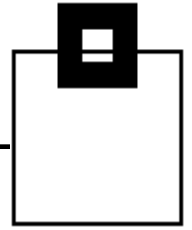
So then the IFCID 366 was born. This IFCID is output whenever DB2 detects a *possible* change of behavior from the current release to the next release.



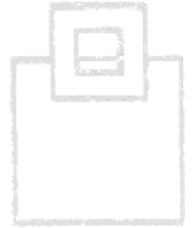
Within this IFCID are a bunch of fields (for the full details please look in your <db2hlq>.SDSNMACS(DSNDQW05) member) including a QW0366FN field that is a number (ICI – Incompatible Change Indicator, you just gotta love another TLA from IBM) that started at 1 and went to 3 with the initial version. A “1” meant a CHAR decimal problem, a “2” meant a VARCHAR decimal problem, and a “3” meant a TIMESTAMP problem.



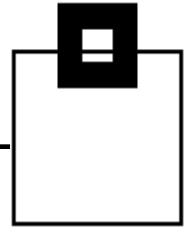
List of DB2 10 BIF Incompatibilities



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

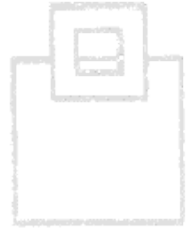


List of DB2 10 BIF Incompatibilities



ICI Description of the incompatibility

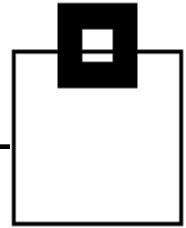
7 Execution of a SQL statement by a client non-Java application that included an unsupported conversion from a string type to a numeric type, while the DB2 z/OS Data Server environment was one of the following (the Data Server issues SQLCODE -301)



- 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

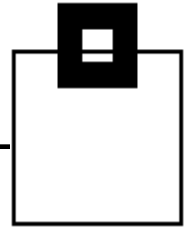


ICI Description of the incompatibility

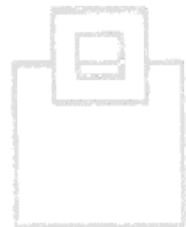
- 8 Execution of a non-Java client that called a Stored Procedure (SP) that is on the DB2 for z/OS Data Server, while subsystem parameter DDF_COMPATIBILITY was set to SP_PARMS_NJV (the Data Server returned output argument values whose data types matched the data types of the call statement arguments).



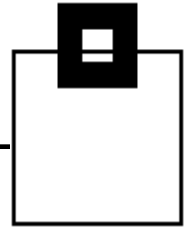
List of DB2 11 BIF Incompatibilities



- | 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 |



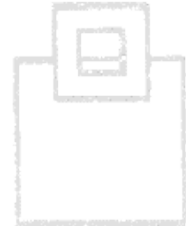
List of DB2 11 BIF Incompatibilities



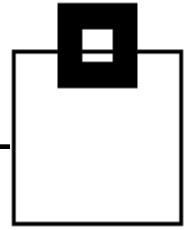
ICI Description of the incompatibility

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

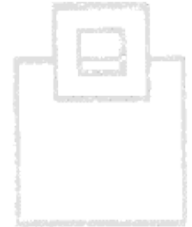


List of DB2 11 BIF Incompatibilities

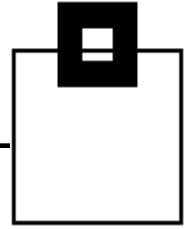


ICI Description of the incompatibility

- 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

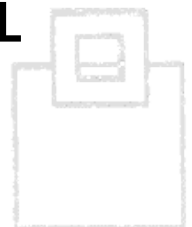
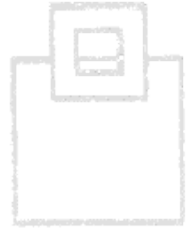


List of DB2 11 BIF Incompatibilities

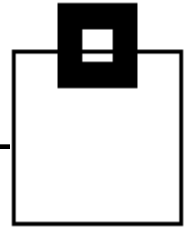


ICI Description of the incompatibility

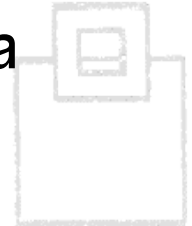
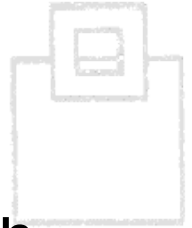
- 1102 V10 XPATH evaluation behavior was in effect, which resulted in an error (e.g. a data type conversion error occurred for a predicate that would otherwise be evaluated to false.). Starting in V11, such errors might be suppressed
- 1103 Execution of a dyn. SQL stmt that uses the ASUTIME limit that 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.



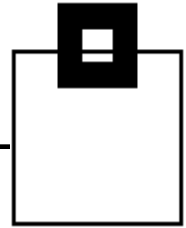
List of DB2 11 BIF Incompatibilities



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.



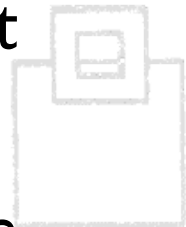
1109 Execution of CAST(String AS TIMESTAMP) with one of the following types of input strings:

- 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

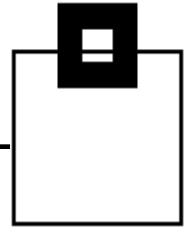


Prior to DB2 11, this behavior is invalid for a cast. It is valid for the TIMESTAMP built-in function only.

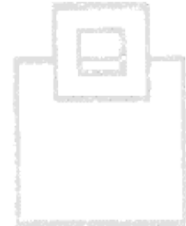
Starting in V11, input to a CAST is not treated as a STORE CLOCK value, or a GENERATED_UNIQUE value.



List of DB2 11 BIF Incompatibilities

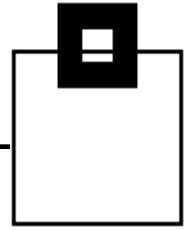


ICI	Description of the incompatibility
1110	The value of the argument of the SPACE built-in function was greater than 32784.
1111	The value of the optional integer argument of the VARCHAR built-in function was greater than 32764.



Where's the BiF? Monitoring IFCIDs

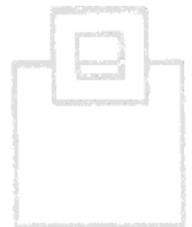
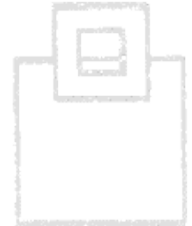
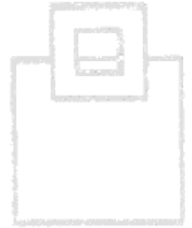
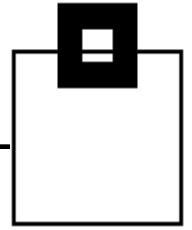
People also started getting lots of 366's and so the 376 was born, but only for DB2 11. This is an aggregated version of the 366.



Where's the BiF? Monitoring IFCIDs

People also started getting lots of 366's and so the 376 was born, but only for DB2 11. This is an aggregated version of the 366.

This was TMI (Too Much Information) as most shops were attempting to use SMF and filling up packs of disks with millions of records that contained Statement Ids of dynamic SQL that had long disappeared from the DSC.

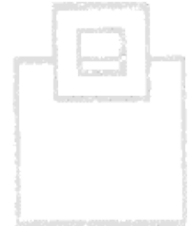
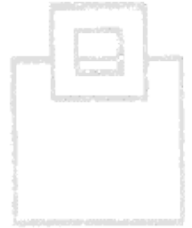
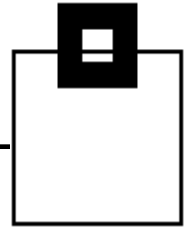


Where's the BiF? Monitoring IFCIDs

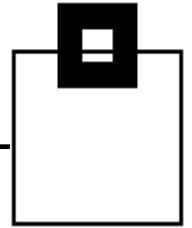
People also started getting lots of 366's and so the 376 was born, but only for DB2 11. This is an aggregated version of the 366.

This was TMI (Too Much Information) as most shops were attempting to use SMF and filling up packs of disks with millions of records that contained Statement Ids of dynamic SQL that had long disappeared from the DSC.

This is "sub-optimal"... (or, in English, a problem!)



Customer Example

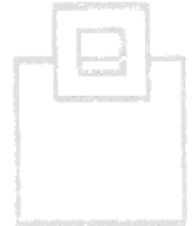
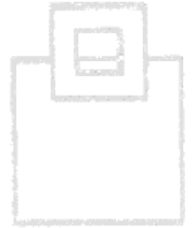


Temporary solution for the problem until the affected applications are identified:

```
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...

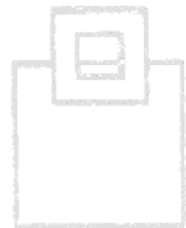
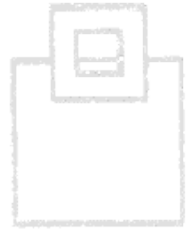
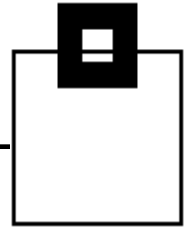


Methodology at customer's site

- 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

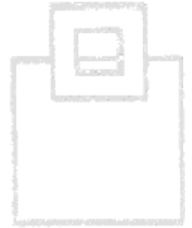
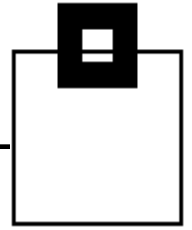
OMEGAMON XE FOR DB2 PERFORMANCE EXPERT

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

The BMC Datacollector has to be active.

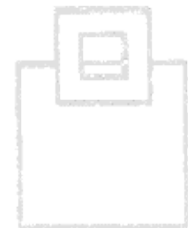
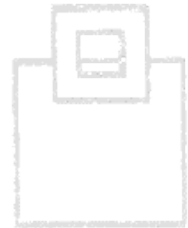
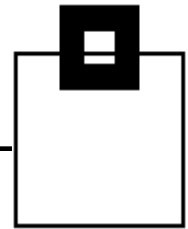
Attention:

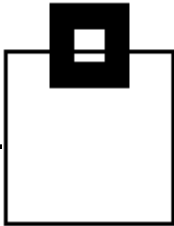
Depending on the frequency of program calls space requirements for the SMF records will increase drastically!



JCL of the evaluation job (OMEGAMON)

```
//PEV330 EXEC PGM=DB2PM
//STEPLIB DD DISP=SHR,DSN=#003,OMPE000,T0,AKANM00
// DD DISP=SHR,DSN=#003,OMPE000.SYS3,AKANM00U
//INPUTDD DD DISP=SHR,DSN=PS,SMPL0G,062,02019073
//JOBSUMDD DD SYSOUT=*
//RTTRCDD1 DD SYSOUT=*
//SYSIN DD *
GLOBAL
        TIMEZONE (-2)
        EXCLUDE (
                PRIMAUTH (
                        IS*)
        )
        INCLUDE (
                IFCID (366)
                GROUP (DBP0035)
        )
RECTRACE
        TRACE
        LEVEL (SHORT)
        DDNAME (RTTRCDD1)
EXEC
```





Top of Data

LOCATION: ~~WUJ000~~

OMEGAMON XE FOR DB2 PERFORMANCE EXPERT (V5R3M0)

PAGE: 1-1

GROUP: ~~DBP000~~

RECORD TRACE - SHORT

REQUESTED FROM: NOT SPECIFIED

MEMBER: ~~PG00~~

TO: NOT SPECIFIED

SUBSYSTEM: ~~PG00~~

ACTUAL FROM: 03/31/15 01:05:43.52

PAGE DATE: 03/31/15

DB2 VERSION: V10

PRMAUTH	CONNECT	INSTANCE	END_USER	WS_NAME	TRANSACT
ORIGAUTH	CORRNAME	CONNTYPE	RECORD TIME	DESTNO ACE IFC	DESCRIPTION
PLANNAME	CORRNMBR		TCB CPU TIME	ID	DATA

CONTROL	DB2CALL	CEB99CE38ADB	CONTROL	DB2CALL						
CONTROL	F12AC005	DB2CALL	01:05:43.52254258	561390	1	366 INCOMPATIBLE	NETWORKID:	CEB99CE38ADB	LUNAME: PG00PG00	LWSEQ: 1
WUJ000	BLANK		N/P			FUNCTIONS				

INCOMPATIBLE FUNCTIONS EXECUTED

!COLLECTION ID : ~~WUJ000~~

!PROGRAM NAME : ~~DBP000~~

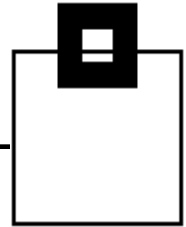
!TYPE : 1 REASON : V9 SYSIBM.CHAR(DECIMAL-EXPR) FUNCTION

!STMT NBR QUERY : 169 SECTION : 1 PLAN NAME QUERY: ~~WUJ000~~

!STMT ID : 4382822 STMT TYPE : DYNAMIC CONTOKEN (TS) : X'199387CC179A5487'

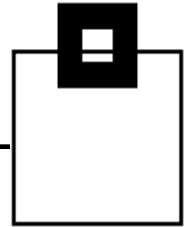
!VERSION LENGTH : 7 VERSION : ~~PG00PG00~~

SET CURRENT PATH = SYSCOMPAT_V9 BIF_COMPATIBILITY=V9_DECIMAL_VARCHAR



```
-- AUSZUG JOB PT2A205
SET CURRENT PATH = SYSCOMPAT_V9;
RESULT OF SQL STATEMENT:
DSNT400I SQLCODE = 000, SUCCESSFUL EXECUTION
SET SUCCESSFUL
--
***INPUT STATEMENT:
SELECT SUBSTR (CHAR (R.RVTD_NR) , 2, 2) AS RVTD_NR
      , SUBSTR (CHAR (R.RENTE_VAID_SL), 2, 2) AS RENTE_VAID_SL
FROM DB2P.TKVA020_RENTE R
FETCH FIRST 2 ROWS ONLY WITH UR;
+-----+
! RVTD_NR ! RENTE_VAID_SL !
+-----+
1_! 70 ! 01 !
2_! 70 ! 01 !
+-----+
SUCCESSFUL RETRIEVAL OF 2 ROW(S)
```

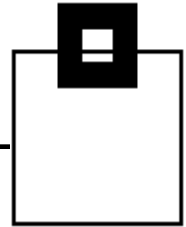
SET CURRENT PATH = SYSCURRENT, SYSIBM BIF_COMPATIBILITY=V9_DECIMAL_VARCHAR



```
-- AUSZUG JOB PFIW109
SET CURRENT PATH = SYSCURRENT, SYSIBM;
RESULT OF SQL STATEMENT:
DSNT400I  SQLCODE = 000,  SUCCESSFUL EXECUTION
SET      SUCCESSFUL
--
***INPUT STATEMENT:
SELECT  SUBSTR (CHAR (R.RYTB_NA)           , 2, 2)      AS RYTB_NA
        , SUBSTR (CHAR (R.RENTE_WAID_BL)   , 2, 2)      AS RENTE_WAID_BL
FROM    QSQP.TCV8030_RENTE      R
FETCH FIRST 2 ROWS ONLY WITH UR;
+-----+
| RYTB_NA | RENTE_WAID_BL |
+-----+
1_!  0  |                |
2_!  0  |                |
+-----+
SUCCESSFUL RETRIEVAL OF          2 ROW(S)
```

Modified SQL Statement

BIF_COMPATIBILITY=V9_DECIMAL_VARCHAR



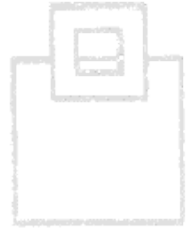
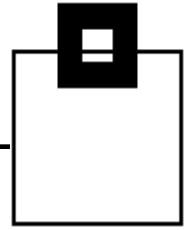
```
--#SET MAXERRORS 0
-- AUSZUG JOB #123456789
SET CURRENT PATH = SYSCURRENT, SYSIBM;
SET SUCCESSFUL
--
***INPUT STATEMENT:
SELECT LPAD (R.RYTB_NA, 2, '0') AS RYTB_NA
, LPAD (R.RENTE_VAID_SL, 2, '0') AS RENTE_VAID_SL
FROM QDZP.TKVADZ0_RENTE R
FETCH FIRST 2 ROWS ONLY WITH UR;
+-----+
! RYTB_NA ! RENTE_VAID_SL !
+-----+
1_ 70      ! 01      !
2_ 70      ! 01      !
+-----+
```

SUCCESSFUL RETRIEVAL OF 2 ROW(S)

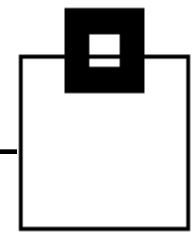


BIF HealthCheck for DB2 z/OS

- 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)



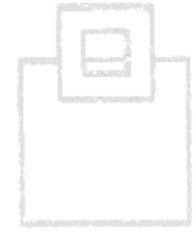
BIF HealthCheck for DB2 z/OS



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

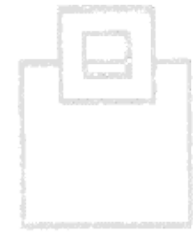
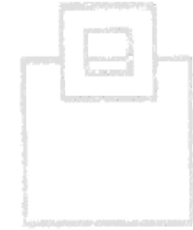
Type	Plan	Collection	Package	ICI	Count
Dyn	DSNESPUR	DSNESPUR	DSNESM68	1	5
Dyn	DSNESPUR	DSNESPUR	DSNESM68	5	4
Stat	BIFTEST	BIFTEST_TEST	O2TESTB	1	2



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



BIF HealthCheck for DB2 z/OS

- Download* the Freeware at

<https://www.segus.com/bif>

SEGUS

home products support consulting events corporate partners

> Home > > Products > > DB2 z/OS Pocket tools > > BIF HealthCheck for DB2 z/OS > Request Freeware

Request Freeware

BIF HealthCheck is available as licensed freeware. That means, that you can use the product at no charge at your company in accordance with our Freeware License Agreement.

Freeware License Agreement

I accept the Freeware License Agreement zu

Request freeware license

I'd like a temporary password for one month.

I already received a password and would like to download the software.

I'd like a password for one year, for the following CPU:

CPU Model

CPU Type

CPU Serial

You can gather the information by system command "D M=CPU".

At which company do you use the freeware?

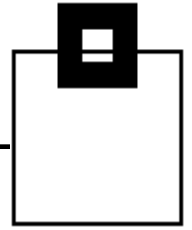
Company

Phone of a technical contact

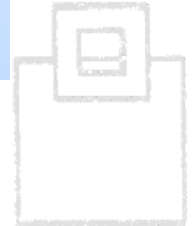
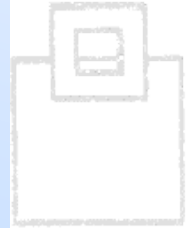
Send

*requires a free account at segus.com

Customers Goal



To already redress in DB2 10 all incompatibilities before the migration to DB2 11 and to set the ZPARM **BIF_COMPATIBILITY**, still in DB2 10, to be **CURRENT**.



What we have learnt

Freeware

- In contrast to the previously performed examinations using Omegamon, with BIF HealthCheck you don't need to collect SMF data for more than 12 months before the migration and analyze them.

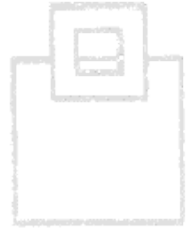
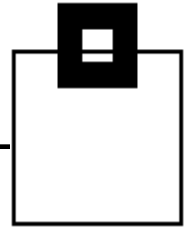
Freeware

- With BIF HealthCheck you do not even need SMF data. The BIF HealthCheck straight-forward approach saves enormous time, manpower, CPU and storage.

Freeware

- "Where is the BiF?" Is answered directly and the BIF is categorized and shown.

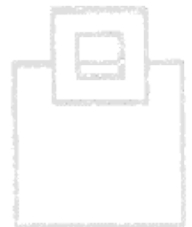
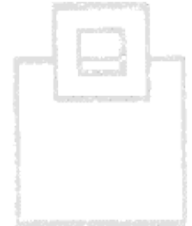
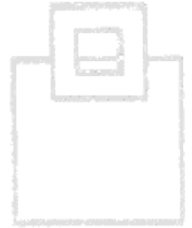
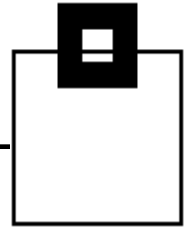
- You get the entire SQL text displayed, even for dynamic SQL.



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!



Where's the BiF?

