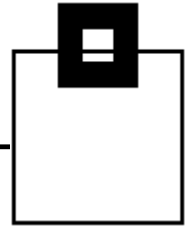


4 in 1
Bufferpool Usage
PKLIST Chains
Application Workload
SQL Time Travel

Roy Boxwell
SOFTWARE ENGINEERING GmbH
&
SEGUS Inc



AGENDA



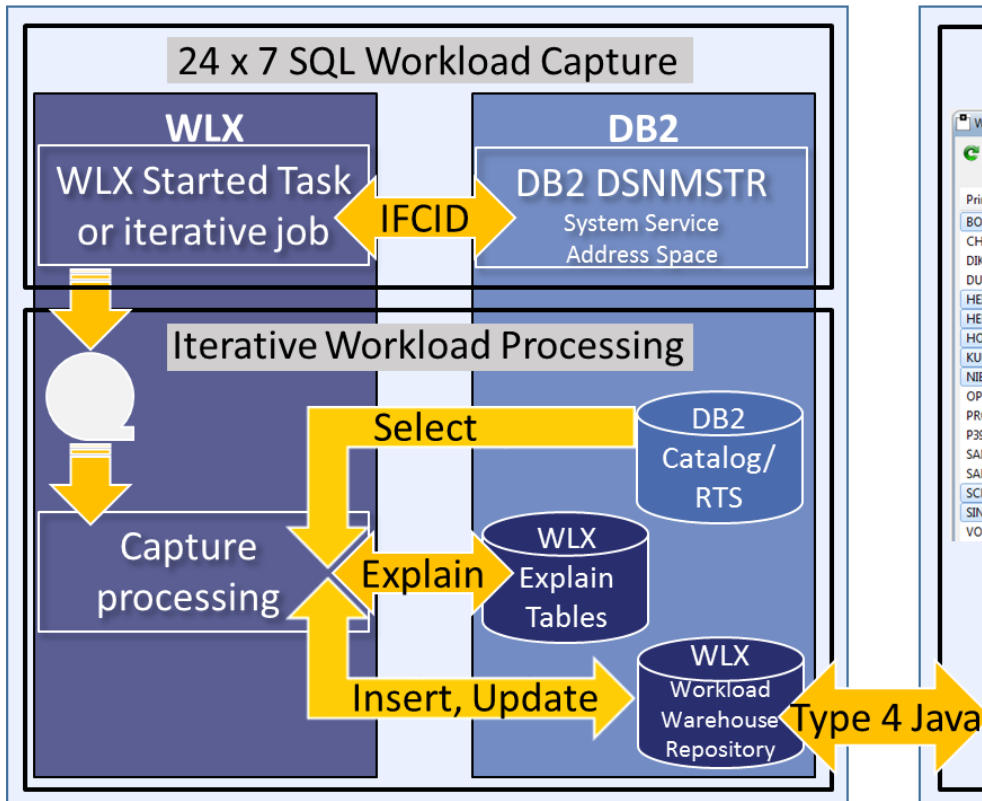
1. WLX - How it works and “What’s new”
2. Bufferpool Usage: Who’s been using my pool?
3. PKLIST Chains: Are your Packages bound correctly?
4. Application Workload: New functionality added!
5. SQL Time Travel: Follow your SQLs through time
6. Q&A Session



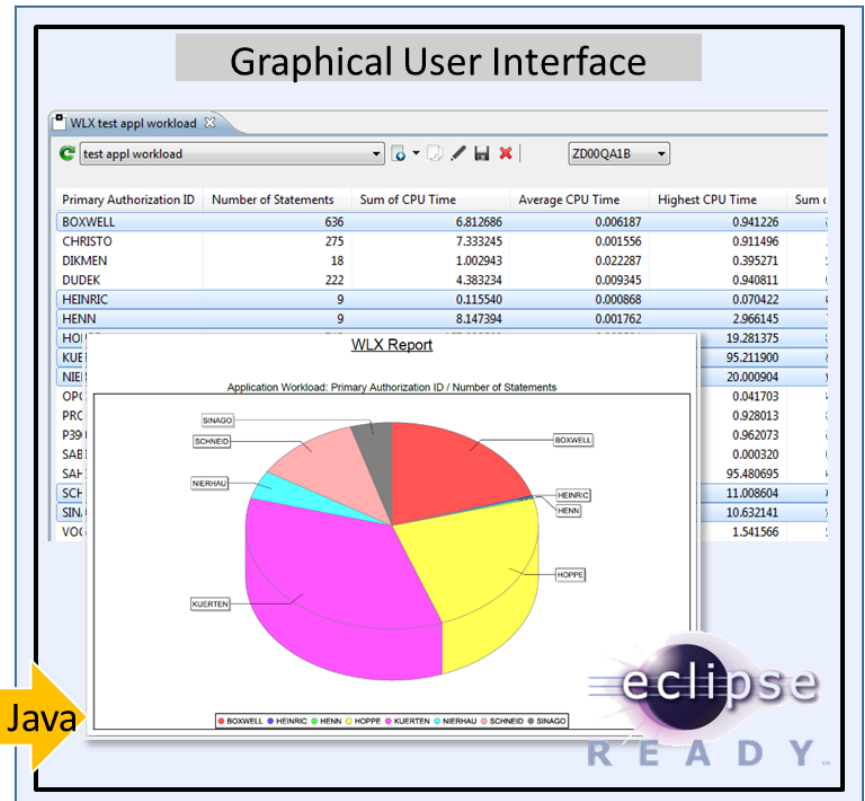
WLX - How it works

Captures the hard-to-get SQLs,
even the ones that disappear ...

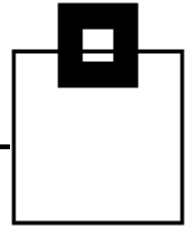
Mainframe Engine



Workstation Engine



WLX - How it works



What's New:

- Complete Utility support with IFCIDs 23, 24, 25, 219, and 220. With these IFCIDs you can easily review and tune your complete batch utility set up and flow. Easily drill down to see where the I/O wait times are occurring and review dataset allocations.

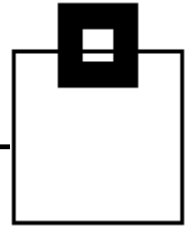


SQL WorkloadExpert : Utility Check

Utility Check ZD00QA1B

Utility ID	Database name	Tablesp...	Part...	DBID	PSID	Utility name	Utility phase	Utility sh...	Subtask c...	Number of...	Utility elapsed time	Utility CPU time
PTFRO000ROU0...	R510D0PT	R510X006	0	281	404	REORG	SORTBLD		8	1	0.000000	0.000000
PTFRO000ROU0...	R510D0PT	R510S006	0	281	401	REORG	SORTIN		0	0	0.000000	0.000000
PTFRO000ROU0...	R510D0PT	R510S006	0	281	401	REORG	SORT		0	1	0.000000	0.000000
PTFRO000ROU0...	R510D0PT	R510S006	0	281	401	REORG	SORTOUT		0	1	0.000000	0.000000
PTFRO000ROU0...	R510D0PT	R510S006	0	281	401	REORG	SORTIN		0	0	0.000000	0.000000
PTFRO000ROU0...	R510D0PT	R510S006	0	281	401	REORG	SORT		0	0	0.000000	0.000000
PTFRO000ROU0...	R510D0PT	R510S006	0	281	401	REORG	SORTOUT		0	0	0.000000	0.000000
PTFRO000ROU0...	R510D0PT	R510S006	0	281	401	REORG	LOG		8	2	0.000000	0.000000
PTFRO000ROU0...	R510D0PT	R510S006	0	281	401	REORG	LOGAPPLY		0	0	0.000000	0.000000
PTFRO000ROU0...	R510D0PT	R510S006	0	281	401	REORG	SWITCH		9	0	0.000000	0.000000
PTFRO000ROU0...	R510D0PT	R510S006	0	281	401	REORG	UTILTERM		9	0	0.000000	0.000000
PTFRO000ROU0...	R510D0PT	R510S01	0	281	2	REORG	UNLOAD		0	0	0.000000	0.000000

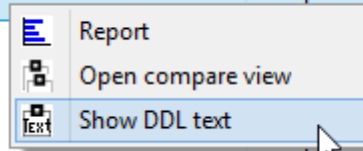
WLX - How it works



What's New:

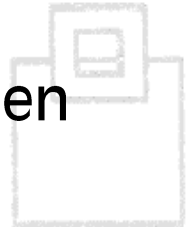
- DDL Support with IFCID 62. Every time anyone or anything creates/alters/drops an object the IFCID 62 is output. Now you can see exactly who did what and when to your DB2 objects. Optionally supported is DDL Generation so that you can see a change history.

2015-11-19-11.26.54.138008	D3	ALTER TABLE	TABLE	IQA0610	IQATW00
2015-11-19-11.26.54.287528	D3	ALTER TABLE	TABLE	IQA0610	IQATW00
2015-11-19-11.26.54.420673	D3	ALTER TABLE	TABLE	IQA0610	IQATW00
2015-11-19-11.26.54.551698	D3	ALTER TABLE	TABLE	IQA0610	IQATW00
2015-11-19-11.26.54.695247	D3	ALTER TABLE	TABLE	IOA0610	IOATW00

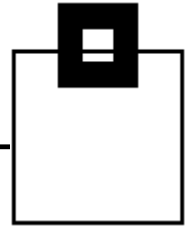


```
DDL statement

CREATE TABLE "IQA0610"."IQATW001"
("WLX_TIMESTAMP"          TIMESTAMP( 6)          NOT NULL
,"STMT_GROUP_SSID"        CHAR( 8)                FOR SBCS DATA NOT NULL
,"STMT_ID"                BIGINT                  NOT NULL
,"STMT_ORIGIN"            CHAR( 1)                FOR SBCS DATA NOT NULL
,"STMT_TIMESTAMP"         TIMESTAMP( 6)          NOT NULL
,"STMT_TYPE"              CHAR( 1)                FOR SBCS DATA NOT NULL
,"USER_NO"                BIGINT                  NOT NULL
,"COPIES_NO"              BIGINT                  NOT NULL
,"STATUS"                 SMALLINT                NOT NULL
,"LITERAL_REPLACEMENT"    CHAR( 1)                FOR SBCS DATA NOT NULL
```



WLX - How it works

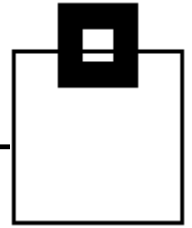


What's New:

- Enhanced audit support with IFCIDs 140, 141, 142, 143, and 144. Now you get near real-time information about any failed authorization checks, GRANT and REVOKE, CREATE/ALTER/DROP against “audit” tables (Including MLS tables), 1st change of an audited object in a unit of recovery, and 1st select from an audited object in a unit of recovery.
- EXPLAIN table support (see later in this presentation for details)
- webve launcher (see later in this presentation for details)



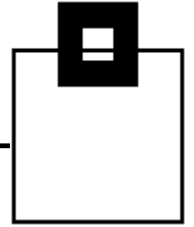
AGENDA



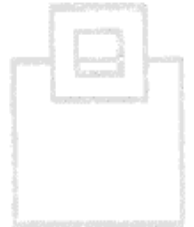
1. WLX - How it works
2. **Bufferpool Usage: Who's been using my pool?**
3. PKLIST Chains: Are your Packages bound correctly?
4. Application Workload: New functionality added!
5. SQL Time Travel: Follow your SQLs through time
6. Q&A Session



Bufferpool Usage



WLX collects all the SQL that runs on a system and optionally all the used objects. In this case it is also possible to then trawl through the data and output Bufferpool usage statistics.



Now remember that WLX is based on workload and single SQLs and that the KPIs are SQL based so the data returned from the Bufferpool Usage is fuzzy at best!

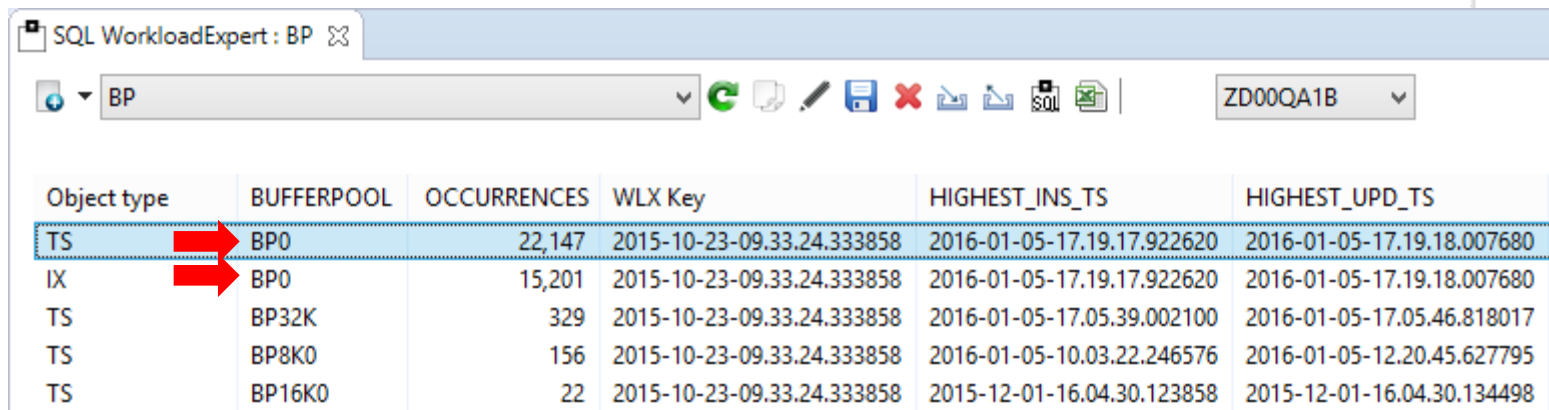


However, if SQLs are single table SQLs then the data is 100% correct.



Bufferpool Usage

Looking at our test center data you can instantly see that we are very bad! We **only** use the DB2 Catalog and Directory Bufferpools!

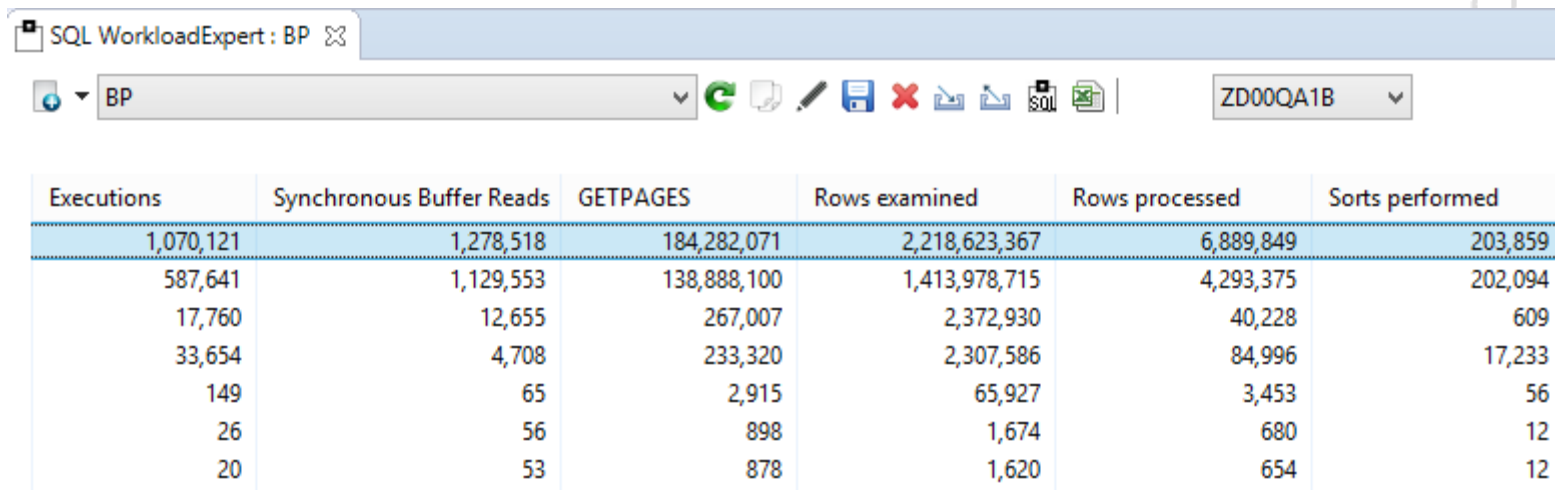


Object type	BUFFERPOOL	OCCURRENCES	WLX Key	HIGHEST_INS_TS	HIGHEST_UPD_TS
TS	BP0	22,147	2015-10-23-09.33.24.333858	2016-01-05-17.19.17.922620	2016-01-05-17.19.18.007680
IX	BP0	15,201	2015-10-23-09.33.24.333858	2016-01-05-17.19.17.922620	2016-01-05-17.19.18.007680
TS	BP32K	329	2015-10-23-09.33.24.333858	2016-01-05-17.05.39.002100	2016-01-05-17.05.46.818017
TS	BP8K0	156	2015-10-23-09.33.24.333858	2016-01-05-10.03.22.246576	2016-01-05-12.20.45.627795
TS	BP16K0	22	2015-10-23-09.33.24.333858	2015-12-01-16.04.30.123858	2015-12-01-16.04.30.134498

Further, you can see that we are using the same BP for TS and for IX which is also against normal ROT...

Bufferpool Usage

Scrolling right we then get all the KPIs (there are more even further off to the right hand side)



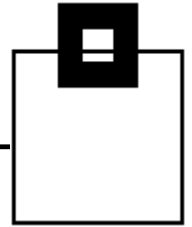
SQL WorkloadExpert : BP

BP ZD00QA1B

Executions	Synchronous Buffer Reads	GETPAGES	Rows examined	Rows processed	Sorts performed
1,070,121	1,278,518	184,282,071	2,218,623,367	6,889,849	203,859
587,641	1,129,553	138,888,100	1,413,978,715	4,293,375	202,094
17,760	12,655	267,007	2,372,930	40,228	609
33,654	4,708	233,320	2,307,586	84,996	17,233
149	65	2,915	65,927	3,453	56
26	56	898	1,674	680	12
20	53	878	1,620	654	12

Very nice “Top Level” Bufferpool KPI data!

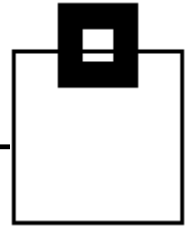
AGENDA



1. WLX - How it works
2. Bufferpool Usage: Who's been using my pool?
- 3. PKLIST Chains: Are your Packages bound correctly?**
4. Application Workload: New functionality added!
5. SQL Time Travel: Follow your SQLs through time
6. Q&A Session



PKLIST Chains



Again, WLX collects all the SQL that runs on a system and for static SQL the Collection ID and Package. Using this data you can review how your multi-collection Plans are being bound and then, perhaps, change the order of the Collection IDs in the BIND statement to speed up processing.



Naturally this is only useful if you are **not** using
SET CURRENT PACKAGE PATH
or

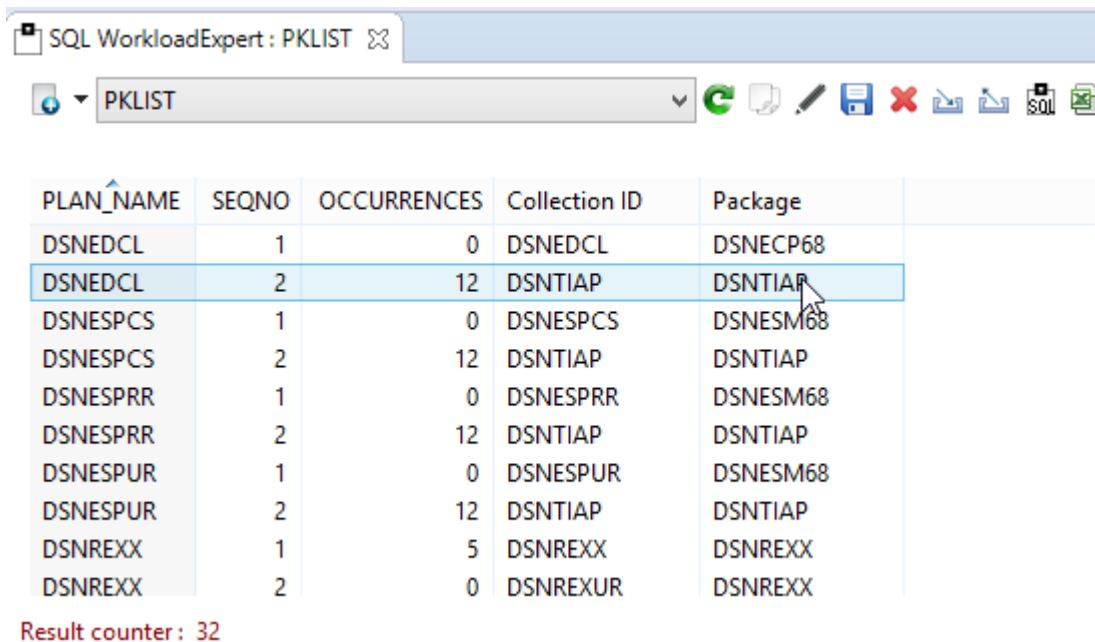


SET CURRENT PACKAGESET
as then DB2 does no search for the package to be executed!



PKLIST Chains

In our test center there is not that much going on but even here you can see that we have multiple packages that could be better bound:



PLAN_NAME	SEQNO	OCCURRENCES	Collection ID	Package
DSNEDCL	1	0	DSNEDCL	DSNECP68
DSNEDCL	2	12	DSNTIAP	DSNTIAP
DSNESPSCS	1	0	DSNESPSCS	DSNESM68
DSNESPSCS	2	12	DSNTIAP	DSNTIAP
DSNESPRR	1	0	DSNESPRR	DSNESM68
DSNESPRR	2	12	DSNTIAP	DSNTIAP
DSNESPUR	1	0	DSNESPUR	DSNESM68
DSNESPUR	2	12	DSNTIAP	DSNTIAP
DSNREXX	1	5	DSNREXX	DSNREXX
DSNREXX	2	0	DSNREXUR	DSNREXX

Result counter: 32

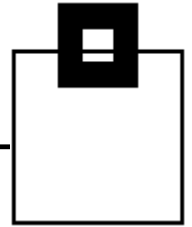
PKLIST Chains

This Use Case also returns all multi-collection bound plans, even if not used in the Workload, just to give the DBA a good overview of how large the existing PKLIST chains are.

Here's some interesting "real world" data:

W...	1	2563	R...
W...	2	30	A...
W...	3	107	B...
W...	4	105	E...
W...	5	191	H...
W...	6	80	H...
W...	7	120	N...
W...	8	0	N...
W...	9	22	R...
W...	10	9	T...
W...	11	97	W...
W...	12	92	W...
W...	13	0	*
W...	14	2563	R...

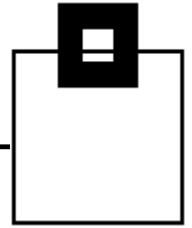
AGENDA



1. WLX - How it works
2. Bufferpool Usage: Who's been using my pool?
3. PKLIST Chains: Are your Packages bound correctly?
4. **Application Workload: New functionality added!**
5. SQL Time Travel: Follow your SQLs through time
6. Q&A Session



Application Workload

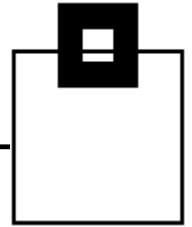


Application Workload is the bread and butter of WLX and we have enhanced this base functionality with some extremely useful features:

1. The ability to view the EXPLAIN table data that WLX, optionally, generates and saves.
2. We have added an ability to call Visual Explain from within WLX using the webve API. This will do a dynamic EXPLAIN and display the VE output in a browser screen. For this to work, you will need DS 4.1.0.0 or higher and you must have the, DS supplied, internal Web Server up and running.

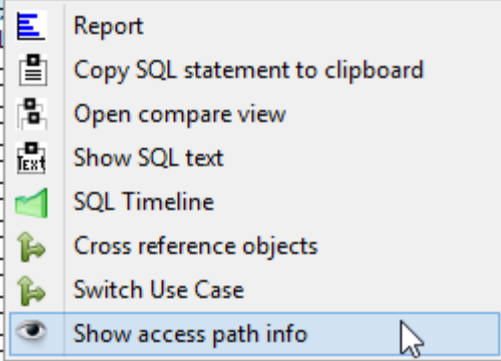


Application Workload



When in the Application Workload Use Case just right-clicking on an SQL pulls up this context menu:

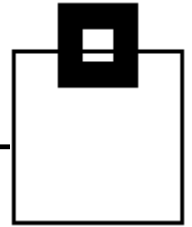
Package	Executions	Primary Authorization ID	CPU Time
PHC			3.661574
SYSL			3.459614
PHC			3.424317
PHC			3.329793
PHC			2.879885
PHC			2.804829
PHC			2.586268
PHC			2.424773
PHC			2.311044
PHC			2.227524



Here you can see that we have added a bunch of features! First up is the “Show access path info” icon



Application Workload



The next views have three areas of interest. First is the key details:

Properties SQL WorkloadExpert : Application Workload Drill-Down1 SQL WorkloadExpert : Explain view dynamic9

Statement identifier	
WLX timestamp	2015-10-23 09:33:24.333858
Statement group SSID	QA1B
Statement ID	10167
Statement origin	D
Statement timestamp	2015-12-09 14:19:55.033002



Then the Estimated values (from the DB2 Optimizer) and the Real values (from the workload):

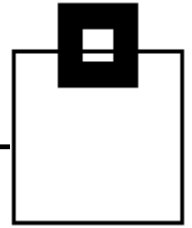
Estimated values		Real values	
Elapsed	5.269000	Elapsed	1.479829
Serviceunits	14967		
CPU *	0.374175	CPU	1.220524
* calculated (serviceunits / factor)			



The service units factor is a WLX default and set to 40,000.



Application Workload



The bottom view then contains the EXPLAIN data as stored in the WLX product EXPLAIN history tables:

Plan table	Statement table	Predicate table	Filter table	Detcost table		
Table creator	TBNAME	ACCESSTYPE_DESC	ACCESSCREATOR	ACCESSNAME	MATCHCOLS	
SYSIBM	SYSINDEXSPACESTATS	Table space scan			0	
SYSIBM	SYSDATABASE	Index	SYSIBM	DSNDDH01	1	
SYSIBM	SYSINDEXES	Index	SYSIBM	DSNDXX01	2	
SYSIBM	SYSTABLES	Index	SYSIBM	DSNDTX01	2	
DSNACC	EXCEPT_TBL	Table space scan			0	

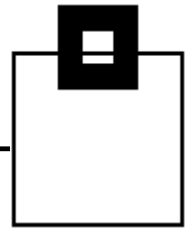
METHOD	METHOD_DESC	INDEXONLY	SORTN	SORTC	PREFETCH
0		N			Pure sequential
1	Nested loop join	N		JOIN	Unknown
0		N			Unknown
1	Nested loop join	N			Unknown
1	Nested loop join	N			Pure sequential



And all the other columns as well! Naturally clicking on one of the other Tabs shows that data as well...



Application Workload

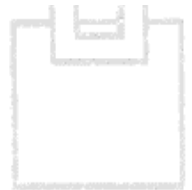


Plan table	Statement table	Predicate table	Filter table	Detcost table					
GROUP_MEMBER	EXPLAIN_TIME	Statement Type		COST_CATEGORY	PROCMS	PROCSU	REASON	STMT_ENCODE	TOTAL_COST
	2015-12-09-15.30.45.160000	SELECT		A	5,269	14,967		M	62,101.32911777496

Plan table	Statement table	Predicate table	Filter table	Detcost table			
QBLOCKNO	PREDNO	PREDICATE_TYPE	LEFT_HAND_SIDE	LEFT_HAND_PNO	LHS_TABNO	LHS_QBNO	RIGHT_HAND_SID
1	1	AND		0	0	0	
1	2	OR		0	0	0	
1	3	RANGE	COLEXP	0	0	0	VALUE
1	4	RANGE	COLEXP	0	0	0	VALUE

Plan table	Statement table	Predicate table	Filter table	Detcost table			
QBLOCKNO	PLANNO	ORDERNO	PREDNO	STAGE	ORDERCLASS	EXPLAIN_TIME	MIXOPSEQNO
2	2	1	28	MATCHING	-1	2015-12-09-15.30.45.160000	0
2	2	2	29	MATCHING	-1	2015-12-09-15.30.45.160000	0
2	3	1	30	STAGE1	-1	2015-12-09-15.30.45.160000	0

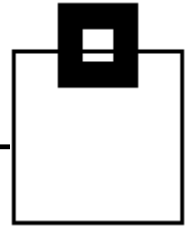
Plan table	Statement table	Predicate table	Filter table	Detcost table				
QBLOCKNO	PLANNO	OPENIO	OPENCPU	OPENCOST	DMIO	DMCPU	DMTOT	SUBQIO
1	1	0	0	0	81.1999969	8,223,382	451.862793	0
1	2	0	0	0	0.0172163	3,283.3891602	0.1500078	0
2	1	0	0	0	1.1799402	15,384.578125	2.9014168	0



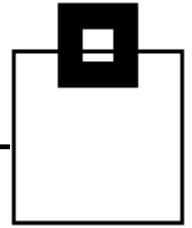
Application Workload

Now this data is the data that is optionally saved when running WLX. What if you would like to force an EXPLAIN and then display it in Visual Explain?

Now you can!



Application Workload



On the left hand side of the new View you see the Technical User Id and Password

The Database connection and Port as well as IP Address and Scheme for blank qualifiers

Finally the Web Server internal Port that is by default 56788

Choose connection and then push the explain button

User	
Id	boxwell
Password	••••••••

Database	
Name	ZD00QB1A
Port	5103
Host/IP	192.168.222.10
Schema	MVNXTEST

Web Server	
Port	56788



Application Workload

On the right you get your Visual Explain display!

The screenshot displays a database query execution interface. On the left, a 'Properties' table lists various query metrics. On the right, a 'Visual Explain' diagram shows the query execution plan with a tooltip for the selected query block.

Properties Table:

Name	Value
Output Cardinality	1,353
Query Block Number	1
Type	SELECT
Context	TOP LEVEL
Parent Query Block Number	0
Runtime Parent Query Block Number	-1
Do at Open	N
Times	1,354
Cost Information	
Cumulative Total Cost	1,742.33
Cumulative I/O Cost	491.84
Cumulative CPU Cost	35,525,344.00

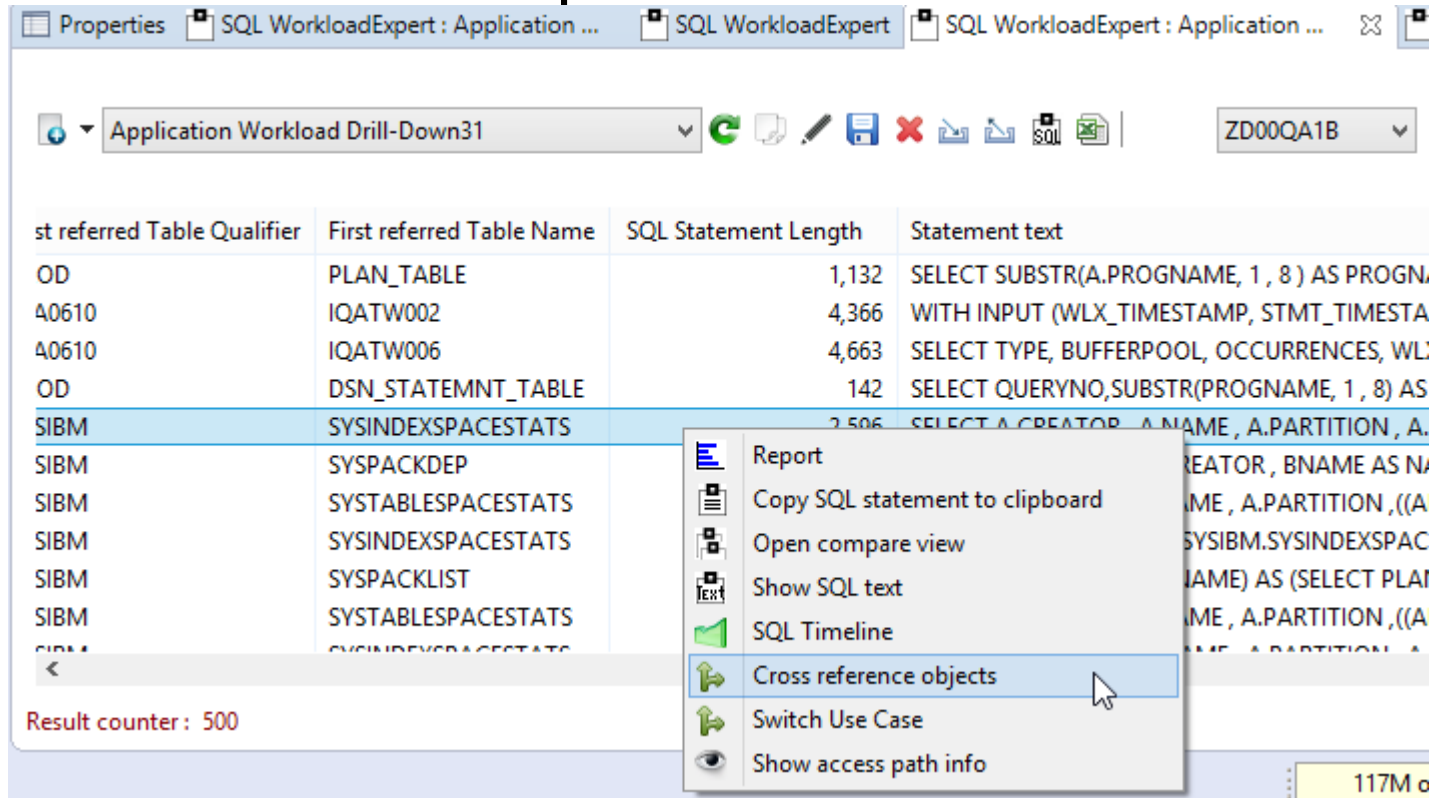
Visual Explain Plan:

The plan shows a sequence of operations: (1) QUERY, (2) QB1 (selected), (3) WFSCAN, (4) WKFILE, (5) SORT, (6) WFSCAN, and (7) T1. A tooltip for the selected node (2) QB1 provides the following details:

Node Type	Query block
Name	QB1
Cardinality	1,353
Total Cost	1,742.33
CPU Cost	35,525,344.00
I/O Cost	491.84

Application Workload

Another new feature is the ability to view cross reference data. If you are on an SQL in WLX and you click on the context menu option:

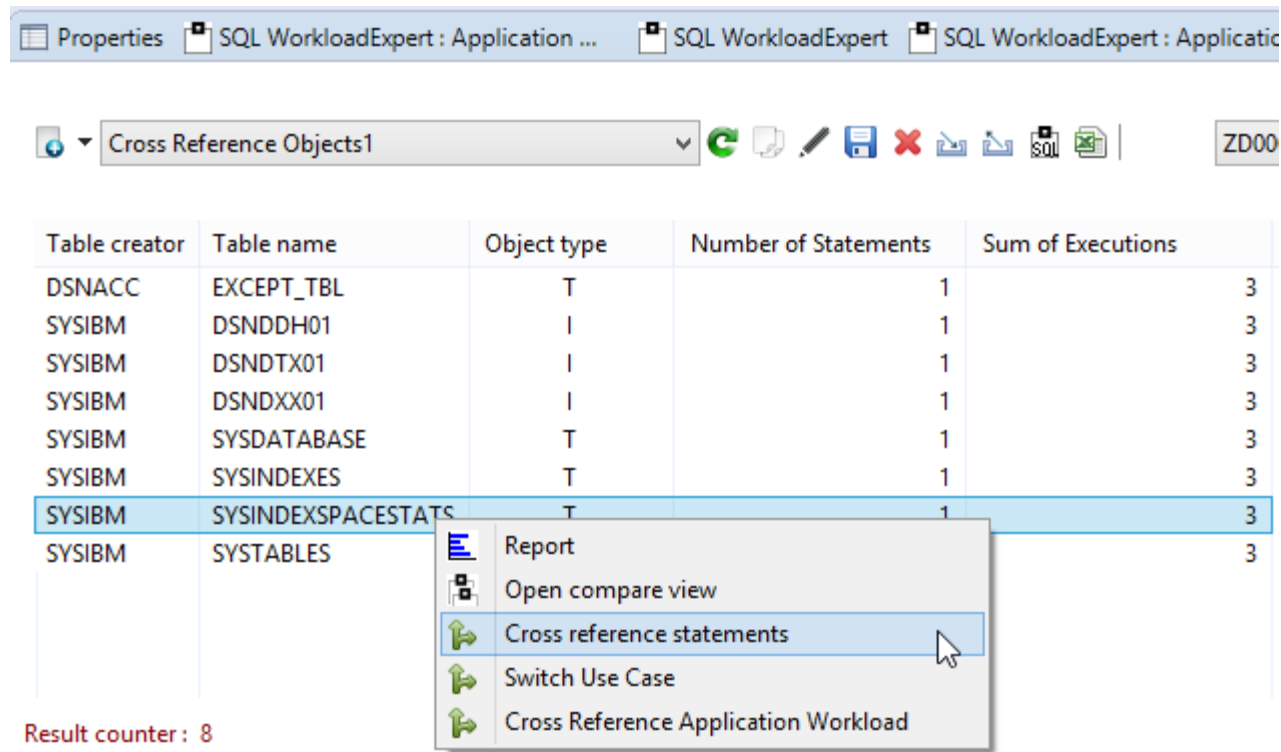


The screenshot shows the SQL WorkloadExpert application window. The main table displays SQL statements with columns: 'st referred Table Qualifier', 'First referred Table Name', 'SQL Statement Length', and 'Statement text'. The row for 'SYSINDEXSPACESTATS' is selected. A context menu is open over this row, showing options: 'Report', 'Copy SQL statement to clipboard', 'Open compare view', 'Show SQL text', 'SQL Timeline', 'Cross reference objects' (highlighted), 'Switch Use Case', and 'Show access path info'. The 'Result counter' at the bottom left shows '500'.

st referred Table Qualifier	First referred Table Name	SQL Statement Length	Statement text
OD	PLAN_TABLE	1,132	SELECT SUBSTR(A.PROGNAME, 1, 8) AS PROGNAME
A0610	IQATW002	4,366	WITH INPUT (WLX_TIMESTAMP, STMT_TIMESTAMP)
A0610	IQATW006	4,663	SELECT TYPE, BUFFERPOOL, OCCURRENCES, WLX_TIMESTAMP
OD	DSN_STATEMENT_TABLE	142	SELECT QUERYNO, SUBSTR(PROGNAME, 1, 8) AS PROGNAME
SIBM	SYSINDEXSPACESTATS	2,506	SELECT A.CREATOR, A.NAME, A.PARTITION, A...
SIBM	SYSPACKDEP		CREATOR, BNAME AS NAME
SIBM	SYSTABLESPACESTATS		NAME, A.PARTITION, ((A...
SIBM	SYSINDEXSPACESTATS		SYSIBM.SYSINDEXSPACE...
SIBM	SYSPACKLIST		NAME) AS (SELECT PLAN...
SIBM	SYSTABLESPACESTATS		NAME, A.PARTITION, ((A...
SIBM	SYSINDEXSPACESTATS		NAME, A.PARTITION, A...

Application Workload

You see the list of objects in that SQL and using right-click again you get another context menu where you can review all the statements that referenced that object:



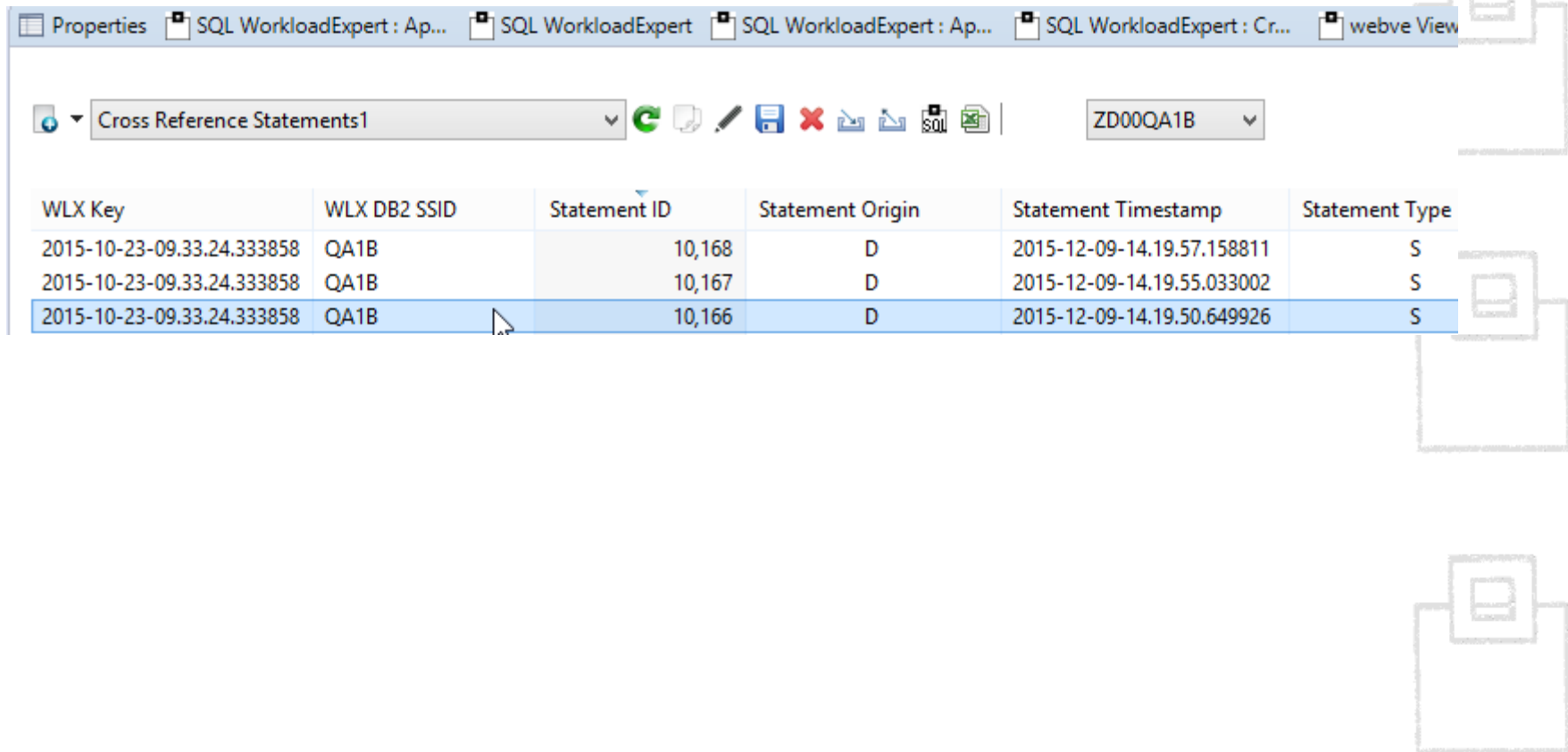
The screenshot shows the SQL WorkloadExpert interface. At the top, there are tabs for 'Properties', 'SQL WorkloadExpert : Application ...', 'SQL WorkloadExpert', and 'SQL WorkloadExpert : Applicatio...'. Below the tabs is a toolbar with various icons, including a green 'C' icon, a document icon, a pencil icon, a save icon, a delete icon, a print icon, a SQL icon, and a refresh icon. A dropdown menu is open, showing 'Cross Reference Objects1' and 'ZD00'. Below the toolbar is a table with the following columns: 'Table creator', 'Table name', 'Object type', 'Number of Statements', and 'Sum of Executions'. The table contains the following data:

Table creator	Table name	Object type	Number of Statements	Sum of Executions
DSNACC	EXCEPT_TBL	T	1	3
SYSIBM	DSNDDH01	I	1	3
SYSIBM	DSNDTX01	I	1	3
SYSIBM	DSNDXX01	I	1	3
SYSIBM	SYSDATABASE	T	1	3
SYSIBM	SYSINDEXES	T	1	3
SYSIBM	SYSINDEXSPACESTATS	T	1	3
SYSIBM	SYSTABLES			3

A context menu is open over the 'SYSTABLES' row, showing the following options: 'Report', 'Open compare view', 'Cross reference statements' (highlighted), 'Switch Use Case', and 'Cross Reference Application Workload'. At the bottom left, there is a red text label 'Result counter : 8'.

Application Workload

You see the list of statements that referred to that object:

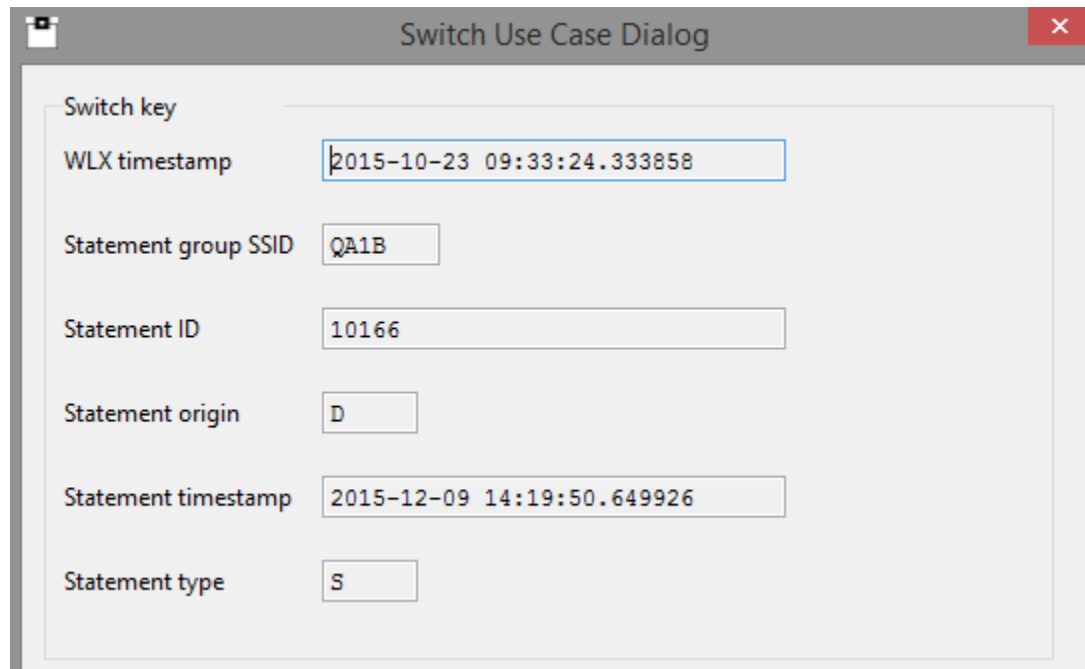


The screenshot shows the SQL WorkloadExpert application window. The title bar includes tabs for 'Properties', 'SQL WorkloadExpert : Ap...', 'SQL WorkloadExpert', 'SQL WorkloadExpert : Ap...', 'SQL WorkloadExpert : Cr...', and 'webve View'. The main window has a toolbar with icons for file operations and a dropdown menu set to 'Cross Reference Statements1'. A search filter 'ZD00QA1B' is entered in a text box. Below the toolbar is a table with six columns: 'WLX Key', 'WLX DB2 SSID', 'Statement ID', 'Statement Origin', 'Statement Timestamp', and 'Statement Type'. The table contains three rows of data, with the third row highlighted in blue.

WLX Key	WLX DB2 SSID	Statement ID	Statement Origin	Statement Timestamp	Statement Type
2015-10-23-09.33.24.333858	QA1B	10,168	D	2015-12-09-14.19.57.158811	S
2015-10-23-09.33.24.333858	QA1B	10,167	D	2015-12-09-14.19.55.033002	S
2015-10-23-09.33.24.333858	QA1B	10,166	D	2015-12-09-14.19.50.649926	S

Application Workload

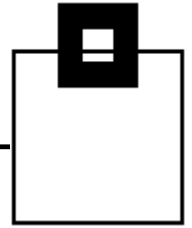
The last new function in the context menu is the “Switch Use Case”, when selected you first get a new panel where the top half contains the current “key” data:



A screenshot of a software dialog box titled "Switch Use Case Dialog". The dialog contains several input fields for configuration. The fields and their values are as follows:

Field	Value
Switch key	
WLX timestamp	2015-10-23 09:33:24.333858
Statement group SSID	QA1B
Statement ID	10166
Statement origin	D
Statement timestamp	2015-12-09 14:19:50.649926
Statement type	S

Application Workload



The bottom half contains the available Use Cases to switch to:

Choose use case

- ☒ Application Workload
- ☐ Delay detection
- ☐ Up and Down scaling
- ☐ Same SQL / mult. schemas
- ☐ Multi-row Fetch detect
- ☐ Cross Reference Statements
- ☐ Audit
- ☐ SQL INTENTs
- ☐ Object Update Dynamic
- ☐ SYSADM object updates
- ☐ Disk I/O
- ☐ CPU usage of packages

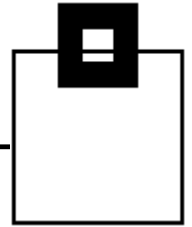
OK Cancel



The idea here is to enable fast problem discovery.



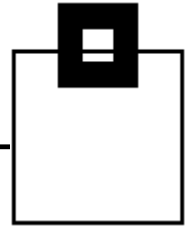
AGENDA



1. WLX - How it works
2. Bufferpool Usage: Who's been using my pool?
3. PKLIST Chains: Are your Packages bound correctly?
4. Application Workload: New functionality added!
5. **SQL Time Travel: Follow your SQLs through time**
6. Q&A Session



SQL Time Travel



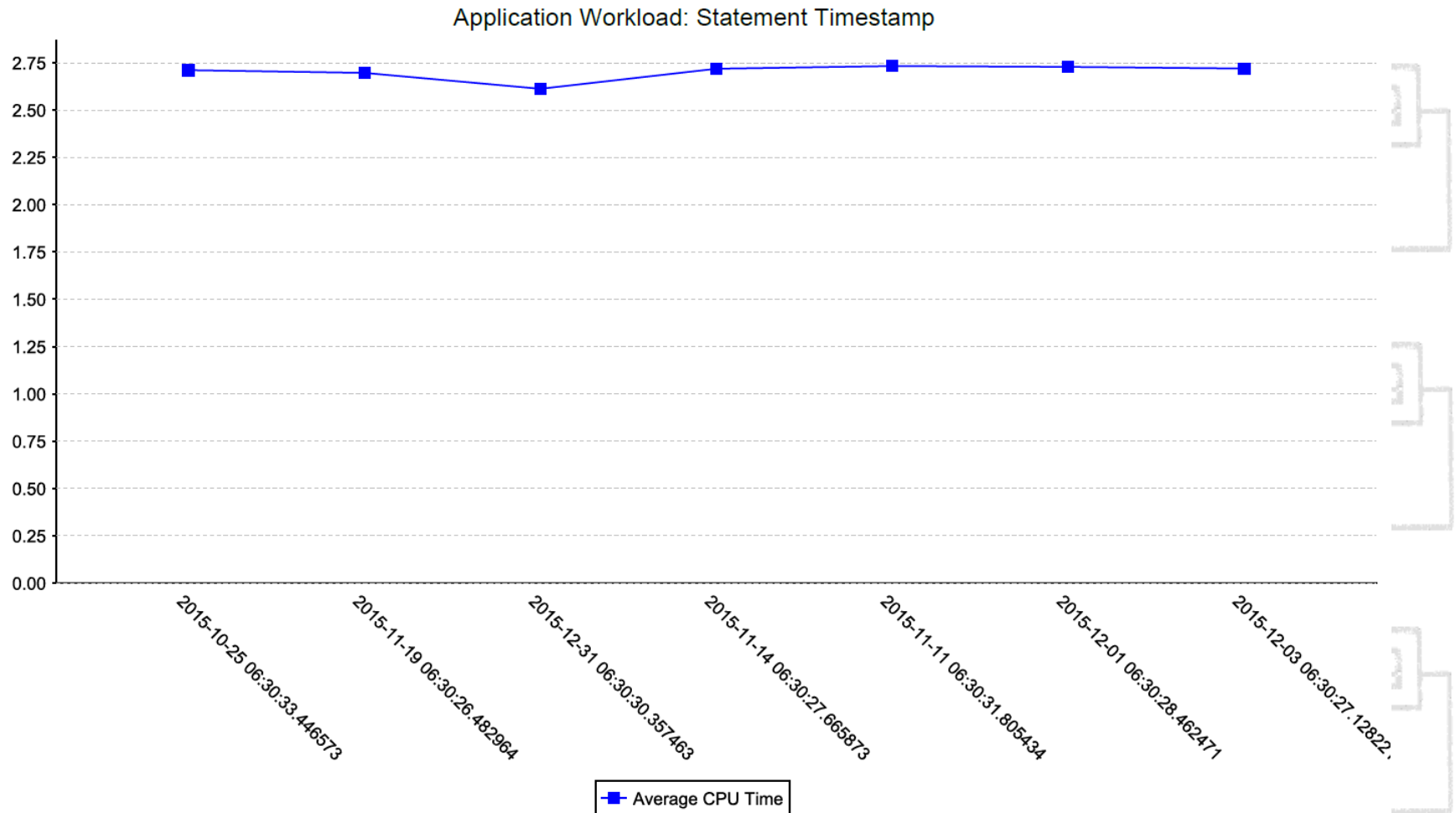
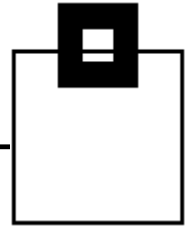
Ever wondered how your SQL performed in the past?
Now you can find out!

The screenshot shows the SQL WorkloadExpert interface. At the top, there are tabs for 'Properties' and 'SQL WorkloadExpert : Application Workload Drill-Down41'. Below the tabs is a toolbar with various icons and a dropdown menu showing 'Application Workload Drill-Down41'. To the right of the toolbar is a dropdown menu showing 'ZD00QA1B'. Below the toolbar is a table with the following columns: 'Collection ID', 'Executions', 'Average CPU Time', 'CPU Time per hour', 'Elapsed Time', and 'Average Elapsed Time'. The table contains seven rows of data for 'RTDX0510_PTFTOOL'. A context menu is open over the table, showing options: 'Report', 'Copy SQL statement to clipboard', 'Open compare view', 'Show SQL text', 'SQL Timeline' (highlighted), 'Cross reference objects', 'Switch Use Case', and 'Show access path info'.

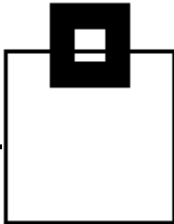
Collection ID	Executions	Average CPU Time	CPU Time per hour	Elapsed Time	Average Elapsed Time
RTDX0510_PTFTOOL	17	2.710857	0.120000	59.542477	3.443733
RTDX0510_PTFTOOL	8	2.696468			3.680416
RTDX0510_PTFTOOL	6	2.612292			3.125418
RTDX0510_PTFTOOL	5	2.718348			3.656145
RTDX0510_PTFTOOL	3	2.732516			3.917540
RTDX0510_PTFTOOL	2	2.728057			3.917152
RTDX0510_PTFTOOL	2	2.719423			3.980670

Bring up the context menu after selecting your time ranges

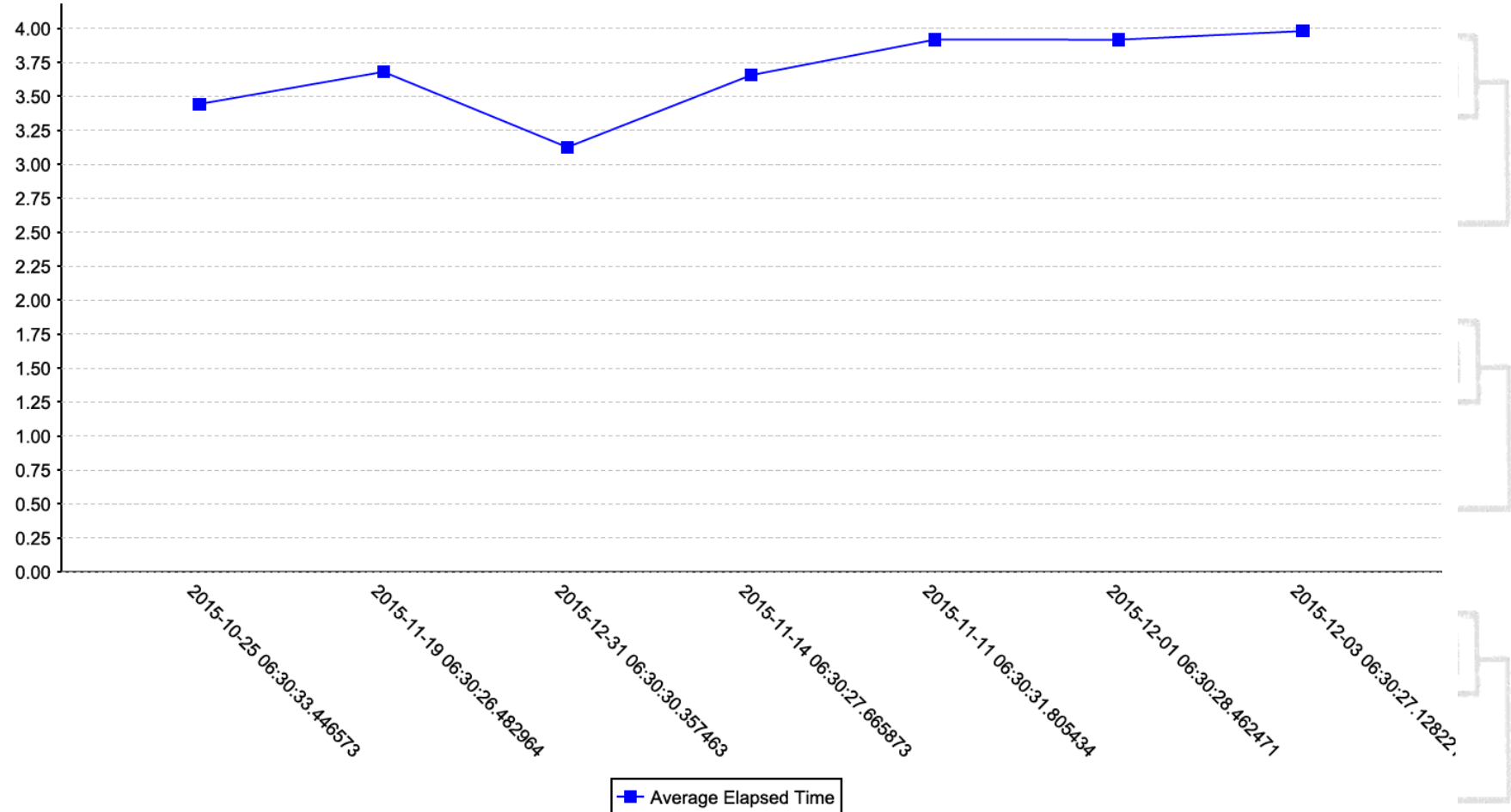
SQL Time Travel



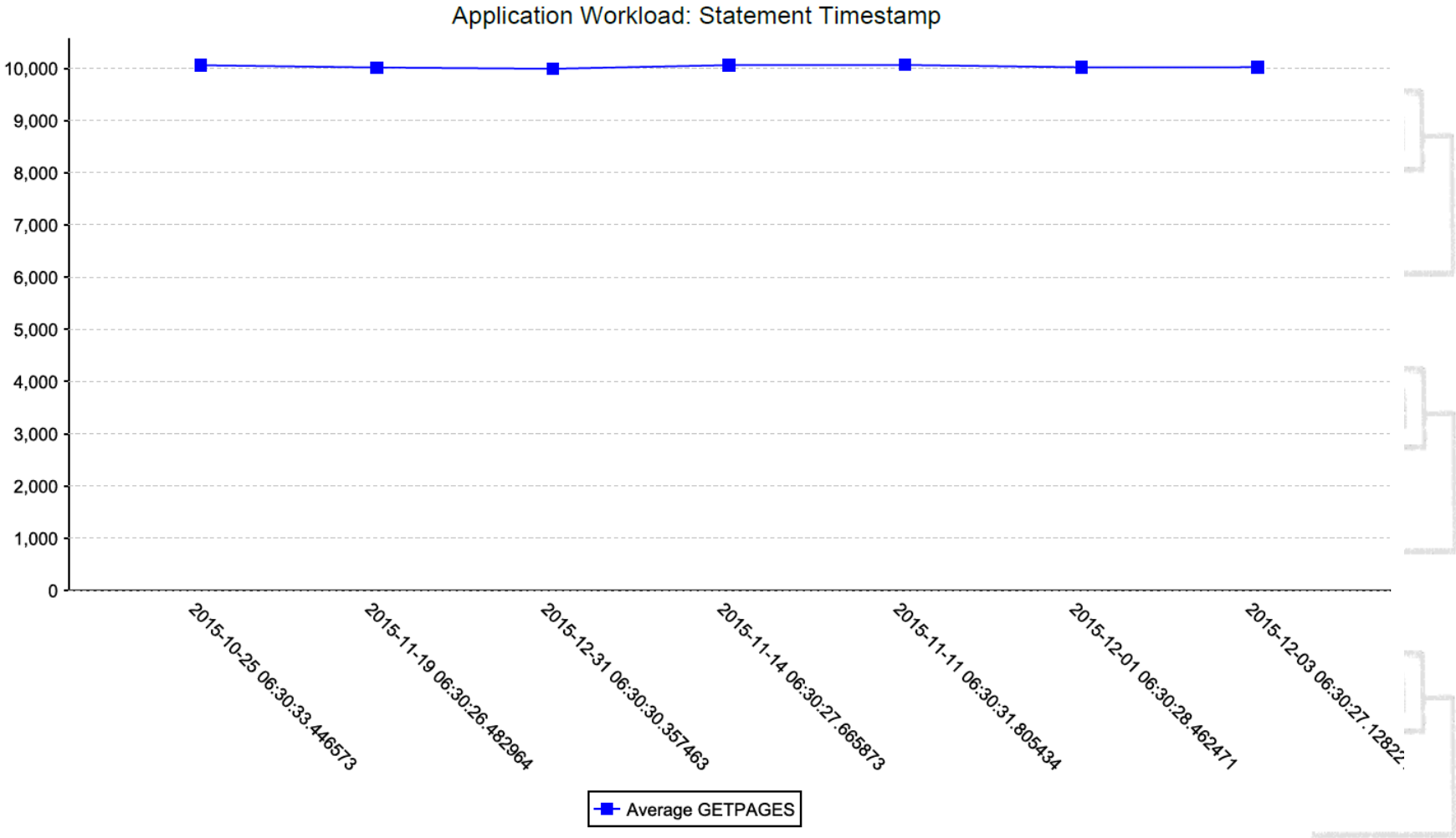
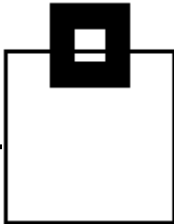
SQL Time Travel



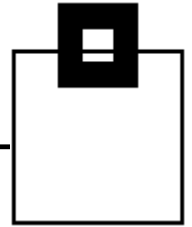
Application Workload: Statement Timestamp



SQL Time Travel

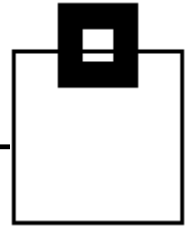


AGENDA



1. WLX - How it works
2. Bufferpool Usage: Who's been using my pool?
3. PKLIST Chains: Are your Packages bound correctly?
4. Application Workload: New functionality added!
5. SQL Time Travel: Follow your SQLs through time
6. Q&A Session





Thank you for listening!



If you want more info then please subscribe to my newsletter by going to www.seg.de or www.segus.com

Roy Boxwell
SOFTWARE ENGINEERING GmbH
&
SEGUS Inc.

R.Boxwell@seg.de

