

Finding BIFs - ICIs¹

And How to Lead a Problem-Free Life With Them in the Future

Navigating the Challenges of moving to a new db2 Release

Siegfried Fürst - SOFTWARE ENGINEERING GmbH



Migration Planning and BIFs

In programming, you quickly learn the benefits of Built-in functions (BIFs). They can reduce unnecessary coding - and therefore save time and money. There are hundreds of BIFs available, they are everywhere and it seems like they have been around forever. So why do we need to discuss them now?

When IBM was preparing to release db2 Version 10, they pointed out problems that could arise with older versions of certain BIFs: CHAR, VARCHAR, TIMESTAMP, to name a few. This was due to IBM moving towards a more uniform SQL standard. IBM stated that the output of certain data formats in Applications would be different: "there is a possible impact to your db2 environment." These SQL changes affect several db2 releases.

BIF Compatibility Solutions

IBM recognized that there would be massive effort required at sites to make all of the necessary changes and offered certain tolerance measures in order to maintain compatibility between release levels.

Using the ZPARM BIF_COMPATIBILITY (in db2 10), a db2 subsystem can be forced into behaving as if it were still running under db2 9.

¹ This paper is just about BIFs, but BIFs are only a small part of the ICI (Incompatibility Change Indicator) list. For information about ICIs see Newsletter „2016-05 What's in an acronym? ICI“. <https://www.seg.de/en/2016-05-whats-in-an-acronym-ici>. See also flyer **BIF/ICI HealthCheck for db2 z/OS** <https://www.segus.com/media/flyer-bif.pdf>

The following excerpts from IBM's db2 10 documentation show examples of the different output formats. CURRENT is the "normal" functioning of db2 10, V9 is the compatibility mode of db2 10 running as if it were db2 9, V9_DECIMAL_VARCHAR is db2 10 running as if it were db2 9 but now for CHAR and VARCHAR.

Table 1. Example output for CHAR based on BIF_COMPATIBILITY settings

BIF_COMPATIBILITY setting	OUTPUT for CHAR(000.1)	OUTPUT for CHAR(1000.)	OUTPUT for CHAR(1.1)
V9	' 000.1 '	' 1000. '	' 1.1 '
V9_DECIMAL_VARCHAR	' 000.1 '	' 1000. '	' 1.1 '
CURRENT	' .1 '	'1000 '	'1.1 '

Table 2. Example output for VARCHAR and CAST(decimal-expression AS VARCHAR) based on BIF_COMPATIBILITY settings

BIF_COMPATIBILITY setting	Output for VARCHAR(000.1) or CAST(000.1 AS VARCHAR(3))	Output for VARCHAR(1000.) or CAST(1000. AS VARCHAR(5))	Output for VARCHAR(1.1) or CAST(1.1 AS VARCHAR(3))
V9	' .1 '	'1000 '	'1.1 '
V9_DECIMAL_VARCHAR	'0.1 '	'1000. '	'1.1 '
CURRENT	' .1 '	'1000 '	'1.1 '

Table 3. Example output for CAST(decimal-expression AS CHAR) based on BIF_COMPATIBILITY settings

BIF_COMPATIBILITY setting	Output for CAST(000.1 AS CHAR(6))	Output for CAST(1000. AS CHAR(6))	Output for CAST(1.1 AS CHAR(4))
V9	' .1 '	'1000 '	'1.1 '
V9_DECIMAL_VARCHAR	'0.1 '	'1000. '	' 1.1 '
CURRENT	' .1 '	'1000 '	'1.1 '

Additionally, db2 11 APPLCOMPAT allows the isolation of the db2 subsystem migration from the Applications level. The BIND/REBIND option APPLCOMPAT lets programs under db2 11 run as if they were still running under db2 10.

The compatibility options make planning for the migration somewhat easier and support Applications stability. They permit the customer to delay the necessary program changes by allowing for consistent BIF behavior, even after the migration.

So Why Should you Care?

As we have just seen, it is not absolutely necessary to take any action now in order to address the compatibility issues. According to IBM, there is enough time to change all SQL in Applications Programs before the migration to db2 13 – which is around 5-6 years' out – right now, IBM states: "APPLCOMPAT is only valid for two release".

One thing is certain: the problem won't just go away, it will only be delayed. And, when it is addressed again, you will still have to know where the BIF's are used and which programs are affected.

Where are all the BIF's?

In addition to the circumvention solutions previously described, IBM has introduced new IFCIDS for uncovering the SQL candidates: IFCID 366 (db2 10) and IFCID 376 (db2 11).

In z/OS, the traditional way to capture useful system information is to rely on SMF records. IFCID 366 produces an SMF (102) record for every SQL statement call. Large installations can see many millions of rows being written. This results in tremendous overhead being required to store and process these records.

In response, IBM introduced IFCID 376 (db211), which aggregates these calls by writing only one record for each unique static or dynamic SQL, thus reducing the need for storage and, hopefully, processing time.

How are these Records captured?

The IFCIDs outlined above are stored in one of three ways:

1. Written as SMF record type 102 - with the resulting high overhead.
2. Written as part of running GTF (Generalized Trace Facility) – which can cause significant CPU cycles to be used, possibly impacting the 4-hour rolling average and thus leading to additional costs.
3. Stored in OPx-Buffer – this being the most cost-effective method.

Regardless of the method chosen, it is still important to be able to:

1. Identify the statement that is incompatible.
2. Tie this statement to a program
3. Judge the overall impact of the statement and define a plan of action for remediation.

Static vs. Dynamic SQL

Once the IFCID records have been reviewed, the task still remains of culling through the data in order to identify the offending statement(s) and program(s).

In the case of static SQL, we can usually very easily tie the statement number to the record and thus the SQL source.

In the case of dynamic SQL, however, the task is far more complicated: There is a high probability that the statement in question is no longer available in the Dynamic Statement Cache (DSC), as they can appear only briefly before being flushed out. In addition, control of the source statement may not be available in all cases, such as when the SQL comes from third-party software. However, it is still extremely useful to know which actionable items can be eliminated from your own to-do list.

What's the best Method to Use?

Ideally, to capture both the static and dynamic SQL statements, a site will require:

1. Complete monitoring of both the Environmental Descriptive Manager statement pool (EDM), and the DSC. The method must insure that all SQL statements-even flushed ones-are captured.
2. The complete SQL statement must be tied to the IFCID event.
3. The program containing the SQL must also be identified.

In order to make results easy to navigate and to facilitate inter-department communications, (e.g. between Systems Programming, Database Administration and Applications Development), a graphical user interface should be available. The interface should also allow the exporting of results to Office tools for further analysis and reporting.

In addition to point-and-click methods, large installations may also find the support of mainframe processing of results via batch automation a "nice-to-have" feature.

What is collected?

1. Virtually all SQL executed in the Plex
2. The time and date of execution
3. Relevant information regarding the origination
4. Various cost factors

WLX BIF-Usage Use Case

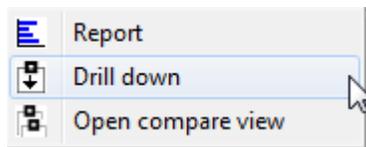
SQL workloadExpert for db2 z/OS (aka "WLX") follows exactly this methodology with its BIF-Usage Use Case, (one over 30 different Use Cases in the product).

WLX ist a Workload Warehouse and offers a variety of solutions for auditing, workload performance and optimization, application developpement ans database administration. It is integrated as plug-in for IBM Datastudio or native in Eclipse. The collection of data on the host takes place in Real-time.

The screen captures on the next page briefly show an overview of the BIF-Usage Use Case.

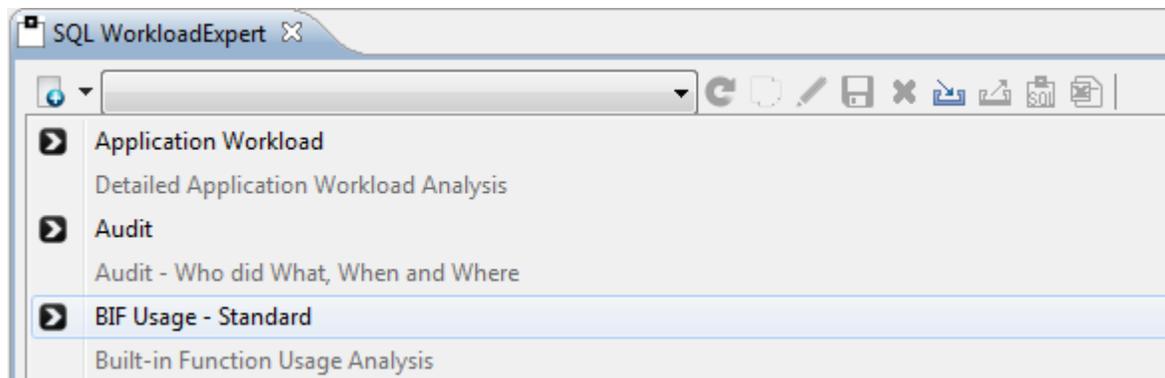
Using the BIF-Usage Use Case

"BIF-Usage" offers two separate presentation formats: Standard and Expanded. The Standard method shows grouping over two levels, while "Expanded" immediately lists all BIFs. To change methods, the drill-down function is used. This is activated by a double click or via the context menu



Access the context menu by clicking the right mouse button.

Use Case Selection - Standard



Level 1 – Collection, Package

WLX Key	Collection ID	Package	Count
2015-02-09-10.01.01.177164		IQADBACP	8
2015-02-09-10.01.01.177164		IQAXPLN	8
2015-02-09-10.01.01.177164	IQA_COLLECTION_...	IQAXPLN	8
2015-02-09-10.01.01.177164	DSNESPUR	DSNESM68	3
2015-02-09-10.01.01.177164	IQA_COLLECTION_...	TESTPGM	2
2015-02-09-13.57.55.867719	IQA_COLLECTION_...	IQAXPLN	12
2015-02-09-13.57.55.867719	DSNESPUR	DSNESM68	11

Click on the last row to show the ICI Number and reason for the BIF-usage.

Level 2 – ICI Number

WLX Key	Collection ID	Package	ICI number	Count	Reason
2015-02-09-13.57.55.867719	DSNESPUR	DSNESM68	5	10	Keyword CUBE used as unqualified UDF
2015-02-09-13.57.55.867719	DSNESPUR	DSNESM68	1	1	DB2 9 CHAR Usage

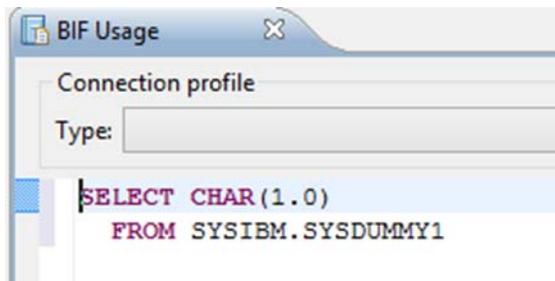
Click on the db2 9 CHAR Usage to see the details.

Detail Level

WLX Key	Collection ID	Package	ICI number	Reason	WLX DB2 SSID	Statement ID	Statement Origin	BIF_TIMESTAMP
2015-02-09-13.57.55.867719	DSNESPUR	DSNESM68	1	DB2 9 CHAR Usage	QA1B	161	D	2015-02-09-14.55.26

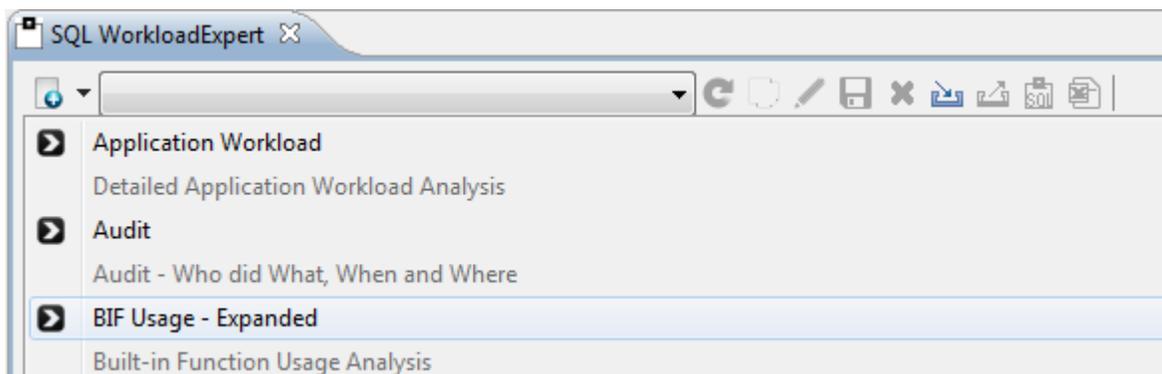
Click on the selection and choose "Show SQL text"

Complete SQL Statement Text



Use Case Selection – Expanded

The “Expanded” BIF-usage use case shows the same information in fewer clicks. However, this time the CUBE BIF is highlighted.

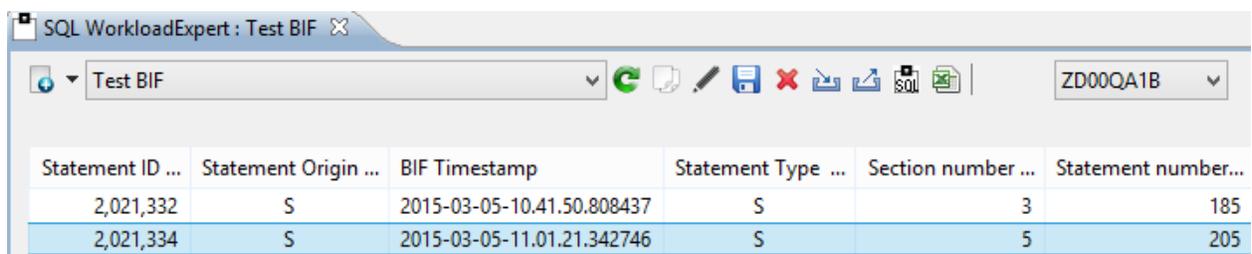


Detail Level – Part 1

The screenshot shows the "SQL WorkloadExpert : Test BIF" window. The title bar includes the text "SQL WorkloadExpert : Test BIF" and a close button. Below the title bar, there is a search box containing "Test BIF" and a dropdown menu showing "ZD00QA1B". The main area of the window displays a table with the following data:

WLX Key	Collection ID	Package	ICI numbe...	Reason	WLX DB2
2015-03-04-13.48.32.142444	MDB2VNEX_TEST	O2TESTB	1	DB2 9 CHAR Usage	QA1B
2015-03-04-13.48.32.142444	MDB2VNEX_TEST	O2TESTB	5	Keyword CUBE used as unqualified UDF	QA1B

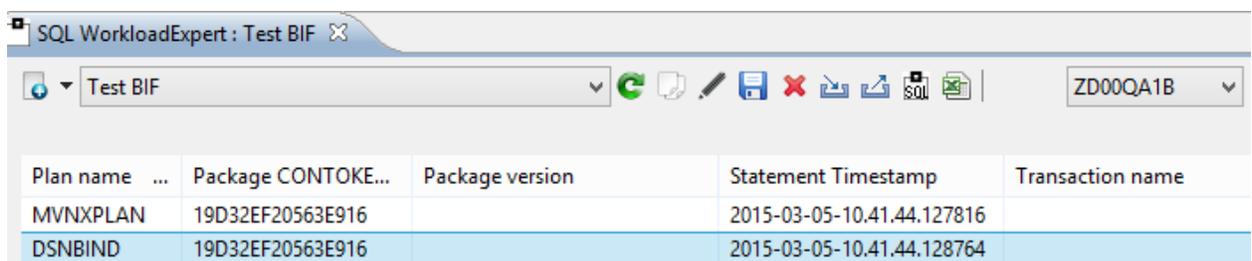
Detail Level – Part 2



The screenshot shows the SQL WorkloadExpert interface for 'Test BIF'. The table displays the following data:

Statement ID ...	Statement Origin ...	BIF Timestamp	Statement Type ...	Section number ...	Statement number...
2,021,332	S	2015-03-05-10.41.50.808437	S	3	185
2,021,334	S	2015-03-05-11.01.21.342746	S	5	205

Detail Level – Part 3

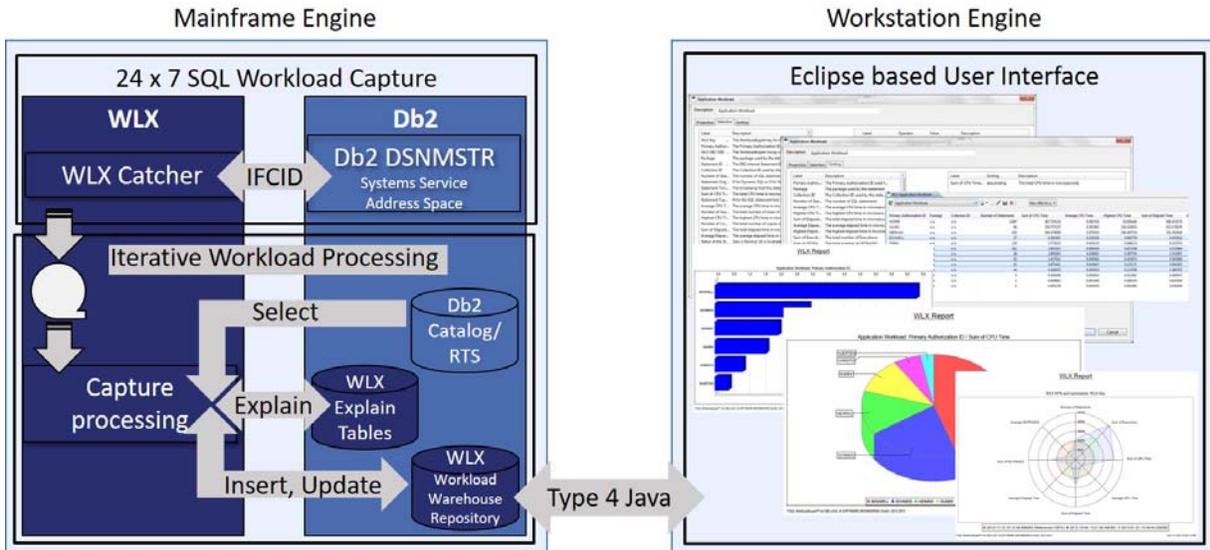


The screenshot shows the SQL WorkloadExpert interface for 'Test BIF'. The table displays the following data:

Plan name ...	Package CONTOKE...	Package version	Statement Timestamp	Transaction name
MVNXPLAN	19D32EF20563E916		2015-03-05-10.41.44.127816	
DSNBIND	19D32EF20563E916		2015-03-05-10.41.44.128764	

Delivering the BIF-Usage Use Case

The BIF-Usage Use case is only one of more than 30 optional Use Cases within WLX. The product consists of a mainframe component (z/OS 2.1 and db2 10 and above), and a workstation piece with an eclipse GUI; which can be installed as a plug in in Datastudio 4.1.1 or Eclipse native.



The complete WLX package covers the following areas of mainframe processing:



One Company's Comments

As a participant of the May 2015 GSE-Working Group Bdb2D (db2 Systems Administration) in Düsseldorf, we had the great pleasure of being accepted into the 3rd phase of SEC's GIVE and TAKE Program: BIF-Usage. (This is only one of WLX's Use Cases.)

Our WLX experience from a Systems viewpoint:

- The installation was uncomplicated and was finished within a few hours.
- WLX proved to be extremely robust in daily use.
- The use of resources was negligible.

Results of utilizing the BIF-Usage Use Case:

- Just a couple of minutes of collecting data was already enough to obtain extremely useful results.
- The graphical user interface is intuitive and offers export facilities for using Microsoft Office tools.
- In addition, the open interface allows the integration of Batch automation to process the results.
- The meaningful results can be provided to the Applications Development Team in order to ensure that the version migration will be conducted in a qualified and purposeful manner.

WLX (BIF-Usage) offers a well-rounded all-in-one solution: data collection, thoroughness, and presentation. WLX provides transparency around the expenditure necessary for changes due to version incompatibilities. It also provides a realistic expectation for planning and execution of the required changes.

(A member of the db2 User Group db2 Systems Administration, GSE Deutschland)

Summary

The result of NOT addressing BIFs in a timely manner can mean the difference between business as usual and complete chaos. (Service outages, propagating inconsistencies or corrupted data throughout the enterprise, application failures...)

Using “traditional” tools to locate the BIFs which need to be addressed is time-consuming, error-prone, and costly.

With shrinking teams and budgets, there is a definite need in the data center for modern, innovative tools such as WLX’s BIF Usage Use Case.

WLX

- Catch all SQL running on a typical system, without the overhead of a trace or utilizing monitors.
- Analyze SQL Workload by aggregate, or in detail.
- Comparatively analyze access paths and SQL.
- Report and chart metrics via a standard Eclipse interface—a free and commonly-accepted desktop tool.

Literature: db2 Workload Analysis & Trending

- (1) Flyer BIF **BIF HealthCheck & WLX BIF Usage for db2 z/OS** - <https://www.segus.com/media/flyer-bif.pdf>
- (2) BIF Freeware **BIF HealthCheck Request Freeware** - <https://www.segus.com/bif>
- (3) Flyer **SQL WorkloadExpert for db2 z/OS, 2013-2015 workshop results**, What Users want to do with an SQL Data Warehouse - <https://www.segus.com/media/flyer-wlx-uc2.pdf>
- (4) Flyer **SQL Work loadExpert for db2 z/OS**, Captures SQL cost-efficiently 24x7 without a trace <https://www.segus.com/media/flyer-wlx.pdf>
- (5) Flyer **SQL WorkloadExpert for db2 z/OS, Practical Uses** - <https://www.segus.com/media/flyer-wlx-uc.pdf>
- (6) Presentation **BIF Compatibility** – A german customer presentation translated and anonymized - <https://www.segus.com/media/p-wlx-bif-cpb-en.pdf>
- (7) Web **The SOFTWARE ENGINEERING GIVE and TAKE Program Nr. 1, 2 ,3** for the German db2 User Group - <https://www.segus.com/give-take-program-bif-explain-index-maintenance>
- (8) Statements **SQL WorkloadExpert for db2 z/OS, BIF USAGE, Customer Comments across the Industrie** - Health Care, Insurance, Banking, Car Manufacturer <https://www.segus.com/media/p-wlx-gt-statements2015-en.pdf>
- (9) Newsletter **Where is the BIF?** Roy Boxwell - <https://www.segus.com/en/2015-01-bifcids-wheres-bif>
- (10) Video **Where is the BIF?** (11 minutes video: Roy Boxwell) <https://www.segus.com/media/webinar-wlx-dw1.mp4>
- (11) User Success **How a minor change had a major impact** <https://www.segus.com/media/howaminorchange.pdf>

SEGUS Inc

14151 Park Meadow Dr.
Chantilly, VA 20151
USA

(800) 327-9650
www.segus.com
info@segus.com

SOFTWARE ENGINEERING GMBH

Heinrichstrasse 83-85
D-404239 Dusseldorf
Germany

+49-211-961-49-0
www.seg.de
info@seg.de