softwareonz

...a better way on z

DB2 Access Paths

Surviving and Thriving

The DB2 Education Webinar Series by Craig S. Mullins and SoftwareOnZ – Part 1

Agenda

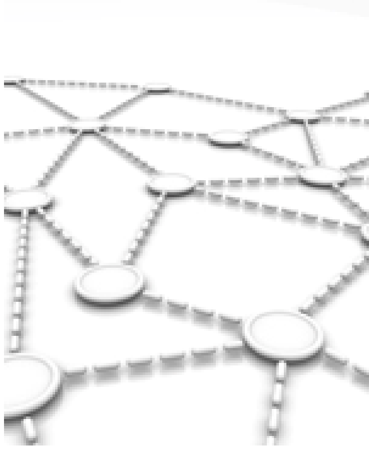
- BIND and REBIND Essentials
  - Recent BIND advances in DB2 9
- Access Paths and Change Management
  - Lack of Control
- Version Management
  - Version Migration Issues
  - Understand and Prepare!
- New Product for Managing BINDs
  - zAPX Overview and Walkthru

BIND / REBIND Review

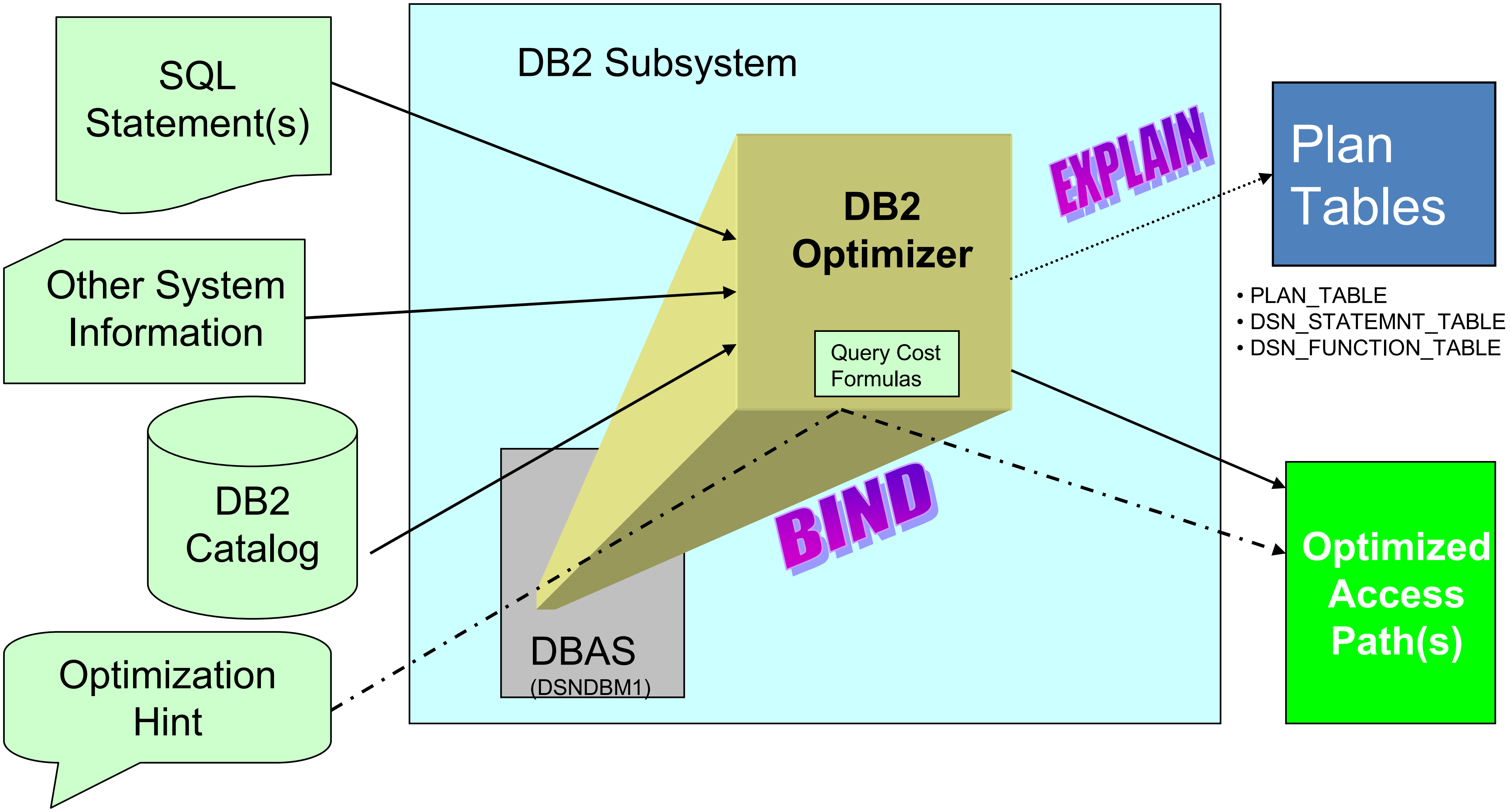
- The BIND and REBIND commands:
  - Are used to create DB2 plans and packages
  - Many options to choose from, including:
    - DEGREE (ANY | 1)
    - EXPLAIN (YES | NO)
    - ISOLATION (RR | RS | CS | UR | NC)
    - OPTHINT(id)
    - ACQUIRE (USE | ALLOCATE)
    - RELEASE (COMMIT | DEALLOCATE)
    - VALIDATE (RUN | BIND)
    - CURRENTDATA (YES | NO)
    - and more…

BIND/REBIND Review (continued)

- BIND and REBIND are critical for application performance
- It is a wise course of action to plan your REBIND strategy
- BIND reads DBRM and converts SQL to access paths
  - SQL can change
- REBIND re-evaluates the access paths of a pre-existing plan or package
  - SQL cannot change



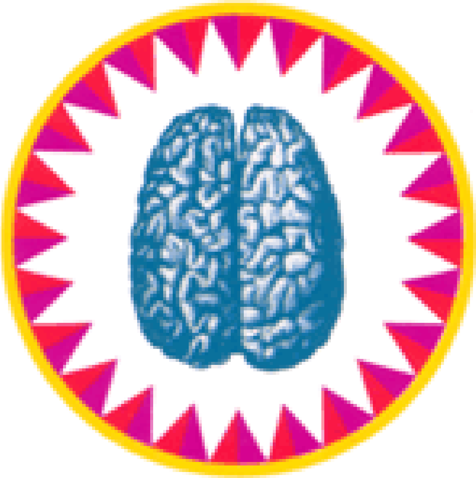
BIND and Optimization



- PLAN_TABLE
- DSN_STATEMNT_TABLE
- DSN_FUNCTION_TABLE

BIND / REBIND Accesses…

- SQL
- BIND Parameters
- DB2 Catalog
  - Database Object Information
    - Table Columns
    - Table Space LOCKSIZE
  - Database Statistics
- System Information
  - CPU
  - System Software
- DB2 Subsystem Information
  - RID Pool Parameters - SIZE
  - Buffer Pool Parameters – VPPSEQT
- Query Cost Formulas – in the Optimizer code



OK, So What?

- We know that:
  - BIND and REBIND are critical for application performance
  - It is a wise course of action to plan your REBIND strategy
- There are several common approaches:
  - Daily maintenance: REBIND after RUNSTATS
    - Perhaps not every day, but REBIND are done after RUNSTATS
  - Global REBIND after migration to new DB2 version
  - Global REBIND after installing new PTFs
    - Above two mean access paths only change when DB2 changes
  - REBIND after *x* days / weeks / months …
  - Let it Ride! ("If it ain't broke, don't fix it.")

Let It Ride

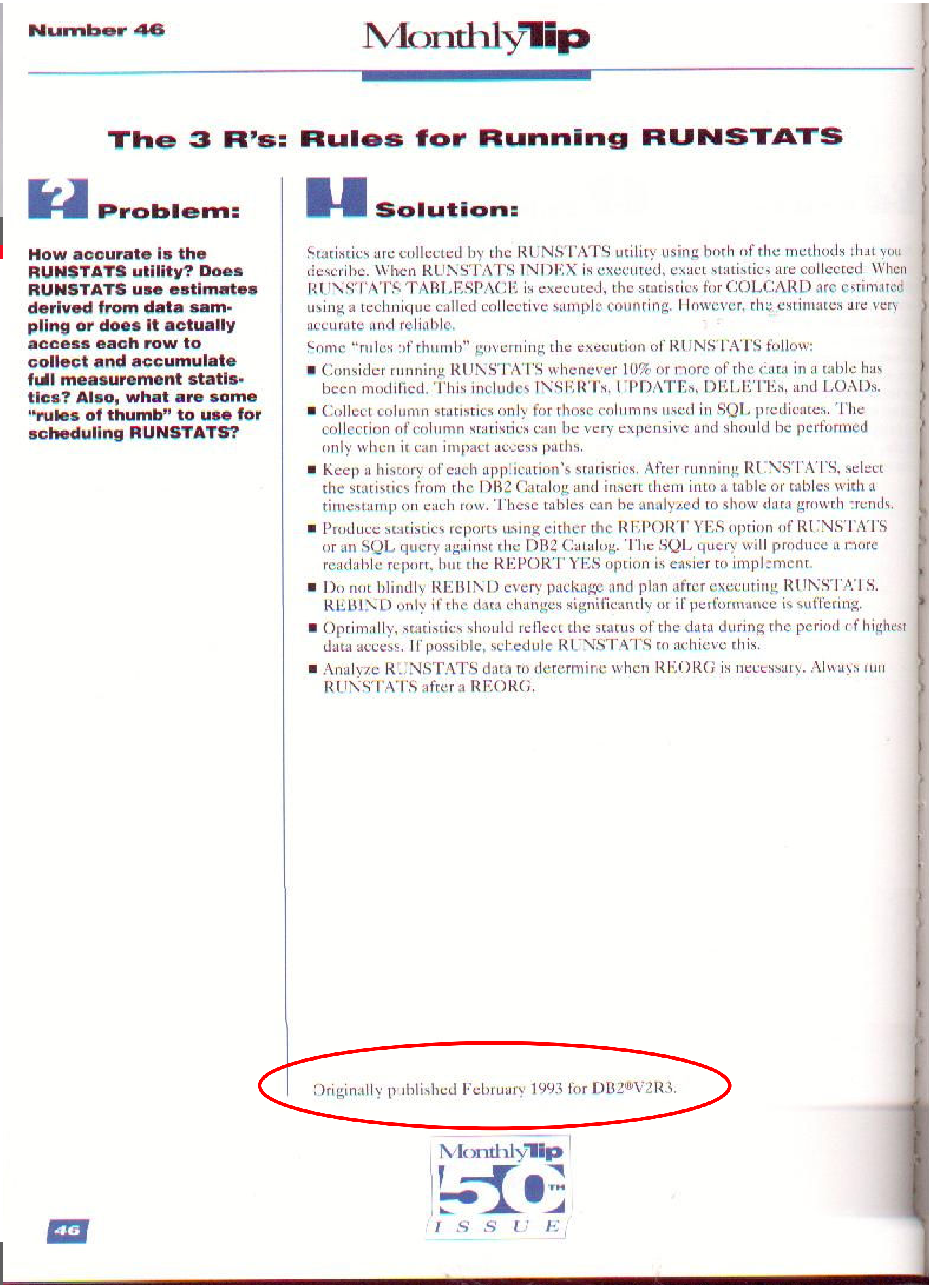
- Programs once bound, are (almost) never rebound.
- Reason:
  - Fear of access path degradation
- Result:
  - No improvement to access paths
  - No CPU savings from new DB2 efficiencies
  - Sub-optimal performance
  - Every DB2 program potentially suffers for fear that one or two SQL statements will become inefficient

Regular REBIND

- **Better Approach**: Regular REBINDing
  - The Three R's *(next slide)*
- Reason:
  - Access paths are more up-to-date based on the current state of the data.
- Result:
  - Generally, improved access paths
  - CPU savings from new DB2 efficiencies
  - Optimal performance
- Of course, you can still get those "problem" access paths.

The Three R's

- REORG
- RUNSTATS
- REBIND



Number 46 MonthlyTip

The 3 R's: Rules for Running RUNSTATS

Problem:

How accurate is the RUNSTATS utility? Does RUNSTATS use estimates derived from data sampling or does it actually access each row to collect and accumulate full measurement statistics? Also, what are some "rules of thumb" to use for scheduling RUNSTATS?

Solution:

Statistics are collected by the RUNSTATS utility using both of the methods that you describe. When RUNSTATS INDEX is executed, exact statistics are collected. When RUNSTATS TABLESPACE is executed, the statistics for COLCARD are estimated using a technique called collective sample counting. However, the estimates are very accurate and reliable.

Some "rules of thumb" governing the execution of RUNSTATS follow:

- Consider running RUNSTATS whenever 10% or more of the data in a table has been modified. This includes INSERTs, UPDATEs, DELETEs, and LOADs.
- Collect column statistics only for those columns used in SQL predicates. The collection of column statistics can be very expensive and should be performed only when it can impact access paths.
- Keep a history of each application's statistics. After running RUNSTATS, select the statistics from the DB2 Catalog and insert them into a table or tables with a timestamp on each row. These tables can be analyzed to show data growth trends.
- Produce statistics reports using either the REPORT YES option of RUNSTATS or an SQL query against the DB2 Catalog. The SQL query will produce a more readable report, but the REPORT YES option is easier to implement.
- Do not blindly REBIND every package and plan after executing RUNSTATS. REBIND only if the data changes significantly or if performance is suffering.
- Optimally, statistics should reflect the status of the data during the period of highest data access. If possible, schedule RUNSTATS to achieve this.
- Analyze RUNSTATS data to determine when REORG is necessary. Always run RUNSTATS after a REORG.

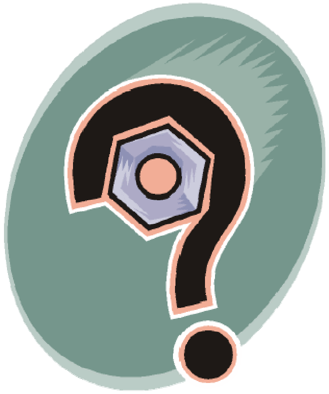
Originally published February 1993 for DB2®V2R3.

MonthlyTip 50TH ISSUE

46

Problems With the Three R's

- They pose a lot of questions…
  - When should you REORGanize?
    - To properly determine requires RUNSTATS (or RTS).
    - So should it be RUNSTATS, REORG, RUNSTATS, REBIND?
  - When should you run RUNSTATS?
    - To properly determine you need to know the make-up, usage, and volatility of your data.
  - When should you REBIND?
    - When statistics have changed significantly enough to change access paths.
    - But…



Catalog Statistics

- Why correct statistics are so important
  - The DB2 Optimizer makes all access path decisions
  - Accurate data helps the Optimizer make the correct decisions
  - Incorrect statistics may cause:
    - Less efficient join sequence
    - Less efficient method of accessing individual tables (e.g. sync I/O instead of prefetch)
    - Wrong or no index may be used
- According to Terry Purcell (IBM)
  - "As much as 50% of all bad access paths are caused by incorrect statistics."

Getting Correct Statistics

- Ways to update statistics
  - RUNSTATS utility
  - REORG with inline statistics
  - LOAD with inline statistics
  - Using SQL for statistics manipulation
  - Transferring statistics from another system
  - Using tools for manipulation

OK, so When Should we REBIND?

- When do we REBIND?
  - The best answer to this questions is:
    - "Whenever data has changed significantly enough that it may impact the performance of the existing access paths."
      - **The problem is knowing exactly when this happens.**
  - DB2 application performance can be negatively affected by uncontrolled REBINDs.
  - Causes
    - Optimizer inefficiency
    - Volatile tables
    - Catalog pollution
    - Inefficient use of RUNSTATS

Reviewing the Steps: The ~~3~~ 5 R’s

- RUNSTATS (or RTS)
- REORG
- RUNSTATS
- REBIND
- Recheck
  - In other words, what did the REBIND do?
    - Access path changes – better or worse?

How Do We Do This?

- How do you determine what access paths have changed?
  - Program changes?
  - Just access paths changes?
- Do you evaluate every program that is rebound in production?
  - Or do you just wait for irate users to call?

We will answer these questions...

The Latest in BIND from DB2

Access Path Stability

- **Access Path Stability**, which works on packages only, uses the PLANMGMT parameter to keep backups versions of your program's access paths.
- Why?
  - Because sometimes, after rebinding your program, performance degrades.
  - With plan stability you can fall back to a previous package.

PLANMGMT BIND Options

Previous and active copies of package.

- PLANMGMT(OFF) - No change to existing behavior. A package continues to have one active copy.
- PLANMGMT(BASIC) - A package has one active copy. One additional prior copy (PREVIOUS) is preserved.
- PLANMGMT(EXTENDED) - A package has one active copy, and two additional prior copies (PREVIOUS and ORIGINAL) are preserved.

Original, previous and active copies of package.

The Ol' Switcheroo

- SWITCH (PREVIOUS) - changes the current and previous packages:
  - The existing current package takes the place of the previous package.
  - The existing previous package takes the place of the current package.

Only if you bound using PLANMGMT EXTENDED (refer to previous slide).

- SWITCH (ORIGINAL) - clones the original copy to take the place of the current copy:
  - The existing current copy replaces the previous copy.
  - The existing previous copy is discarded.

Change Management

- Your Mainframe Environment Requires Strict Change Control Procedures
  - Application Program Changes
  - Database Changes
  - System Software Changes
  - DB2 Subsystem Changes
- But what about Access Path changes?

When Are Access Paths Changed?

- Any time the program changes
  - BIND is required
  - Unless the SQL does not change and you have a tool to manage the process
- Every time we REBIND whether the program changed or not
- New DB2 Releases and Versions
  - Sometimes we must REBIND to get SQL performance improvements
  - But performance gains are ***not*** guaranteed

Version Migration Concerns

- Typically, you do not HAVE to REBIND all of your packages and plans when you move to a new version of DB2.
  - However, it is a really good idea.
- There are a lot of optimizer enhancements and performance improvements that you won't get without a REBIND.
- And there are some REBINDs you cannot avoid.
  - For DB2 9, plans and packages from DB2 V4 or earlier will be automatically rebound when accessed by DB2 9.

Access Path Change Management

- Given the issues, features, and migration concerns, what do we need going forward?
  - A way to predict the performance of SQL before REBINDing in production
    - Both Static and Dynamic SQL
    - Preview the access paths resulting from execution of BIND commands
    - Show before and after access paths
      - Predicting better/worse/same
  - A way to analyze and determine the impact of REBINDing before:
    - DB2 version migrations
    - Major PTFs
  - A way to integrate BIND/REBIND into existing change control processes
    - Avoid BINDs where performance will deterioriate
    - Identify new statements
      - Show the access path for new statements

softwareonz

...a better way on z

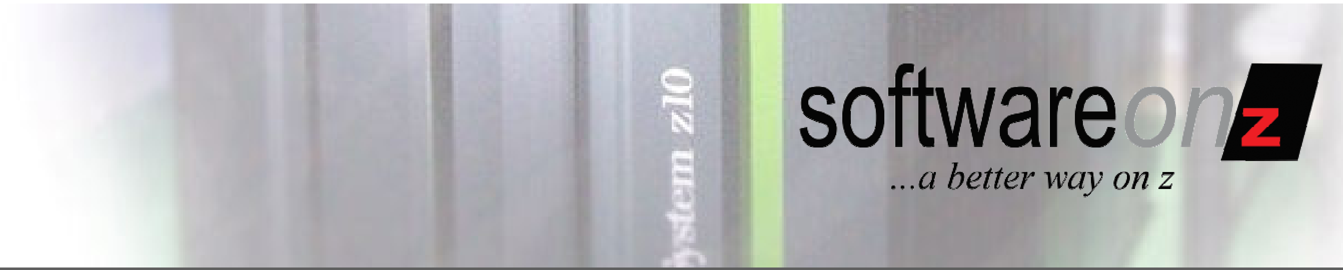
Introducing

APX

New Generation SQL Access Path Rating & Comparison

Z Systems
Database Solutions GmbH

Access Path Expert



Explain
Explain & Prototyping

Batch
Batch Comparison Reults

Worklist
Show Worklist

DSC
Show Dynamic Statement Cache

DB2 Catalog
DB2 Catalog Reference & Browser

SQL
Execute SQL

DB2 Packages
Search DB2 Packages and History

DB2 Objects
Search DB2 Objects

Highlights

- Graphical User Interface
- Single Point of Control for all DB2 systems
- Explain
- Automation
- DB2 Catalog Management

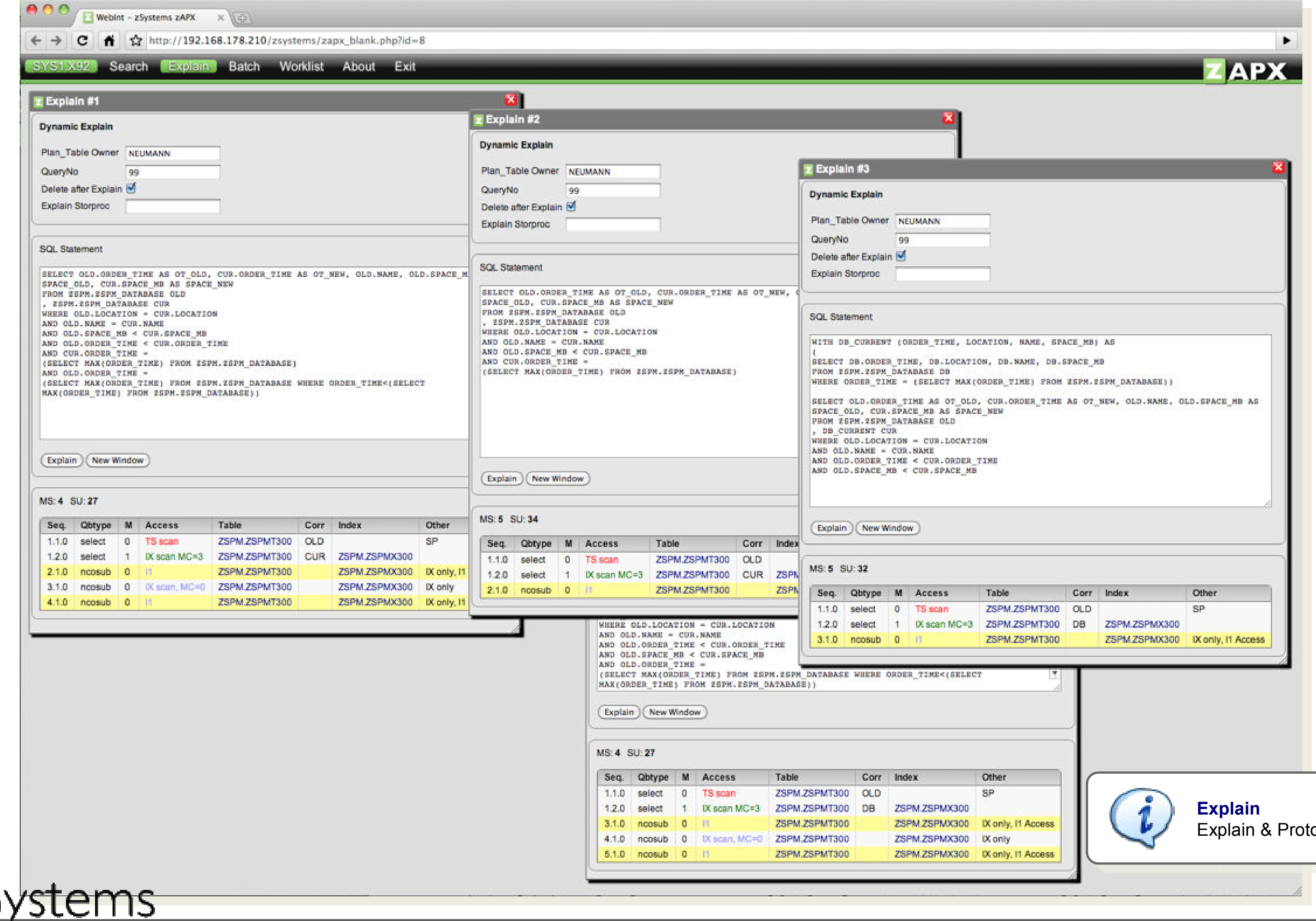
EXPLAIN

Explain
Explain & Prototyping

- Explain with controlled product authority
- Dynamic highlighting of Access Path Differences
- Static and dynamic SQL
- Correlates Views, Aliases and Synonyms to Plan_Table

Explain & Compare

softwareonz
...a better way on z



Explain #1

Dynamic Explain

Plan_Table Owner: NEUMANN
QueryNo: 99
Delete after Explain ☑
Explain Storproc:

SQL Statement

```
SELECT OLD.ORDER_TIME AS OT_OLD, CUR.ORDER_TIME AS OT_NEW, OLD.NAME, OLD.SPACE_M
SPACE_OLD, CUR.SPACE_MB AS SPACE_NEW
FROM ZSPM.ZSPM_DATABASE OLD
, ZSPM.ZSPM_DATABASE CUR
WHERE OLD.LOCATION = CUR.LOCATION
AND OLD.NAME = CUR.NAME
AND OLD.SPACE_MB < CUR.SPACE_MB
AND OLD.ORDER_TIME < CUR.ORDER_TIME
AND CUR.ORDER_TIME =
(SELECT MAX(ORDER_TIME) FROM ZSPM.ZSPM_DATABASE)
AND OLD.ORDER_TIME =
(SELECT MAX(ORDER_TIME) FROM ZSPM.ZSPM_DATABASE WHERE ORDER_TIME<(SELECT
MAX(ORDER_TIME) FROM ZSPM.ZSPM_DATABASE))
```

MS: 4 SU: **27**

Seq.	Qbtype	M	Access	Table	Corr	Index	Other
1.1.0	select	0	TS scan	ZSPM.ZSPMT300	OLD		SP
1.2.0	select	1	IX scan MC=3	ZSPM.ZSPMT300	CUR	ZSPM.ZSPMX300	
2.1.0	ncosub	0	I1	ZSPM.ZSPMT300		ZSPM.ZSPMX300	IX only, I1
3.1.0	ncosub	0	IX scan, MC=0	ZSPM.ZSPMT300		ZSPM.ZSPMX300	IX only
4.1.0	ncosub	0	I1	ZSPM.ZSPMT300		ZSPM.ZSPMX300	IX only, I1

Explain #2

Dynamic Explain

Plan_Table Owner: NEUMANN
QueryNo: 99
Delete after Explain ☑
Explain Storproc:

SQL Statement

```
SELECT OLD.ORDER_TIME AS OT_OLD, CUR.ORDER_TIME AS OT_NEW, C
SPACE_OLD, CUR.SPACE_MB AS SPACE_NEW
FROM ZSPM.ZSPM_DATABASE OLD
, ZSPM.ZSPM_DATABASE CUR
WHERE OLD.LOCATION = CUR.LOCATION
AND OLD.NAME = CUR.NAME
AND OLD.SPACE_MB < CUR.SPACE_MB
AND CUR.ORDER_TIME =
(SELECT MAX(ORDER_TIME) FROM ZSPM.ZSPM_DATABASE)
```

MS: 5 SU: **34**

Seq.	Qbtype	M	Access	Table	Corr	Index
1.1.0	select	0	TS scan	ZSPM.ZSPMT300	OLD	
1.2.0	select	1	IX scan MC=3	ZSPM.ZSPMT300	CUR	ZSPM
2.1.0	ncosub	0	I1	ZSPM.ZSPMT300		ZSPM

Explain #3

Dynamic Explain

Plan_Table Owner: NEUMANN
QueryNo: 99
Delete after Explain ☑
Explain Storproc:

SQL Statement

```
WITH DB_CURRENT (ORDER_TIME, LOCATION, NAME, SPACE_MB) AS
(
SELECT DB.ORDER_TIME, DB.LOCATION, DB.NAME, DB.SPACE_MB
FROM ZSPM.ZSPM_DATABASE DB
WHERE ORDER_TIME = (SELECT MAX(ORDER_TIME) FROM ZSPM.ZSPM_DATABASE))

SELECT OLD.ORDER_TIME AS OT_OLD, CUR.ORDER_TIME AS OT_NEW, OLD.NAME, OLD.SPACE_MB AS
SPACE_OLD, CUR.SPACE_MB AS SPACE_NEW
FROM ZSPM.ZSPM_DATABASE OLD
, DB_CURRENT CUR
WHERE OLD.LOCATION = CUR.LOCATION
AND OLD.NAME = CUR.NAME
AND OLD.ORDER_TIME < CUR.ORDER_TIME
AND OLD.SPACE_MB < CUR.SPACE_MB
```

MS: **5** SU: **32**

Seq.	Qbtype	M	Access	Table	Corr	Index	Other
1.1.0	select	0	TS scan	ZSPM.ZSPMT300	OLD		SP
1.2.0	select	1	IX scan MC=3	ZSPM.ZSPMT300	DB	ZSPM.ZSPMX300	
3.1.0	ncosub	0	I1	ZSPM.ZSPMT300		ZSPM.ZSPMX300	IX only, I1 Access

```
WHERE OLD.LOCATION = CUR.LOCATION
AND OLD.NAME = CUR.NAME
AND OLD.ORDER_TIME < CUR.ORDER_TIME
AND OLD.SPACE_MB < CUR.SPACE_MB
AND OLD.ORDER_TIME =
(SELECT MAX(ORDER_TIME) FROM ZSPM.ZSPM_DATABASE WHERE ORDER_TIME<(SELECT
MAX(ORDER_TIME) FROM ZSPM.ZSPM_DATABASE))
```

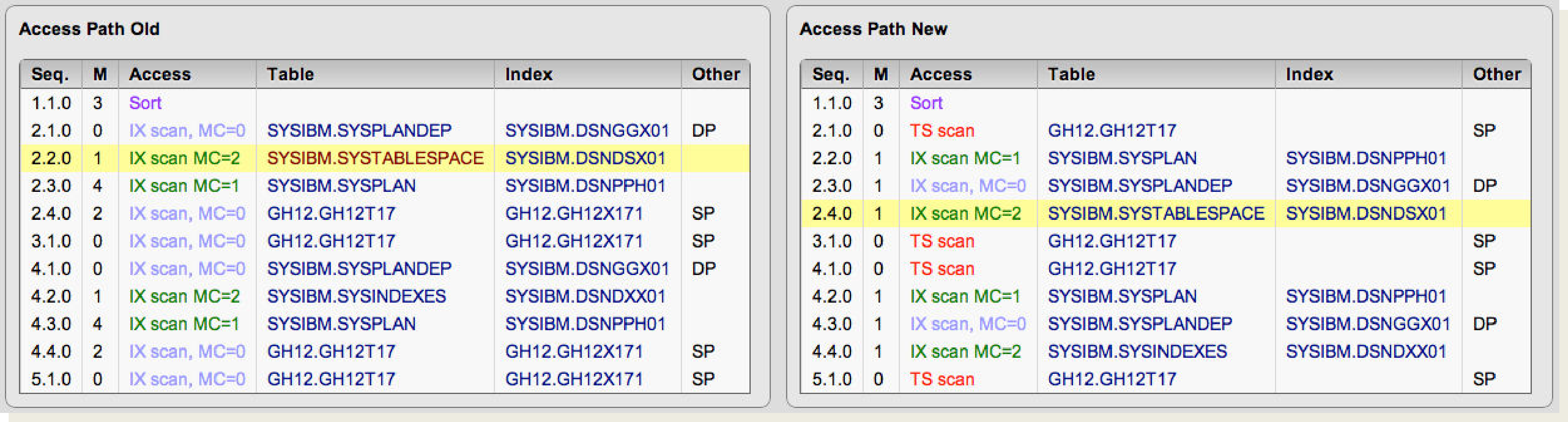
MS: 4 SU: **27**

Seq.	Qbtype	M	Access	Table	Corr	Index	Other
1.1.0	select	0	TS scan	ZSPM.ZSPMT300	OLD		SP
1.2.0	select	1	IX scan MC=3	ZSPM.ZSPMT300	DB	ZSPM.ZSPMX300	
3.1.0	ncosub	0	I1	ZSPM.ZSPMT300		ZSPM.ZSPMX300	IX only, I1 Access
4.1.0	ncosub	0	IX scan, MC=0	ZSPM.ZSPMT300		ZSPM.ZSPMX300	IX only
5.1.0	ncosub	0	I1	ZSPM.ZSPMT300		ZSPM.ZSPMX300	IX only, I1 Access

Explain
Explain & Prototyping

zSystems
Database Solutions GmbH

Difference Highlighting



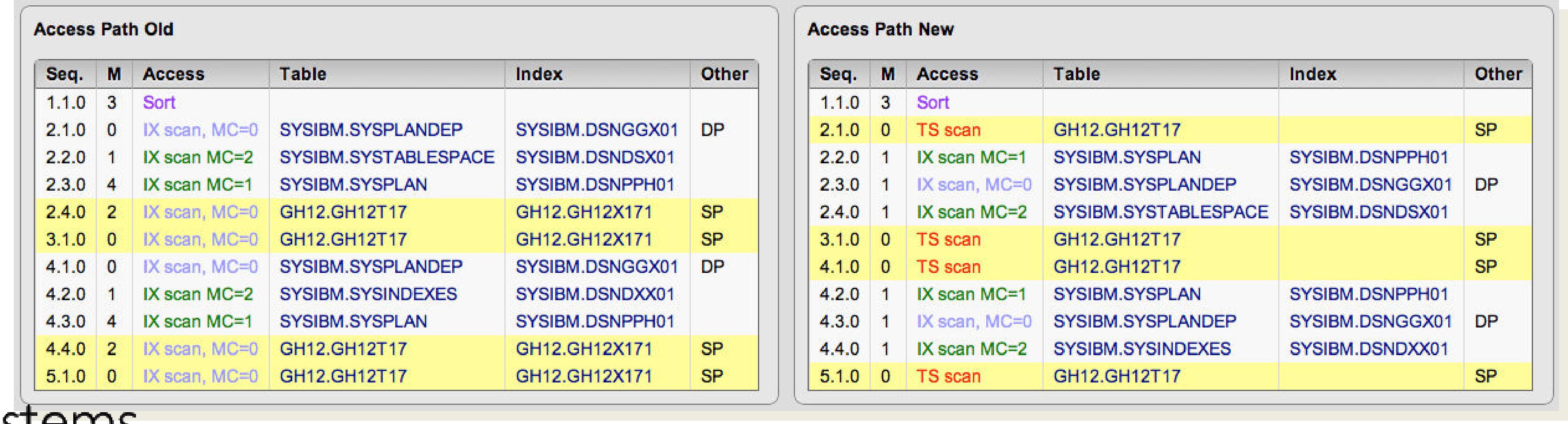
Access Path Old

Seq.	M	Access	Table	Index	Other
1.1.0	3	Sort			
2.1.0	0	IX scan, MC=0	SYSIBM.SYSPLANDEP	SYSIBM.DSNGGX01	DP
2.2.0	1	IX scan MC=2	SYSIBM.SYSTABLESPACE	SYSIBM.DSNDSX01	
2.3.0	4	IX scan MC=1	SYSIBM.SYSPLAN	SYSIBM.DSNPPH01	
2.4.0	2	IX scan, MC=0	GH12.GH12T17	GH12.GH12X171	SP
3.1.0	0	IX scan, MC=0	GH12.GH12T17	GH12.GH12X171	SP
4.1.0	0	IX scan, MC=0	SYSIBM.SYSPLANDEP	SYSIBM.DSNGGX01	DP
4.2.0	1	IX scan MC=2	SYSIBM.SYSINDEXES	SYSIBM.DSNDXX01	
4.3.0	4	IX scan MC=1	SYSIBM.SYSPLAN	SYSIBM.DSNPPH01	
4.4.0	2	IX scan, MC=0	GH12.GH12T17	GH12.GH12X171	SP
5.1.0	0	IX scan, MC=0	GH12.GH12T17	GH12.GH12X171	SP

Access Path New

Seq.	M	Access	Table	Index	Other
1.1.0	3	Sort			
2.1.0	0	TS scan	GH12.GH12T17		SP
2.2.0	1	IX scan MC=1	SYSIBM.SYSPLAN	SYSIBM.DSNPPH01	
2.3.0	1	IX scan, MC=0	SYSIBM.SYSPLANDEP	SYSIBM.DSNGGX01	DP
2.4.0	1	IX scan MC=2	SYSIBM.SYSTABLESPACE	SYSIBM.DSNDSX01	
3.1.0	0	TS scan	GH12.GH12T17		SP
4.1.0	0	TS scan	GH12.GH12T17		SP
4.2.0	1	IX scan MC=1	SYSIBM.SYSPLAN	SYSIBM.DSNPPH01	
4.3.0	1	IX scan, MC=0	SYSIBM.SYSPLANDEP	SYSIBM.DSNGGX01	DP
4.4.0	1	IX scan MC=2	SYSIBM.SYSINDEXES	SYSIBM.DSNDXX01	
5.1.0	0	TS scan	GH12.GH12T17		SP

Dynamic Mouse Over Access Path Difference Highlighting



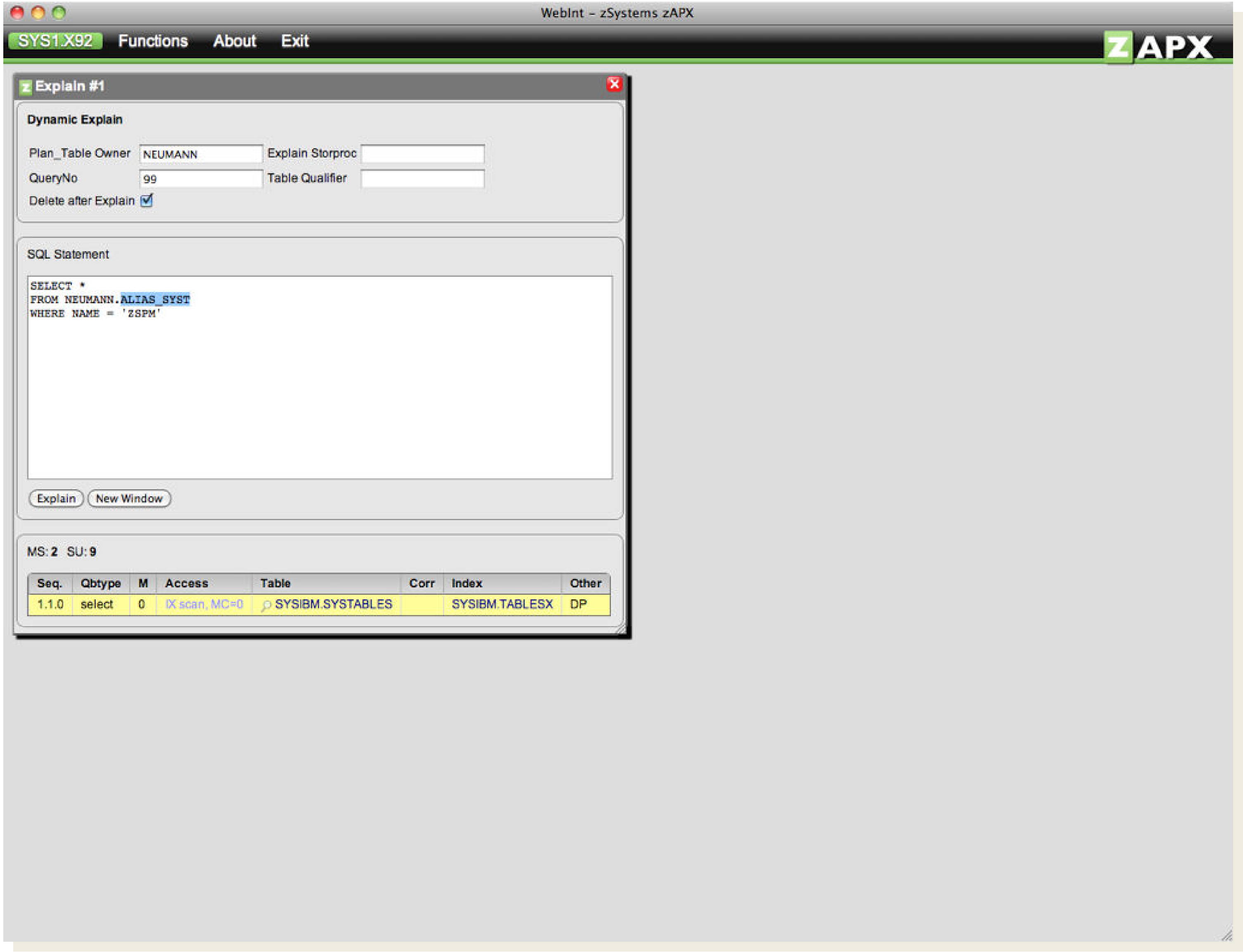
Access Path Old

Seq.	M	Access	Table	Index	Other
1.1.0	3	Sort			
2.1.0	0	IX scan, MC=0	SYSIBM.SYSPLANDEP	SYSIBM.DSNGGX01	DP
2.2.0	1	IX scan MC=2	SYSIBM.SYSTABLESPACE	SYSIBM.DSNDSX01	
2.3.0	4	IX scan MC=1	SYSIBM.SYSPLAN	SYSIBM.DSNPPH01	
2.4.0	2	IX scan, MC=0	GH12.GH12T17	GH12.GH12X171	SP
3.1.0	0	IX scan, MC=0	GH12.GH12T17	GH12.GH12X171	SP
4.1.0	0	IX scan, MC=0	SYSIBM.SYSPLANDEP	SYSIBM.DSNGGX01	DP
4.2.0	1	IX scan MC=2	SYSIBM.SYSINDEXES	SYSIBM.DSNDXX01	
4.3.0	4	IX scan MC=1	SYSIBM.SYSPLAN	SYSIBM.DSNPPH01	
4.4.0	2	IX scan, MC=0	GH12.GH12T17	GH12.GH12X171	SP
5.1.0	0	IX scan, MC=0	GH12.GH12T17	GH12.GH12X171	SP

Access Path New

Seq.	M	Access	Table	Index	Other
1.1.0	3	Sort			
2.1.0	0	TS scan	GH12.GH12T17		SP
2.2.0	1	IX scan MC=1	SYSIBM.SYSPLAN	SYSIBM.DSNPPH01	
2.3.0	1	IX scan, MC=0	SYSIBM.SYSPLANDEP	SYSIBM.DSNGGX01	DP
2.4.0	1	IX scan MC=2	SYSIBM.SYSTABLESPACE	SYSIBM.DSNDSX01	
3.1.0	0	TS scan	GH12.GH12T17		SP
4.1.0	0	TS scan	GH12.GH12T17		SP
4.2.0	1	IX scan MC=1	SYSIBM.SYSPLAN	SYSIBM.DSNPPH01	
4.3.0	1	IX scan, MC=0	SYSIBM.SYSPLANDEP	SYSIBM.DSNGGX01	DP
4.4.0	1	IX scan MC=2	SYSIBM.SYSINDEXES	SYSIBM.DSNDXX01	
5.1.0	0	TS scan	GH12.GH12T17		SP

Highlight Tables, Aliases, Views & Synonyms



Dynamic Explain

Plan_Table Owner	NEUMANN	Explain Storproc	
QueryNo	99	Table Qualifier	
Delete after Explain	☑		

SQL Statement

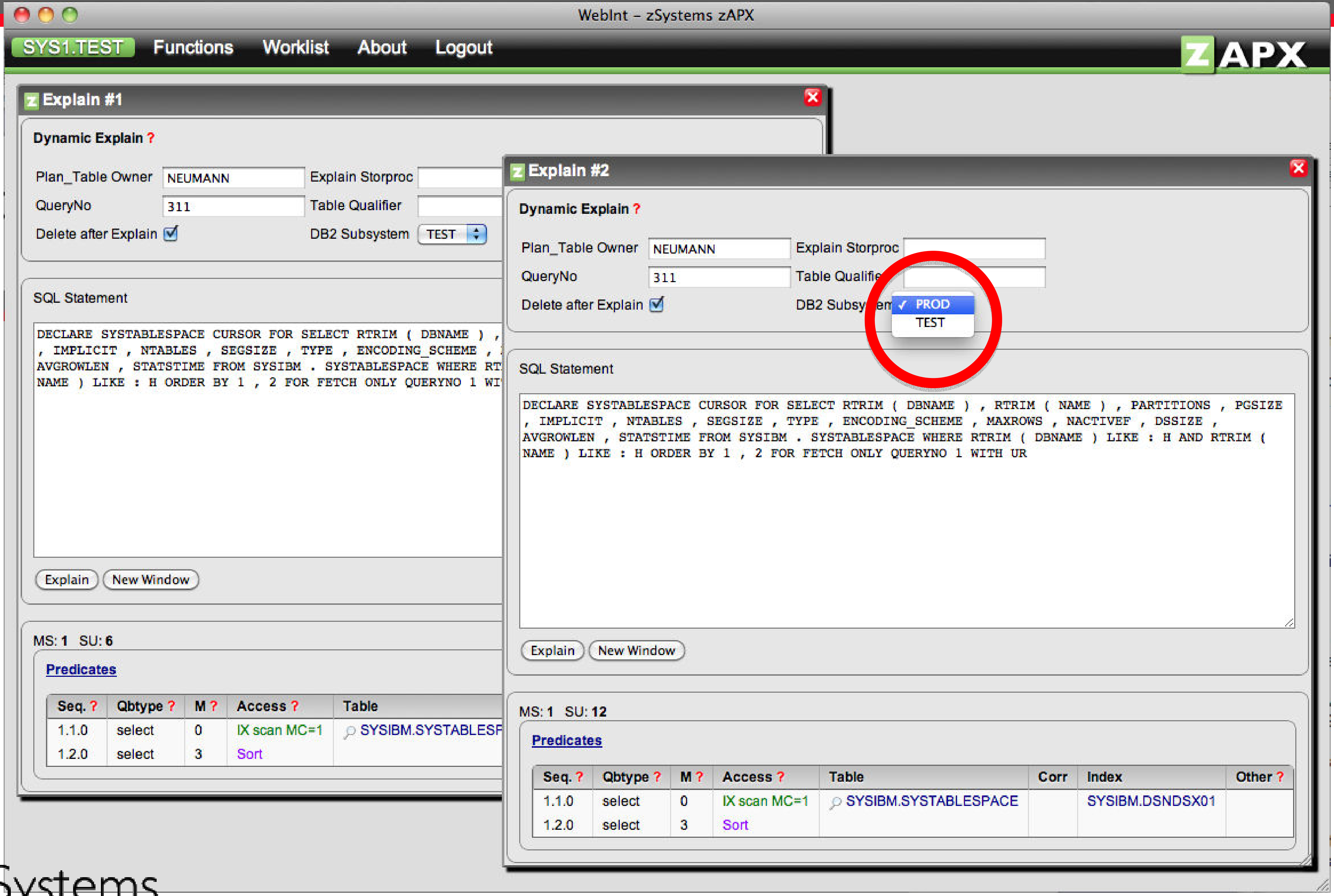
```
SELECT *
FROM NEUMANN.ALIAS_SYST
WHERE NAME = 'ZSPM'
```

MS: **2** SU: **9**

Seq.	Qbtype	M	Access	Table	Corr	Index	Other
1.1.0	select	0	IX scan, MC=0	SYSIBM.SYSTABLES		SYSIBM.TABLESX	DP

Systems
Database Solutions GmbH

Switch DB2 subsystems



Explain #1

Dynamic Explain ?

Plan_Table Owner: NEUMANN
Explain Storproc:
QueryNo: 311
Table Qualifier:
Delete after Explain ☑
DB2 Subsystem: TEST

SQL Statement

```
DECLARE SYSTABLESPACE CURSOR FOR SELECT RTRIM ( DBNAME ) ,
, IMPLICIT , NTABLES , SEGSIZE , TYPE , ENCODING_SCHEME ,
AVGROWLEN , STATSTIME FROM SYSIBM . SYSTABLESPACE WHERE RT
NAME ) LIKE : H ORDER BY 1 , 2 FOR FETCH ONLY QUERYNO 1 WI
```

MS: 1 SU: 6

Predicates

Seq. ?	Qbtype ?	M ?	Access ?	Table
1.1.0	select	0	IX scan MC=1	SYSIBM.SYSTABLESP
1.2.0	select	3	Sort	

Explain #2

Dynamic Explain ?

Plan_Table Owner: NEUMANN
Explain Storproc:
QueryNo: 311
Table Qualifie
Delete after Explain ☑
DB2 Subsystem: ✓ PROD / TEST

SQL Statement

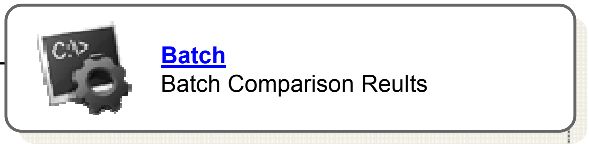
```
DECLARE SYSTABLESPACE CURSOR FOR SELECT RTRIM ( DBNAME ) , RTRIM ( NAME ) , PARTITIONS , PGSIZE
, IMPLICIT , NTABLES , SEGSIZE , TYPE , ENCODING_SCHEME , MAXROWS , NACTIVEF , DSSIZE ,
AVGROWLEN , STATSTIME FROM SYSIBM . SYSTABLESPACE WHERE RTRIM ( DBNAME ) LIKE : H AND RTRIM (
NAME ) LIKE : H ORDER BY 1 , 2 FOR FETCH ONLY QUERYNO 1 WITH UR
```

MS: 1 SU: 12

Predicates

Seq. ?	Qbtype ?	M ?	Access ?	Table	Corr	Index	Other ?
1.1.0	select	0	IX scan MC=1	SYSIBM.SYSTABLESPACE		SYSIBM.DSNDSX01	
1.2.0	select	3	Sort				

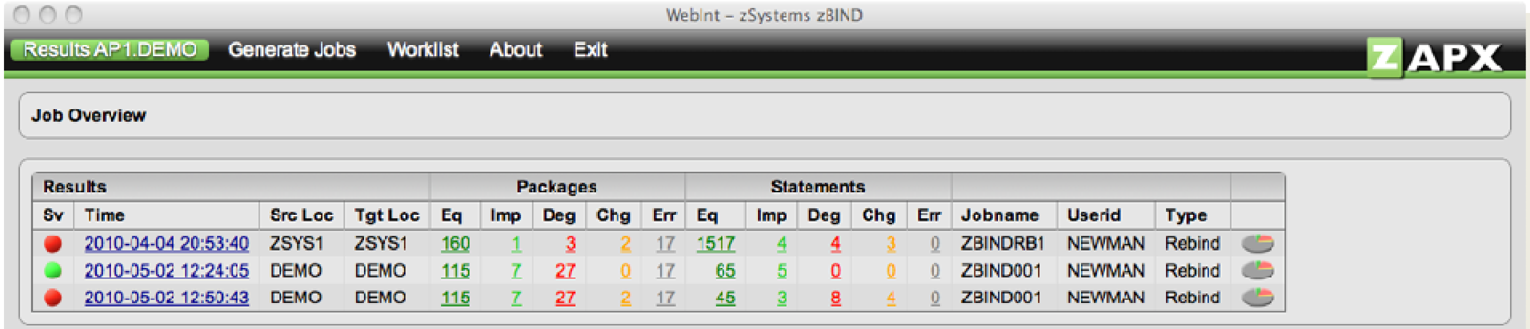
Automation



Batch
Batch Comparison Reults

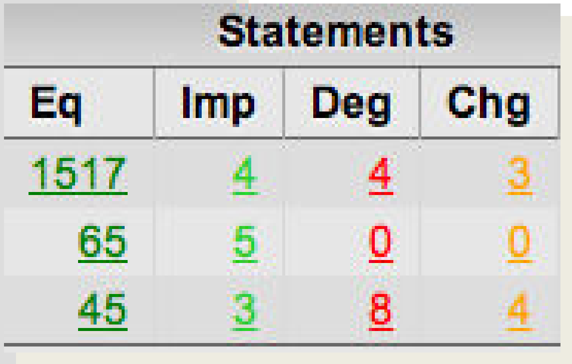
- Z APX explains SQL and compares access paths of static and dynamic SQL for the purpose of:
  - Application Development
  - Quality Assurance
  - Securing REBINDs in the maintenance process
  - Securing dynamic SQL and REBINDS during migration to a new version of DB2

Analysis Job Overview



Job Overview

Results				Packages					Statements								
Sv	Time	Src Loc	Tgt Loc	Eq	Imp	Deg	Chg	Err	Eq	Imp	Deg	Chg	Err	Jobname	Userid	Type	
🔴	2010-04-04 20:53:40	ZSYS1	ZSYS1	160	1	3	2	17	1517	4	4	3	0	ZBINDRB1	NEWMAN	Rebind	
🟢	2010-05-02 12:24:05	DEMO	DEMO	115	7	27	0	17	65	5	0	0	0	ZBIND001	NEWMAN	Rebind	
🔴	2010-05-02 12:50:43	DEMO	DEMO	115	7	27	2	17	45	3	8	4	0	ZBIND001	NEWMAN	Rebind	

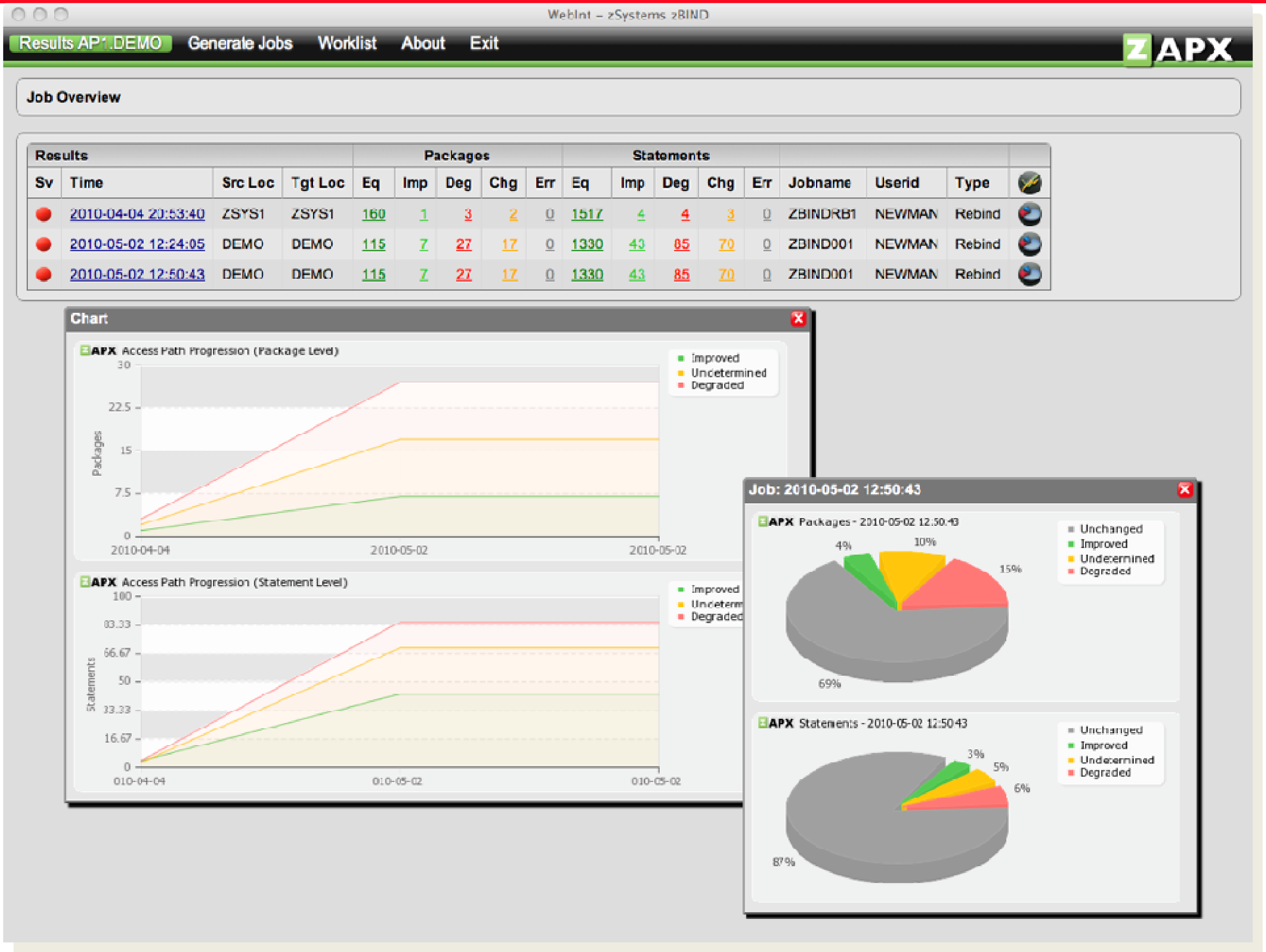


Statements			
Eq	Imp	Deg	Chg
1517	4	4	3
65	5	0	0
45	3	8	4

The web-based Graphical User Interface allows direct access to every category, e.g.

- Equal
- Improved
- Degraded

Graphical Overview

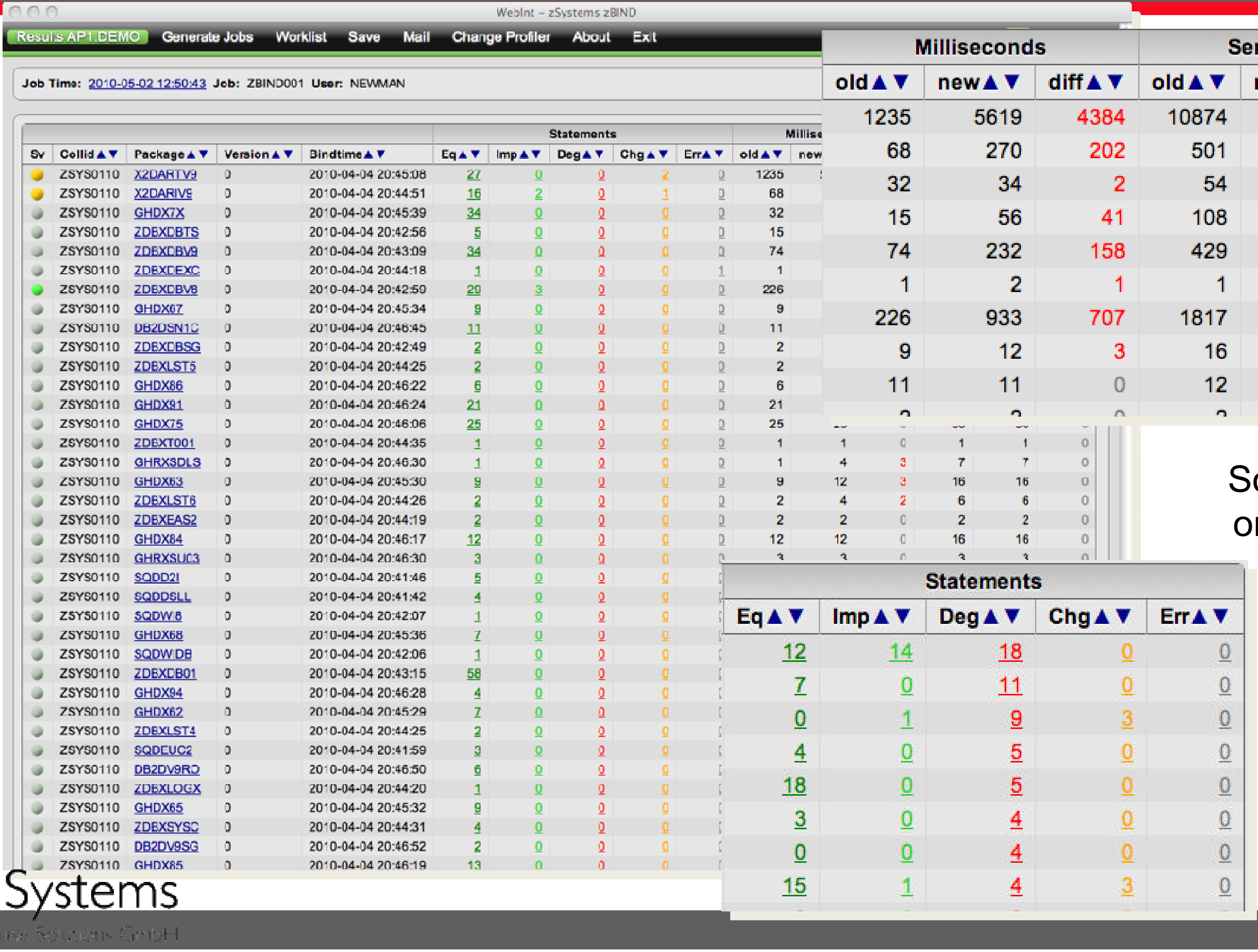


Job Overview

Results				Packages					Statements								
Sv	Time	Src Loc	Tgt Loc	Eq	Imp	Deg	Chg	Err	Eq	Imp	Deg	Chg	Err	Jobname	Userid	Type	
●	2010-04-04 20:53:40	ZSYS1	ZSYS1	160	1	3	2	0	1517	4	4	3	0	ZBINDRB1	NEWMAN	Rebind	
●	2010-05-02 12:24:05	DEMO	DEMO	115	7	27	17	0	1330	43	85	70	0	ZBIND001	NEWMAN	Rebind	
●	2010-05-02 12:50:43	DEMO	DEMO	115	7	27	17	0	1330	43	85	70	0	ZBIND001	NEWMAN	Rebind	

Generate Charts about SQL Access Path Progression over time or Categorys

Package Overview



Job Time: 2010-05-02 12:50:43 **Job:** ZBIND001 **User:** NEWMAN

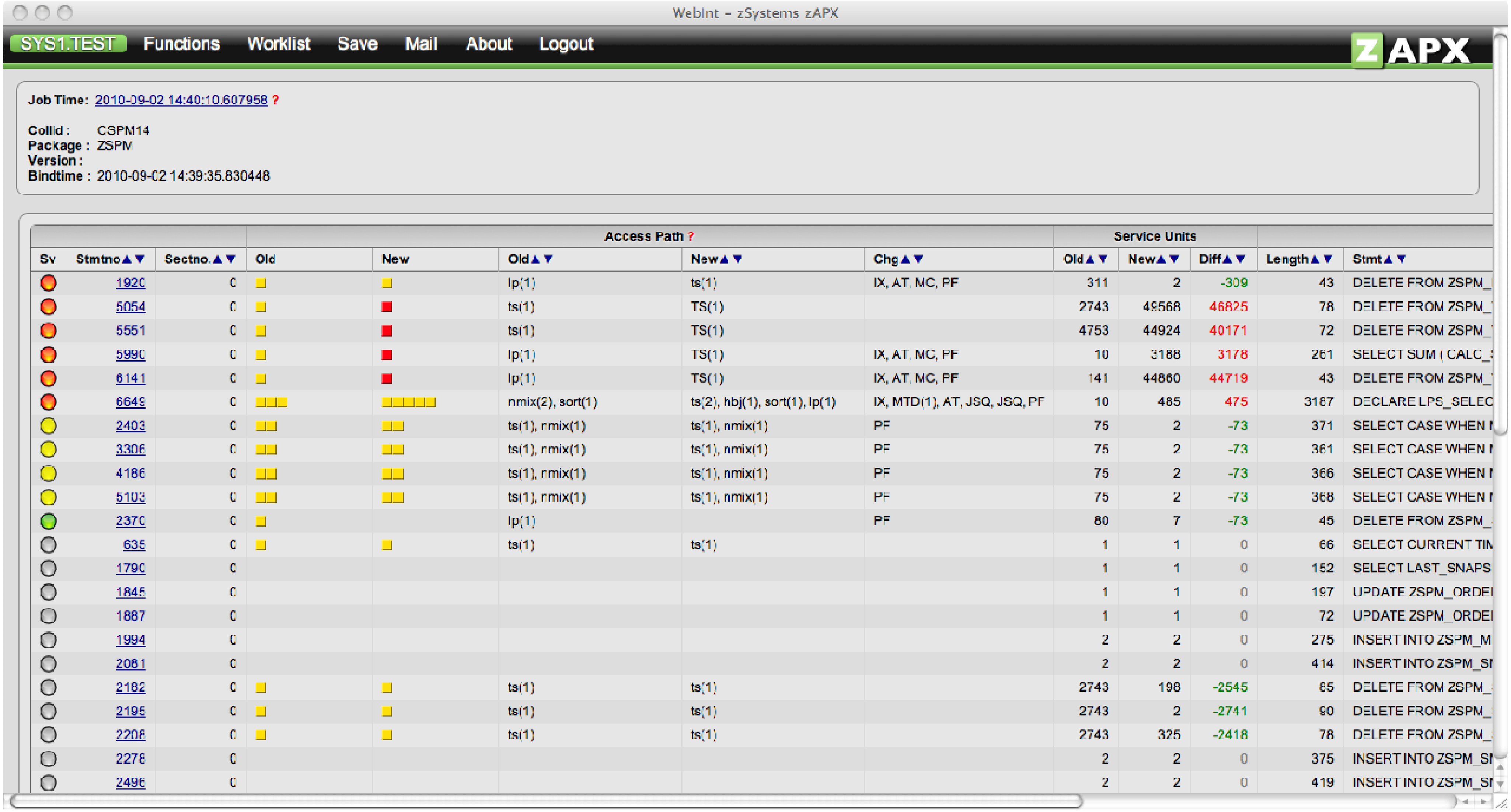
					Statements					Millise					
Sv	Collid ▲▼	Package ▲▼	Version ▲▼	Bindtime ▲▼	Eq ▲▼	Imp ▲▼	Deg ▲▼	Chg ▲▼	Err ▲▼	old ▲▼	new				
🟡	ZSYS0110	X2DARTV9	0	2010-04-04 20:45:08	27	0	0	2	0	1235					
🟡	ZSYS0110	X2DARIV9	0	2010-04-04 20:44:51	16	2	0	1	0	68					
●	ZSYS0110	GHDX7X	0	2010-04-04 20:45:39	34	0	0	0	0	32					
●	ZSYS0110	ZDEXCBTS	0	2010-04-04 20:42:56	5	0	0	0	0	15					
●	ZSYS0110	ZDEXCBV9	0	2010-04-04 20:43:09	34	0	0	0	0	74					
●	ZSYS0110	ZDEXCEXC	0	2010-04-04 20:44:18	1	0	0	0	1	1					
🟢	ZSYS0110	ZDEXCBV8	0	2010-04-04 20:42:59	29	3	0	0	0	226					
●	ZSYS0110	GHDX67	0	2010-04-04 20:45:34	9	0	0	0	0	9					
●	ZSYS0110	DB2DSN1C	0	2010-04-04 20:46:45	11	0	0	0	0	11					
●	ZSYS0110	ZDEXCBSG	0	2010-04-04 20:42:49	2	0	0	0	0	2					
●	ZSYS0110	ZDEXLST5	0	2010-04-04 20:44:25	2	0	0	0	0	2					
●	ZSYS0110	GHDX86	0	2010-04-04 20:46:22	6	0	0	0	0	6					
●	ZSYS0110	GHDX91	0	2010-04-04 20:46:24	21	0	0	0	0	21					
●	ZSYS0110	GHDX75	0	2010-04-04 20:46:06	25	0	0	0	0	25					
●	ZSYS0110	ZDEXT001	0	2010-04-04 20:44:35	1	0	0	0	0	1	1	0	1	1	0
●	ZSYS0110	GHRXSDLS	0	2010-04-04 20:46:30	1	0	0	0	0	1	4	3	7	7	0
●	ZSYS0110	GHDX63	0	2010-04-04 20:45:30	9	0	0	0	0	9	12	3	16	16	0
●	ZSYS0110	ZDEXLST6	0	2010-04-04 20:44:26	2	0	0	0	0	2	4	2	6	6	0
●	ZSYS0110	ZDEXEAS2	0	2010-04-04 20:44:19	2	0	0	0	0	2	2	0	2	2	0
●	ZSYS0110	GHDX84	0	2010-04-04 20:46:17	12	0	0	0	0	12	12	0	16	16	0
●	ZSYS0110	GHRXSUC3	0	2010-04-04 20:46:30	3	0	0	0	0	3	3	0	3	3	0
●	ZSYS0110	SQDD2I	0	2010-04-04 20:41:46	5	0	0	0							
●	ZSYS0110	SQDDSLL	0	2010-04-04 20:41:42	4	0	0	0							
●	ZSYS0110	SQDW8	0	2010-04-04 20:42:07	1	0	0	0							
●	ZSYS0110	GHDX68	0	2010-04-04 20:45:36	7	0	0	0							
●	ZSYS0110	SQDWDB	0	2010-04-04 20:42:06	1	0	0	0							
●	ZSYS0110	ZDEXCB01	0	2010-04-04 20:43:15	58	0	0	0							
●	ZSYS0110	GHDX94	0	2010-04-04 20:46:28	4	0	0	0							
●	ZSYS0110	GHDX62	0	2010-04-04 20:45:29	7	0	0	0							
●	ZSYS0110	ZDEXLST4	0	2010-04-04 20:44:25	2	0	0	0							
●	ZSYS0110	SQDEUC2	0	2010-04-04 20:41:59	3	0	0	0							
●	ZSYS0110	DB2DV9RO	0	2010-04-04 20:46:50	6	0	0	0							
●	ZSYS0110	ZDEXLOGX	0	2010-04-04 20:44:20	1	0	0	0							
●	ZSYS0110	GHDX65	0	2010-04-04 20:45:32	9	0	0	0							
●	ZSYS0110	ZDEXSYSC	0	2010-04-04 20:44:31	4	0	0	0							
●	ZSYS0110	DB2DV9SG	0	2010-04-04 20:46:52	2	0	0	0							
●	ZSYS0110	GHDX85	0	2010-04-04 20:46:19	13	0	0	0							

Milliseconds			Service Units		
old ▲▼	new ▲▼	diff ▲▼	old ▲▼	new ▲▼	diff ▲▼
1235	5619	4384	10874	11063	189
68	270	202	501	520	19
32	34	2	54	58	4
15	56	41	108	110	2
74	232	158	429	430	1
1	2	1	1	2	1
226	933	707	1817	1818	1
9	12	3	16	16	0
11	11	0	12	12	0

Sort by cost
or category

Statements				
Eq ▲▼	Imp ▲▼	Deg ▲▼	Chg ▲▼	Err ▲▼
12	14	18	0	0
7	0	11	0	0
0	1	9	3	0
4	0	5	0	0
18	0	5	0	0
3	0	4	0	0
0	0	4	0	0
15	1	4	3	0

Statement Overview



Job Time: 2010-09-02 14:40:10.607958 ?

Collid : CSPM14
Package : ZSPM
Version :
Bindtime : 2010-09-02 14:39:35.830448

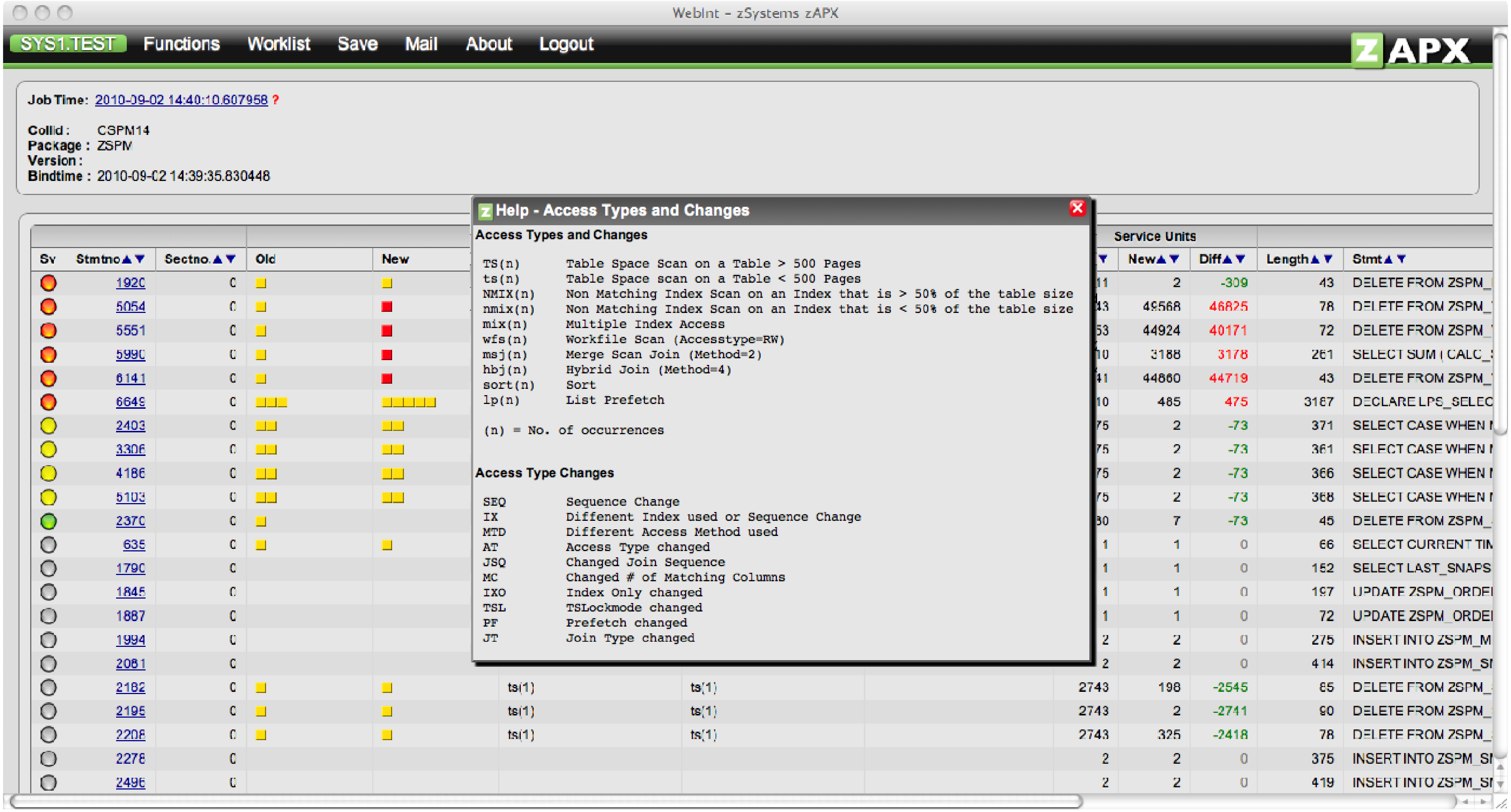
			Access Path ?					Service Units				
Sv	Stmtno ▲▼	Sectno. ▲▼	Old	New	Old ▲▼	New ▲▼	Chg ▲▼	Old ▲▼	New ▲▼	Diff ▲▼	Length ▲▼	Stmt ▲▼
🔴	1920	0	🟨	🟨	lp(1)	ts(1)	IX, AT, MC, PF	311	2	-309	43	DELETE FROM ZSPM_
🔴	5054	0	🟨	🟥	ts(1)	TS(1)		2743	49568	46825	78	DELETE FROM ZSPM_
🔴	5551	0	🟨	🟥	ts(1)	TS(1)		4753	44924	40171	72	DELETE FROM ZSPM_
🔴	5990	0	🟨	🟥	lp(1)	TS(1)	IX, AT, MC, PF	10	3188	3178	261	SELECT SUM (CALC_
🔴	6141	0	🟨	🟥	lp(1)	TS(1)	IX, AT, MC, PF	141	44860	44719	43	DELETE FROM ZSPM_
🔴	6649	0	🟨🟨🟨	🟨🟨🟨🟨🟨	nmix(2), sort(1)	ts(2), hbj(1), sort(1), lp(1)	IX, MTD(1), AT, JSQ, JSQ, PF	10	485	475	3167	DECLARE LPS_SELEC
🟡	2403	0	🟨🟨	🟨🟨	ts(1), nmix(1)	ts(1), nmix(1)	PF	75	2	-73	371	SELECT CASE WHEN
🟡	3306	0	🟨🟨	🟨🟨	ts(1), nmix(1)	ts(1), nmix(1)	PF	75	2	-73	361	SELECT CASE WHEN
🟡	4186	0	🟨🟨	🟨🟨	ts(1), nmix(1)	ts(1), nmix(1)	PF	75	2	-73	366	SELECT CASE WHEN
🟡	5103	0	🟨🟨	🟨🟨	ts(1), nmix(1)	ts(1), nmix(1)	PF	75	2	-73	368	SELECT CASE WHEN
🟢	2370	0	🟨		lp(1)		PF	80	7	-73	45	DELETE FROM ZSPM_
○	635	0	🟨	🟨	ts(1)	ts(1)		1	1	0	66	SELECT CURRENT TIM
○	1790	0						1	1	0	152	SELECT LAST_SNAPS
○	1845	0						1	1	0	197	UPDATE ZSPM_ORDE
○	1887	0						1	1	0	72	UPDATE ZSPM_ORDE
○	1994	0						2	2	0	275	INSERT INTO ZSPM_M
○	2081	0						2	2	0	414	INSERT INTO ZSPM_S
○	2182	0	🟨	🟨	ts(1)	ts(1)		2743	198	-2545	65	DELETE FROM ZSPM_
○	2195	0	🟨	🟨	ts(1)	ts(1)		2743	2	-2741	90	DELETE FROM ZSPM_
○	2208	0	🟨	🟨	ts(1)	ts(1)		2743	325	-2418	78	DELETE FROM ZSPM_
○	2278	0						2	2	0	375	INSERT INTO ZSPM_S
○	2496	0						2	2	0	419	INSERT INTO ZSPM_S

Access path rating & changes at a glance

softwareonz
...a better way on z

Z Systems
Database Solutions GmbH

Statement Overview



Job Time: 2010-09-02 14:40:10.607958 ?

Collid : CSPM14
Package : ZSPM
Version :
Bindtime : 2010-09-02 14:39:35.830448

Help - Access Types and Changes

Access Types and Changes

```
TS(n)      Table Space Scan on a Table > 500 Pages
ts(n)      Table Space scan on a Table < 500 Pages
NMIX(n)    Non Matching Index Scan on an Index that is > 50% of the table size
nmix(n)    Non Matching Index Scan on an Index that is < 50% of the table size
mix(n)     Multiple Index Access
wfs(n)     Workfile Scan (Accesstype=RW)
msj(n)     Merge Scan Join (Method=2)
hbj(n)     Hybrid Join (Method=4)
sort(n)    Sort
lp(n)      List Prefetch

(n) = No. of occurrences
```

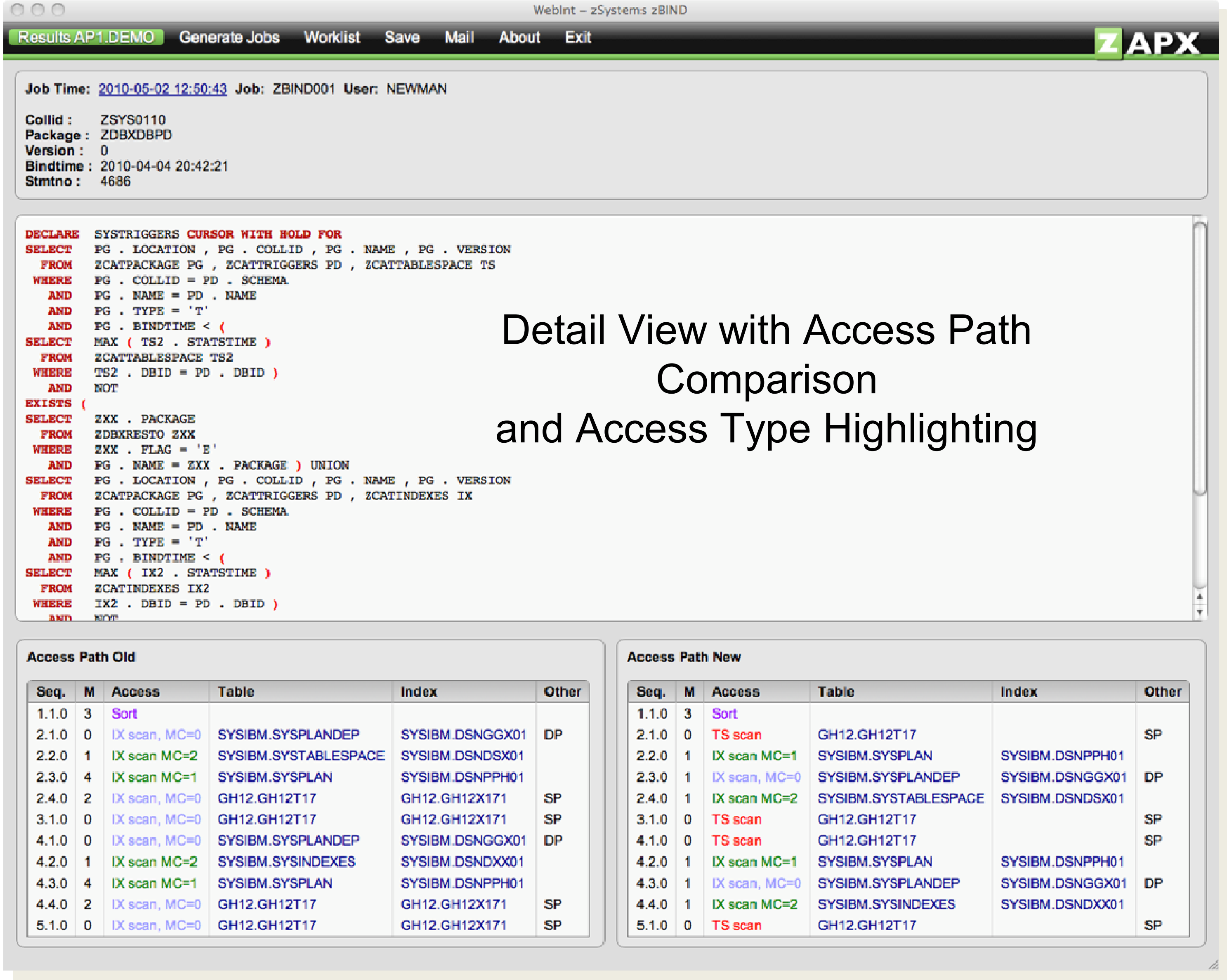
Access Type Changes

```
SEQ        Sequence Change
IX         Diffenent Index used or Sequence Change
MTD        Different Access Method used
AT         Access Type changed
JSQ        Changed Join Sequence
MC         Changed # of Matching Columns
IXO        Index Only changed
TSL        TSLockmode changed
PF         Prefetch changed
JT         Join Type changed
```

								Service Units				
Sv	Stmtno▲▼	Sectno.▲▼	Old	New				▼	New▲▼	Diff▲▼	Length▲▼	Stmt▲▼
●	1920	0	🟨	🟨				11	2	-309	43	DELETE FROM ZSPM_
●	5054	0	🟨	🟥				43	49568	46825	78	DELETE FROM ZSPM_
●	5551	0	🟨	🟥				53	44924	40171	72	DELETE FROM ZSPM_
●	5990	0	🟨	🟥				10	3188	3178	261	SELECT SUM (CALC_
●	6141	0	🟨	🟥				41	44860	44719	43	DELETE FROM ZSPM_
●	6649	0	🟨🟨🟨	🟨🟨🟨🟨🟨				10	485	475	3167	DECLARE LPS_SELEC
●	2403	0	🟨🟨	🟨🟨				75	2	-73	371	SELECT CASE WHEN
●	3306	0	🟨🟨	🟨🟨				75	2	-73	361	SELECT CASE WHEN
●	4186	0	🟨🟨	🟨🟨				75	2	-73	366	SELECT CASE WHEN
●	5103	0	🟨🟨	🟨🟨				75	2	-73	368	SELECT CASE WHEN
●	2370	0	🟨					30	7	-73	45	DELETE FROM ZSPM_
○	635	0	🟨	🟨				1	1	0	66	SELECT CURRENT TIM
○	1790	0						1	1	0	152	SELECT LAST_SNAPS
○	1845	0						1	1	0	197	UPDATE ZSPM_ORDE
○	1887	0						1	1	0	72	UPDATE ZSPM_ORDE
○	1994	0						2	2	0	275	INSERT INTO ZSPM_M
○	2081	0						2	2	0	414	INSERT INTO ZSPM_S
○	2182	0	🟨	🟨	ts(1)	ts(1)		2743	198	-2545	65	DELETE FROM ZSPM_
○	2195	0	🟨	🟨	ts(1)	ts(1)		2743	2	-2741	90	DELETE FROM ZSPM_
○	2208	0	🟨	🟨	ts(1)	ts(1)		2743	325	-2418	78	DELETE FROM ZSPM_
○	2278	0						2	2	0	375	INSERT INTO ZSPM_S
○	2496	0						2	2	0	419	INSERT INTO ZSPM_S

Access path rating & changes at a glance

Stmt Overview



Job Time: 2010-05-02 12:50:43 **Job:** ZBIND001 **User:** NEWMAN

Collid : ZSYS0110
Package : ZDBXDBPD
Version : 0
Bindtime : 2010-04-04 20:42:21
Stmtno : 4686

```
DECLARE  SYSTRIGGERS CURSOR WITH HOLD FOR
SELECT   PG . LOCATION , PG . COLLID , PG . NAME , PG . VERSION
  FROM   ZCATPACKAGE PG , ZCATTRIGGERS PD , ZCATTABLESPACE TS
 WHERE   PG . COLLID = PD . SCHEMA
   AND   PG . NAME = PD . NAME
   AND   PG . TYPE = 'T'
   AND   PG . BINDTIME < (
SELECT   MAX ( TS2 . STATSTIME )
  FROM   ZCATTABLESPACE TS2
 WHERE   TS2 . DBID = PD . DBID )
   AND   NOT
EXISTS (
SELECT   ZXX . PACKAGE
  FROM   ZDBXRESTO ZXX
 WHERE   ZXX . FLAG = 'E'
   AND   PG . NAME = ZXX . PACKAGE ) UNION
SELECT   PG . LOCATION , PG . COLLID , PG . NAME , PG . VERSION
  FROM   ZCATPACKAGE PG , ZCATTRIGGERS PD , ZCATINDEXES IX
 WHERE   PG . COLLID = PD . SCHEMA
   AND   PG . NAME = PD . NAME
   AND   PG . TYPE = 'T'
   AND   PG . BINDTIME < (
SELECT   MAX ( IX2 . STATSTIME )
  FROM   ZCATINDEXES IX2
 WHERE   IX2 . DBID = PD . DBID )
   AND   NOT
```

Detail View with Access Path Comparison and Access Type Highlighting

Access Path Old

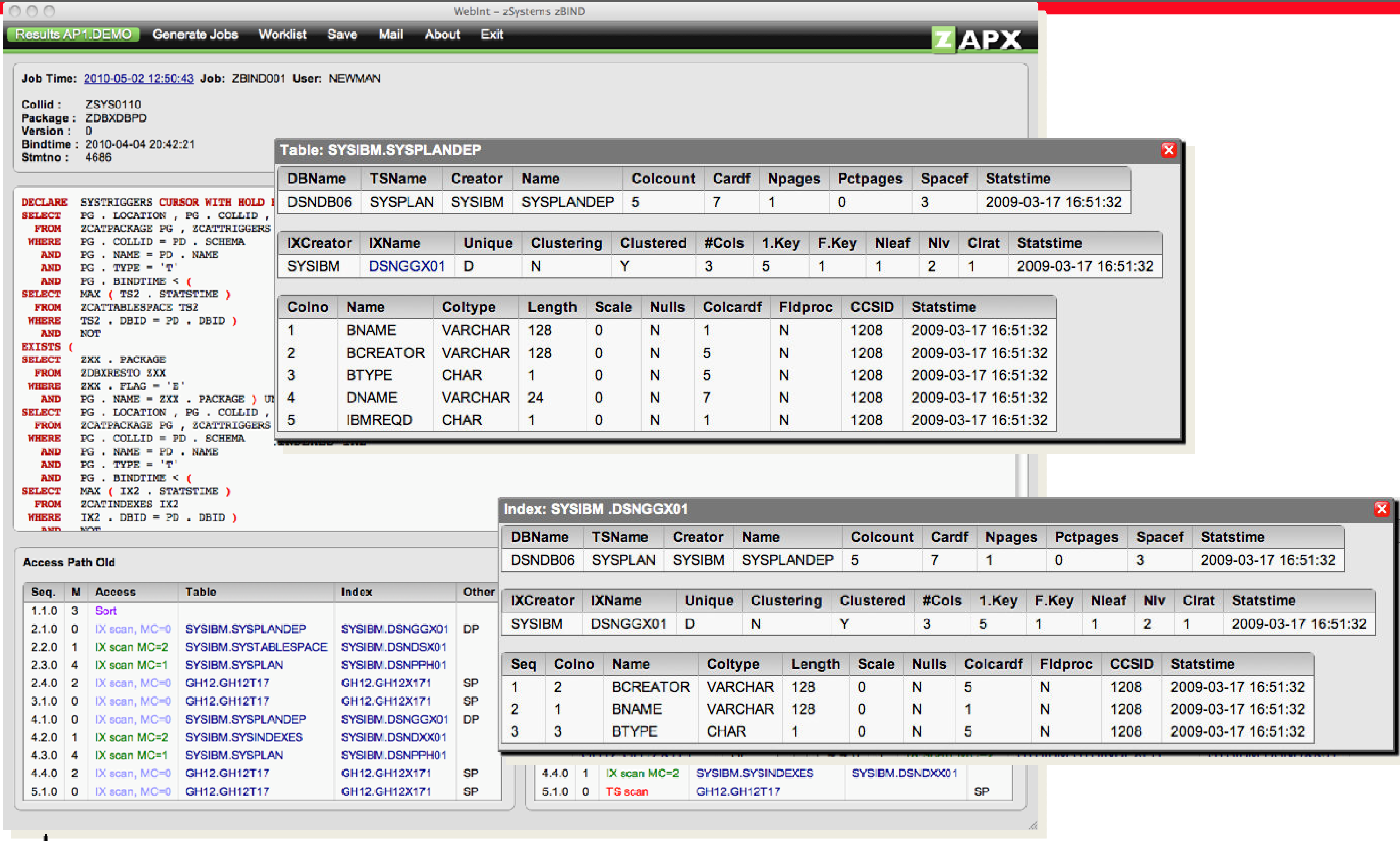
Seq.	M	Access	Table	Index	Other
1.1.0	3	Sort			
2.1.0	0	IX scan, MC=0	SYSIBM.SYSPLANDEP	SYSIBM.DSNGGX01	DP
2.2.0	1	IX scan MC=2	SYSIBM.SYSTABLESPACE	SYSIBM.DSNDSX01	
2.3.0	4	IX scan MC=1	SYSIBM.SYSPLAN	SYSIBM.DSNPPH01	
2.4.0	2	IX scan, MC=0	GH12.GH12T17	GH12.GH12X171	SP
3.1.0	0	IX scan, MC=0	GH12.GH12T17	GH12.GH12X171	SP
4.1.0	0	IX scan, MC=0	SYSIBM.SYSPLANDEP	SYSIBM.DSNGGX01	DP
4.2.0	1	IX scan MC=2	SYSIBM.SYSINDEXES	SYSIBM.DSNDXX01	
4.3.0	4	IX scan MC=1	SYSIBM.SYSPLAN	SYSIBM.DSNPPH01	
4.4.0	2	IX scan, MC=0	GH12.GH12T17	GH12.GH12X171	SP
5.1.0	0	IX scan, MC=0	GH12.GH12T17	GH12.GH12X171	SP

Access Path New

Seq.	M	Access	Table	Index	Other
1.1.0	3	Sort			
2.1.0	0	TS scan	GH12.GH12T17		SP
2.2.0	1	IX scan MC=1	SYSIBM.SYSPLAN	SYSIBM.DSNPPH01	
2.3.0	1	IX scan, MC=0	SYSIBM.SYSPLANDEP	SYSIBM.DSNGGX01	DP
2.4.0	1	IX scan MC=2	SYSIBM.SYSTABLESPACE	SYSIBM.DSNDSX01	
3.1.0	0	TS scan	GH12.GH12T17		SP
4.1.0	0	TS scan	GH12.GH12T17		SP
4.2.0	1	IX scan MC=1	SYSIBM.SYSPLAN	SYSIBM.DSNPPH01	
4.3.0	1	IX scan, MC=0	SYSIBM.SYSPLANDEP	SYSIBM.DSNGGX01	DP
4.4.0	1	IX scan MC=2	SYSIBM.SYSINDEXES	SYSIBM.DSNDXX01	
5.1.0	0	TS scan	GH12.GH12T17		SP

Statistics Browser

softwareonz
...a better way on z



Job Time: 2010-05-02 12:50:43 **Job:** ZBIND001 **User:** NEWMAN

Collid : ZSYS0110
Package : ZDBXDBPD
Version : 0
Bindtime : 2010-04-04 20:42:21
Stmtno : 4686

```
DECLARE  SYSTRIGGERS CURSOR WITH HOLD F
SELECT   PG . LOCATION , PG . COLLID ,
  FROM   ZCATPACKAGE PG , ZCATTRIGGERS
 WHERE   PG . COLLID = PD . SCHEMA
   AND   PG . NAME = PD . NAME
   AND   PG . TYPE = 'T'
   AND   PG . BINDTIME < (
SELECT   MAX ( TS2 . STATSTIME )
  FROM   ZCATTABLESPACE TS2
 WHERE   TS2 . DBID = PD . DBID )
   AND   NOT
EXISTS (
SELECT   ZXX . PACKAGE
  FROM   ZDBXRESTO ZXX
 WHERE   ZXX . FLAG = 'E'
   AND   PG . NAME = ZXX . PACKAGE ) UN
SELECT   PG . LOCATION , PG . COLLID ,
  FROM   ZCATPACKAGE PG , ZCATTRIGGERS
 WHERE   PG . COLLID = PD . SCHEMA
   AND   PG . NAME = PD . NAME
   AND   PG . TYPE = 'T'
   AND   PG . BINDTIME < (
SELECT   MAX ( IX2 . STATSTIME )
  FROM   ZCATINDEXES IX2
 WHERE   IX2 . DBID = PD . DBID )
   AND   NOT
```

Access Path Old

Seq.	M	Access	Table	Index	Other
1.1.0	3	Sort			
2.1.0	0	IX scan, MC=0	SYSIBM.SYSPLANDEP	SYSIBM.DSNGGX01	DP
2.2.0	1	IX scan MC=2	SYSIBM.SYSTABLESPACE	SYSIBM.DSNDSX01	
2.3.0	4	IX scan MC=1	SYSIBM.SYSPLAN	SYSIBM.DSNPPH01	
2.4.0	2	IX scan, MC=0	GH12.GH12T17	GH12.GH12X171	SP
3.1.0	0	IX scan, MC=0	GH12.GH12T17	GH12.GH12X171	SP
4.1.0	0	IX scan, MC=0	SYSIBM.SYSPLANDEP	SYSIBM.DSNGGX01	DP
4.2.0	1	IX scan MC=2	SYSIBM.SYSINDEXES	SYSIBM.DSNDXX01	
4.3.0	4	IX scan MC=1	SYSIBM.SYSPLAN	SYSIBM.DSNPPH01	
4.4.0	2	IX scan, MC=0	GH12.GH12T17	GH12.GH12X171	SP
5.1.0	0	IX scan, MC=0	GH12.GH12T17	GH12.GH12X171	SP

4.4.0	1	IX scan MC=2	SYSIBM.SYSINDEXES	SYSIBM.DSNDXX01	
5.1.0	0	TS scan	GH12.GH12T17		SP

Table: SYSIBM.SYSPLANDEP

DBName	TSName	Creator	Name	Colcount	Cardf	Npages	Pctpages	Spacef	Statstime
DSNDB06	SYSPLAN	SYSIBM	SYSPLANDEP	5	7	1	0	3	2009-03-17 16:51:32

IXCreator	IXName	Unique	Clustering	Clustered	#Cols	1.Key	F.Key	Nleaf	Nlv	Clrat	Statstime
SYSIBM	DSNGGX01	D	N	Y	3	5	1	1	2	1	2009-03-17 16:51:32

Colno	Name	Coltype	Length	Scale	Nulls	Colcardf	Fldproc	CCSID	Statstime
1	BNAME	VARCHAR	128	0	N	1	N	1208	2009-03-17 16:51:32
2	BCREATOR	VARCHAR	128	0	N	5	N	1208	2009-03-17 16:51:32
3	BTYPE	CHAR	1	0	N	5	N	1208	2009-03-17 16:51:32
4	DNAME	VARCHAR	24	0	N	7	N	1208	2009-03-17 16:51:32
5	IBMREQD	CHAR	1	0	N	1	N	1208	2009-03-17 16:51:32

Index: SYSIBM .DSNGGX01

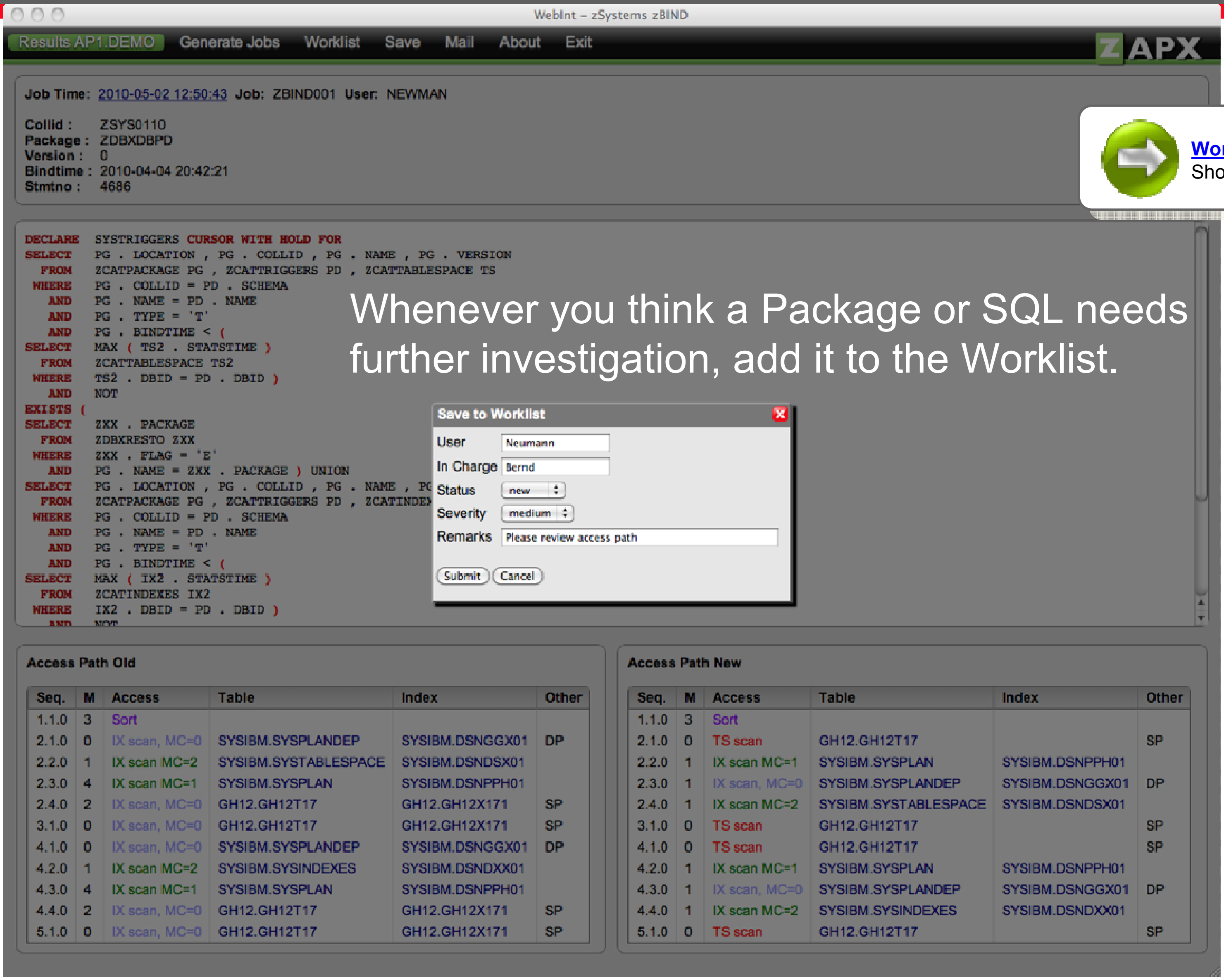
DBName	TSName	Creator	Name	Colcount	Cardf	Npages	Pctpages	Spacef	Statstime
DSNDB06	SYSPLAN	SYSIBM	SYSPLANDEP	5	7	1	0	3	2009-03-17 16:51:32

IXCreator	IXName	Unique	Clustering	Clustered	#Cols	1.Key	F.Key	Nleaf	Nlv	Clrat	Statstime
SYSIBM	DSNGGX01	D	N	Y	3	5	1	1	2	1	2009-03-17 16:51:32

Seq	Colno	Name	Coltype	Length	Scale	Nulls	Colcardf	Fldproc	CCSID	Statstime
1	2	BCREATOR	VARCHAR	128	0	N	5	N	1208	2009-03-17 16:51:32
2	1	BNAME	VARCHAR	128	0	N	1	N	1208	2009-03-17 16:51:32
3	3	BTYPE	CHAR	1	0	N	5	N	1208	2009-03-17 16:51:32

zSystems
Database Solutions GmbH

Worklist



Job Time: 2010-05-02 12:50:43 Job: ZBIND001 User: NEWMAN

Collid : ZSYS0110
Package : ZDBXDBPD
Version : 0
Bindtime : 2010-04-04 20:42:21
Stmtno : 4686

```
DECLARE  SYSTRIGGERS CURSOR WITH HOLD FOR
SELECT   PG . LOCATION , PG . COLLID , PG . NAME , PG . VERSION
  FROM   ZCATPACKAGE PG , ZCATTRIGGERS PD , ZCATTABLESPACE TS
 WHERE   PG . COLLID = PD . SCHEMA
   AND   PG . NAME = PD . NAME
   AND   PG . TYPE = 'T'
   AND   PG . BINDTIME < (
SELECT   MAX ( TS2 . STATSTIME )
  FROM   ZCATTABLESPACE TS2
 WHERE   TS2 . DBID = PD . DBID )
   AND   NOT
EXISTS (
SELECT   ZXX . PACKAGE
  FROM   ZDBXRESTO ZXX
 WHERE   ZXX . FLAG = 'E'
   AND   PG . NAME = ZXX . PACKAGE ) UNION
SELECT   PG . LOCATION , PG . COLLID , PG . NAME , PG
  FROM   ZCATPACKAGE PG , ZCATTRIGGERS PD , ZCATINDEX
 WHERE   PG . COLLID = PD . SCHEMA
   AND   PG . NAME = PD . NAME
   AND   PG . TYPE = 'T'
   AND   PG . BINDTIME < (
SELECT   MAX ( IX2 . STATSTIME )
  FROM   ZCATINDEXES IX2
 WHERE   IX2 . DBID = PD . DBID )
   AND   NOT
```

Whenever you think a Package or SQL needs further investigation, add it to the Worklist.

Save to Worklist

User	Neumann
In Charge	Bernd
Status	new
Severity	medium
Remarks	Please review access path

Access Path Old

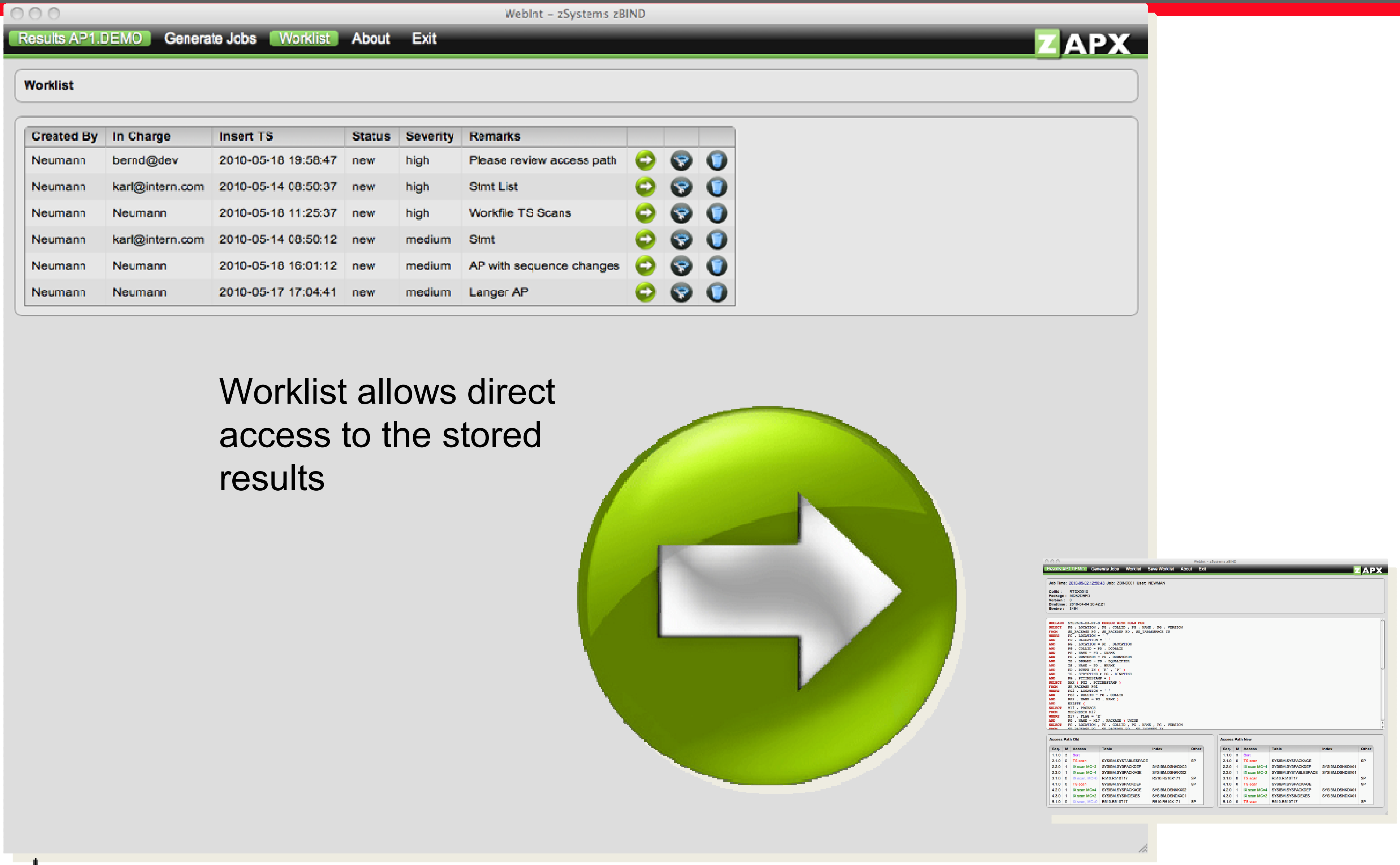
Seq.	M	Access	Table	Index	Other
1.1.0	3	Sort			
2.1.0	0	IX scan, MC=0	SYSIBM.SYSPLANDEP	SYSIBM.DSNGGX01	DP
2.2.0	1	IX scan MC=2	SYSIBM.SYSTABLESPACE	SYSIBM.DSNDSX01	
2.3.0	4	IX scan MC=1	SYSIBM.SYSPLAN	SYSIBM.DSNPPH01	
2.4.0	2	IX scan, MC=0	GH12.GH12T17	GH12.GH12X171	SP
3.1.0	0	IX scan, MC=0	GH12.GH12T17	GH12.GH12X171	SP
4.1.0	0	IX scan, MC=0	SYSIBM.SYSPLANDEP	SYSIBM.DSNGGX01	DP
4.2.0	1	IX scan MC=2	SYSIBM.SYSINDEXES	SYSIBM.DSNDXX01	
4.3.0	4	IX scan MC=1	SYSIBM.SYSPLAN	SYSIBM.DSNPPH01	
4.4.0	2	IX scan, MC=0	GH12.GH12T17	GH12.GH12X171	SP
5.1.0	0	IX scan, MC=0	GH12.GH12T17	GH12.GH12X171	SP

Access Path New

Seq.	M	Access	Table	Index	Other
1.1.0	3	Sort			
2.1.0	0	TS scan	GH12.GH12T17		SP
2.2.0	1	IX scan MC=1	SYSIBM.SYSPLAN	SYSIBM.DSNPPH01	
2.3.0	1	IX scan, MC=0	SYSIBM.SYSPLANDEP	SYSIBM.DSNGGX01	DP
2.4.0	1	IX scan MC=2	SYSIBM.SYSTABLESPACE	SYSIBM.DSNDSX01	
3.1.0	0	TS scan	GH12.GH12T17		SP
4.1.0	0	TS scan	GH12.GH12T17		SP
4.2.0	1	IX scan MC=1	SYSIBM.SYSPLAN	SYSIBM.DSNPPH01	
4.3.0	1	IX scan, MC=0	SYSIBM.SYSPLANDEP	SYSIBM.DSNGGX01	DP
4.4.0	1	IX scan MC=2	SYSIBM.SYSINDEXES	SYSIBM.DSNDXX01	
5.1.0	0	TS scan	GH12.GH12T17		SP

Worklist
Show Worklist

Worklist

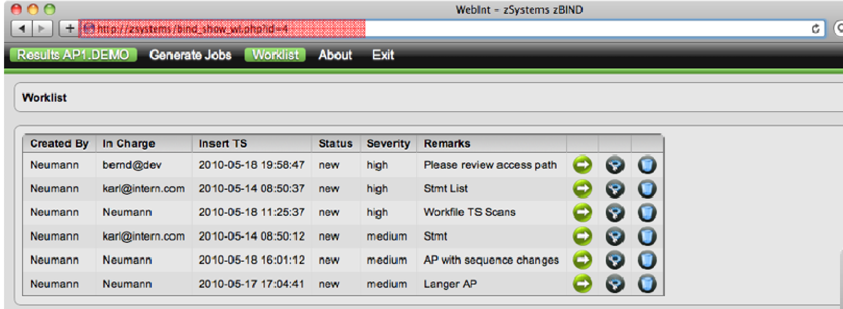


Worklist

Created By	In Charge	Insert TS	Status	Severity	Remarks
Neumann	bernd@dev	2010-05-18 19:58:47	new	high	Please review access path
Neumann	karl@intern.com	2010-05-14 08:50:37	new	high	Stmt List
Neumann	Neumann	2010-05-18 11:25:37	new	high	Workfile TS Scans
Neumann	karl@intern.com	2010-05-14 08:50:12	new	medium	Stmt
Neumann	Neumann	2010-05-18 16:01:12	new	medium	AP with sequence changes
Neumann	Neumann	2010-05-17 17:04:41	new	medium	Langer AP

Worklist allows direct access to the stored results

Send Worklist As Link



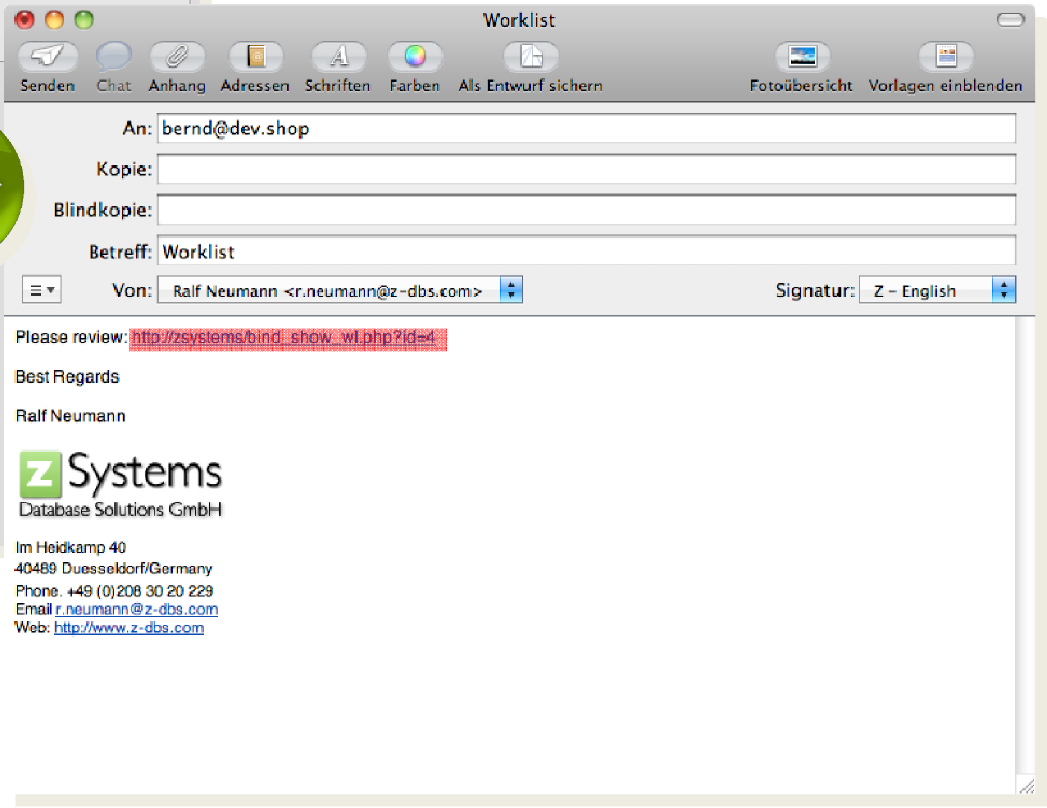
WebInt - zSystems zBIND

http://zsystems/bind_show_wl.php?id=4

Worklist

Created By	In Charge	Insert TS	Status	Severity	Remarks			
Neumann	bernd@dev	2010-05-18 19:58:47	new	high	Please review access path	➡	●	●
Neumann	karl@intern.com	2010-05-14 08:50:37	new	high	Stmt List	➡	●	●
Neumann	Neumann	2010-05-18 11:25:37	new	high	Workfile TS Scans	➡	●	●
Neumann	karl@intern.com	2010-05-14 08:50:12	new	medium	Stmt	➡	●	●
Neumann	Neumann	2010-05-18 16:01:12	new	medium	AP with sequence changes	➡	●	●
Neumann	Neumann	2010-05-17 17:04:41	new	medium	Langer AP	➡	●	●

Z APX user Interface is web-based, so all pages may be stored as bookmark in the worklist and provided to person in charge by email.



Worklist

An: bernd@dev.shop

Kopie:

Blindkopie:

Betreff: Worklist

Von: Ralf Neumann <r.neumann@z-dbs.com>

Signatur: Z - English

Please review: http://zsystems/bind_show_wl.php?id=4

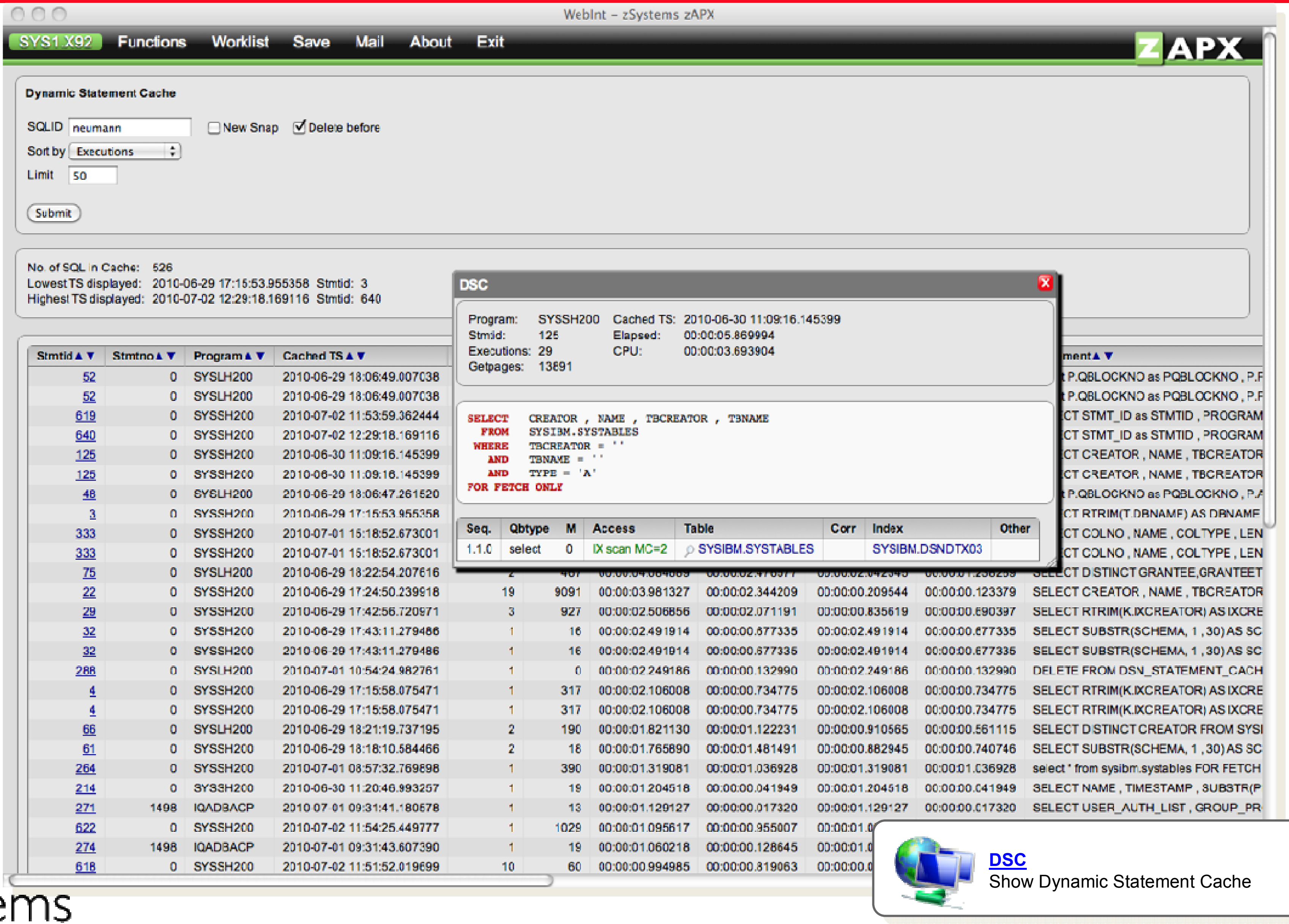
Best Regards

Ralf Neumann

Z Systems
Database Solutions GmbH

Im Heidkamp 40
40489 Duesseldorf/Germany
Phone. +49 (0)208 30 20 229
Email r.neumann@z-dbs.com
Web: http://www.z-dbs.com

Dynamic SQL



Dynamic Statement Cache

SQLID neumann □ New Snap ☑ Delete before

Sort by Executions

Limit 50

Submit

No. of SQL in Cache: 526
Lowest TS displayed: 2010-06-29 17:15:53.955358 Stmtid: 3
Highest TS displayed: 2010-07-02 12:29:18.169116 Stmtid: 640

Stmtid ▲ ▼	Stmtno ▲ ▼	Program ▲ ▼	Cached TS ▲ ▼							ment ▲ ▼
52	0	SYSLH200	2010-06-29 18:06:49.007038							P.QBLOCKNO as PQBLOCKNO , P.F
52	0	SYSLH200	2010-06-29 18:06:49.007038							P.QBLOCKNO as PQBLOCKNO , P.F
619	0	SYSSH200	2010-07-02 11:53:59.362444							CT STMT_ID as STMTID , PROGRAM
640	0	SYSSH200	2010-07-02 12:29:18.169116							CT STMT_ID as STMTID , PROGRAM
125	0	SYSSH200	2010-06-30 11:09:16.145399							CT CREATOR , NAME , TBCREATOR
125	0	SYSSH200	2010-06-30 11:09:16.145399							CT CREATOR , NAME , TBCREATOR
48	0	SYSLH200	2010-06-29 18:06:47.261520							P.QBLOCKNO as PQBLOCKNO , P.A
3	0	SYSSH200	2010-06-29 17:15:53.955358							CT RTRIM(T.DBNAME) AS DBNAME
333	0	SYSSH200	2010-07-01 15:18:52.673001							CT COLNO , NAME , COLTYPE , LEN
333	0	SYSSH200	2010-07-01 15:18:52.673001							CT COLNO , NAME , COLTYPE , LEN
75	0	SYSLH200	2010-06-29 18:22:54.207616	2						SELECT DISTINCT GRANTEE,GRANTEET
22	0	SYSSH200	2010-06-29 17:24:50.239918	19	9091	00:00:03.981327	00:00:02.344209	00:00:00.209544	00:00:00.123379	SELECT CREATOR , NAME , TBCREATOR
29	0	SYSSH200	2010-06-29 17:42:56.720971	3	927	00:00:02.506856	00:00:02.071191	00:00:00.835619	00:00:00.690397	SELECT RTRIM(K.IXCREATOR) AS IXCRE
32	0	SYSSH200	2010-06-29 17:43:11.279486	1	16	00:00:02.491914	00:00:00.677335	00:00:02.491914	00:00:00.677335	SELECT SUBSTR(SCHEMA, 1 ,30) AS SC
32	0	SYSSH200	2010-06-29 17:43:11.279486	1	16	00:00:02.491914	00:00:00.677335	00:00:02.491914	00:00:00.677335	SELECT SUBSTR(SCHEMA, 1 ,30) AS SC
288	0	SYSLH200	2010-07-01 10:54:24.982761	1	0	00:00:02.249186	00:00:00.132990	00:00:02.249186	00:00:00.132990	DELETE FROM DSN_STATEMENT_CACH
4	0	SYSSH200	2010-06-29 17:15:58.075471	1	317	00:00:02.106008	00:00:00.734775	00:00:02.106008	00:00:00.734775	SELECT RTRIM(K.IXCREATOR) AS IXCRE
4	0	SYSSH200	2010-06-29 17:15:58.075471	1	317	00:00:02.106008	00:00:00.734775	00:00:02.106008	00:00:00.734775	SELECT RTRIM(K.IXCREATOR) AS IXCRE
66	0	SYSLH200	2010-06-29 18:21:19.737195	2	190	00:00:01.821130	00:00:01.122231	00:00:00.910565	00:00:00.561115	SELECT DISTINCT CREATOR FROM SYSI
61	0	SYSSH200	2010-06-29 18:18:10.584466	2	18	00:00:01.765890	00:00:01.481491	00:00:00.882945	00:00:00.740746	SELECT SUBSTR(SCHEMA, 1 ,30) AS SC
264	0	SYSSH200	2010-07-01 08:57:32.769698	1	390	00:00:01.319081	00:00:01.036928	00:00:01.319081	00:00:01.036928	select * from sysibm.systables FOR FETCH
214	0	SYSSH200	2010-06-30 11:20:46.993257	1	19	00:00:01.204518	00:00:00.041949	00:00:01.204518	00:00:00.041949	SELECT NAME , TIMESTAMP , SUBSTR(P
271	1498	IQADBACP	2010-07-01 09:31:41.180678	1	13	00:00:01.129127	00:00:00.017320	00:00:01.129127	00:00:00.017320	SELECT USER_AUTH_LIST , GROUP_PR
622	0	SYSSH200	2010-07-02 11:54:25.449777	1	1029	00:00:01.095617	00:00:00.955007	00:00:01.0		
274	1498	IQADBACP	2010-07-01 09:31:43.607390	1	19	00:00:01.060218	00:00:00.128645	00:00:01.0		
618	0	SYSSH200	2010-07-02 11:51:52.019699	10	60	00:00:00.994985	00:00:00.819063	00:00:00.0		

DSC

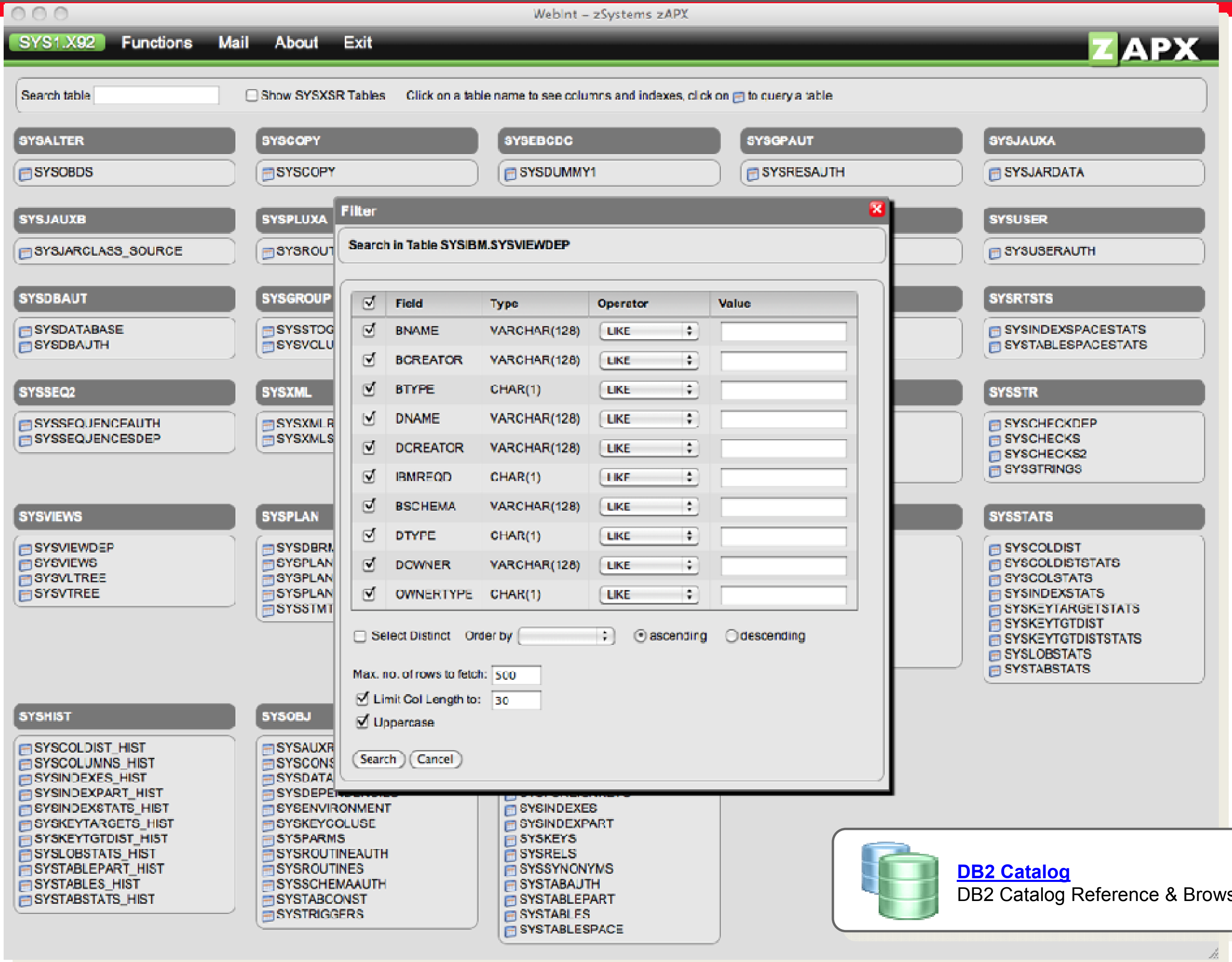
Program: SYSSH200 Cached TS: 2010-06-30 11:09:16.145399
Stmtid: 125 Elapsed: 00:00:05.869994
Executions: 29 CPU: 00:00:03.693904
Getpages: 13891

```
SELECT  CREATOR , NAME , TBCREATOR , TBNAME
  FROM  SYSIBM.SYSTABLES
 WHERE  TBCREATOR = ''
   AND  TBNAME = ''
   AND  TYPE = 'A'
FOR FETCH ONLY
```

Seq.	Qbtype	M	Access	Table	Corr	Index	Other
1.1.0	select	0	IX scan MC=2	SYSIBM.SYSTABLES		SYSIBM.DSNDTX03	

DSC
Show Dynamic Statement Cache

Browse the Catalog



WebInt – zSystems zAPX

Search table □ Show SYSXSR Tables Click on a table name to see columns and indexes, click on ▤ to query a table

SYSALTER
- SYSOBDS

SYSCOPY
- SYSCOPY

SYSEBCDC
- SYSDUMMY1

SYSGPAUT
- SYSRESAUTH

SYSJAUXA
- SYSJARDATA

SYSJAUXB
- SYSJARCLASS_SOURCE

SYSPLUXA
- SYSROUT

SYSUSER
- SYSUSERAUTH

SYSDBAUT
- SYSDATABASE
- SYSDBAUTH

SYSGROUP
- SYSSTOG
- SYSVOLU

SYSRTSTS
- SYSINDEXSPACESTATS
- SYSTABLESPACESTATS

SYSSEQ2
- SYSSEQUENCEAUTH
- SYSSEQUENCESDEP

SYSXML
- SYSXMLR
- SYSXMLS

SYSSTR
- SYSCHECKDEP
- SYSCHECKS
- SYSCHECKS2
- SYSSTRINGS

SYSVIEWS
- SYSVIEWDEP
- SYSVIEWS
- SYSVLTREE
- SYSVTREE

SYSPLAN
- SYSDBRM
- SYSPLAN
- SYSPLAN
- SYSPLAN
- SYSSTMT

SYSSTATS
- SYSCOLDIST
- SYSCOLDISTSTATS
- SYSCOLSTATS
- SYSINDEXSTATS
- SYSKEYTARGETSTATS
- SYSKEYTGTDIST
- SYSKEYTGTDISTSTATS
- SYSLOBSTATS
- SYSTABSTATS

SYSHIST
- SYSCOLDIST_HIST
- SYSCOLUMNS_HIST
- SYSINDEXES_HIST
- SYSINDEXPART_HIST
- SYSINDEXSTATS_HIST
- SYSKEYTARGETS_HIST
- SYSKEYTGTDIST_HIST
- SYSLOBSTATS_HIST
- SYSTABLEPART_HIST
- SYSTABLES_HIST
- SYSTABSTATS_HIST

SYSOBJ
- SYSAUXR
- SYSCONS
- SYSDATA
- SYSDEPE
- SYSENVIRONMENT
- SYSKEYCOLUSE
- SYSPARMS
- SYSROUTINEAUTH
- SYSROUTINES
- SYSSCHEMAAUTH
- SYSTABCONST
- SYSTRIGGERS

- SYSINDEXES
- SYSINDEXPART
- SYSKEYS
- SYSRELS
- SYSSYNONYMS
- SYSTABAUTH
- SYSTABLEPART
- SYSTABLES
- SYSTABLESPACE

Filter

Search in Table SYSIBM.SYSVIEWDEP

☑	Field	Type	Operator	Value
☑	BNAME	VARCHAR(128)	LIKE	
☑	BCREATOR	VARCHAR(128)	LIKE	
☑	BTYPE	CHAR(1)	LIKE	
☑	DNAME	VARCHAR(128)	LIKE	
☑	DCREATOR	VARCHAR(128)	LIKE	
☑	IBMREQD	CHAR(1)	LIKE	
☑	BSCHEMA	VARCHAR(128)	LIKE	
☑	DTYPE	CHAR(1)	LIKE	
☑	DOWNER	VARCHAR(128)	LIKE	
☑	OWNERTYPE	CHAR(1)	LIKE	

□ Select Distinct Order by ● ascending ○ descending

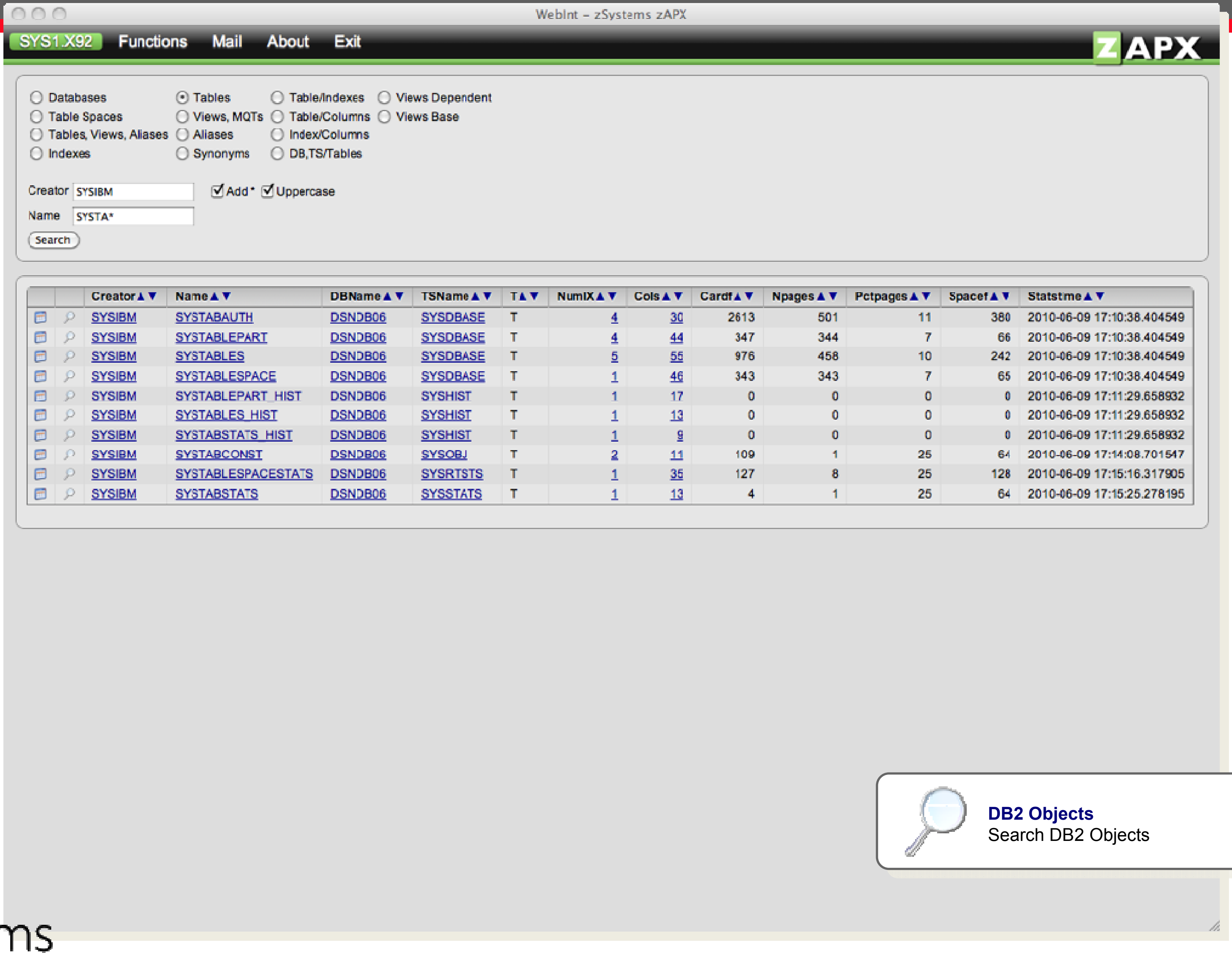
Max. no. of rows to fetch: 500

☑ Limit Col Length to: 30

☑ Uppercase

DB2 Catalog
DB2 Catalog Reference & Browser

Query the Catalog



○ Databases ● Tables ○ Table/Indexes ○ Views Dependent
○ Table Spaces ○ Views, MQTs ○ Table/Columns ○ Views Base
○ Tables, Views, Aliases ○ Aliases ○ Index/Columns
○ Indexes ○ Synonyms ○ DB,TS/Tables

Creator SYSIBM ☑ Add* ☑ Uppercase

Name SYSTA*

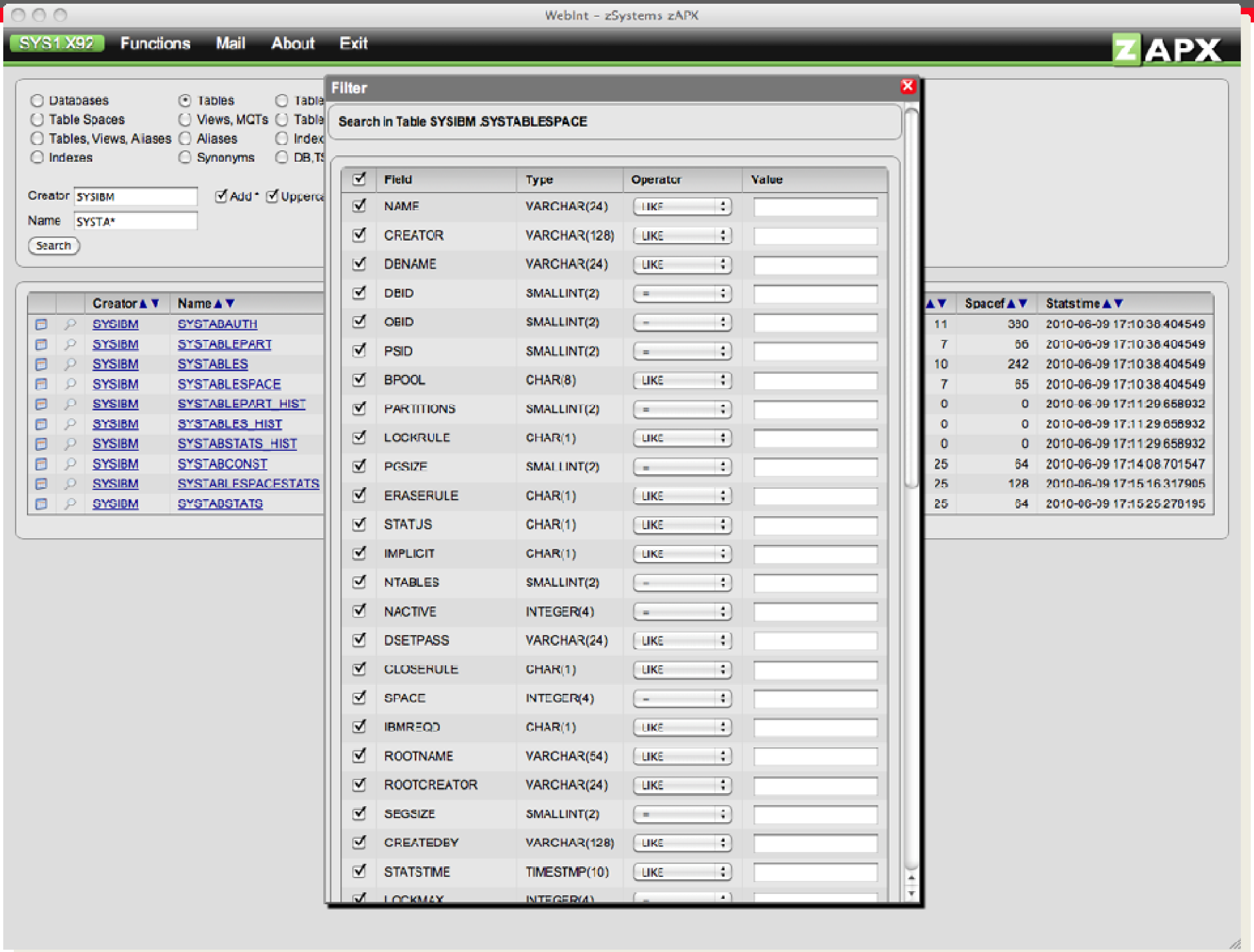
Search

		Creator▲▼	Name▲▼	DBName▲▼	TSName▲▼	T▲▼	NumIX▲▼	Cols▲▼	Cardf▲▼	Npages▲▼	Pctpages▲▼	Spacef▲▼	Statstme▲▼
▤	🔍	SYSIBM	SYSTABAUTH	DSNDB06	SYSDBASE	T	4	30	2613	501	11	380	2010-06-09 17:10:38.404549
▤	🔍	SYSIBM	SYSTABLEPART	DSNDB06	SYSDBASE	T	4	44	347	344	7	66	2010-06-09 17:10:38.404549
▤	🔍	SYSIBM	SYSTABLES	DSNDB06	SYSDBASE	T	5	55	976	458	10	242	2010-06-09 17:10:38.404549
▤	🔍	SYSIBM	SYSTABLESPACE	DSNDB06	SYSDBASE	T	1	46	343	343	7	65	2010-06-09 17:10:38.404549
▤	🔍	SYSIBM	SYSTABLEPART_HIST	DSNDB06	SYSHIST	T	1	17	0	0	0	0	2010-06-09 17:11:29.658932
▤	🔍	SYSIBM	SYSTABLES_HIST	DSNDB06	SYSHIST	T	1	13	0	0	0	0	2010-06-09 17:11:29.658932
▤	🔍	SYSIBM	SYSTABSTATS_HIST	DSNDB06	SYSHIST	T	1	9	0	0	0	0	2010-06-09 17:11:29.658932
▤	🔍	SYSIBM	SYSTABCONST	DSNDB06	SYSOBJ	T	2	11	109	1	25	64	2010-06-09 17:14:08.701547
▤	🔍	SYSIBM	SYSTABLESPACESTATS	DSNDB06	SYSRTSTS	T	1	35	127	8	25	128	2010-06-09 17:15:16.317905
▤	🔍	SYSIBM	SYSTABSTATS	DSNDB06	SYSSTATS	T	1	13	4	1	25	64	2010-06-09 17:15:25.278195

DB2 Objects
Search DB2 Objects

Query Any Table

softwareonz
...a better way on z



WebInt – zSystems zAPX

SYS1.X92 Functions Mail About Exit ZAPX

○ Databases ● Tables ○ Table
○ Table Spaces ○ Views, MQTs ○ Table
○ Tables, Views, Aliases ○ Aliases ○ Index
○ Indexes ○ Synonyms ○ DB,TS

Creator SYSIBM ☑ Add * ☑ Upperca
Name SYSTA*
Search

Creator▲▼	Name▲▼	▲▼	Spacef▲▼	Statstime▲▼
SYSIBM	SYSTABAUTH	11	330	2010-06-09 17:10:38.404549
SYSIBM	SYSTABLEPART	7	56	2010-06-09 17:10:38.404549
SYSIBM	SYSTABLES	10	242	2010-06-09 17:10:38.404549
SYSIBM	SYSTABLESPACE	7	65	2010-06-09 17:10:38.404549
SYSIBM	SYSTABLEPART_HIST	0	0	2010-06-09 17:11:29.658932
SYSIBM	SYSTABLES_HIST	0	0	2010-06-09 17:11:29.658932
SYSIBM	SYSTABSTATS_HIST	0	0	2010-06-09 17:11:29.658932
SYSIBM	SYSTABCONST	25	64	2010-06-09 17:14:08.701547
SYSIBM	SYSTABLESPACESTATS	25	128	2010-06-09 17:15:16.317905
SYSIBM	SYSTABSTATS	25	64	2010-06-09 17:15:25.270195

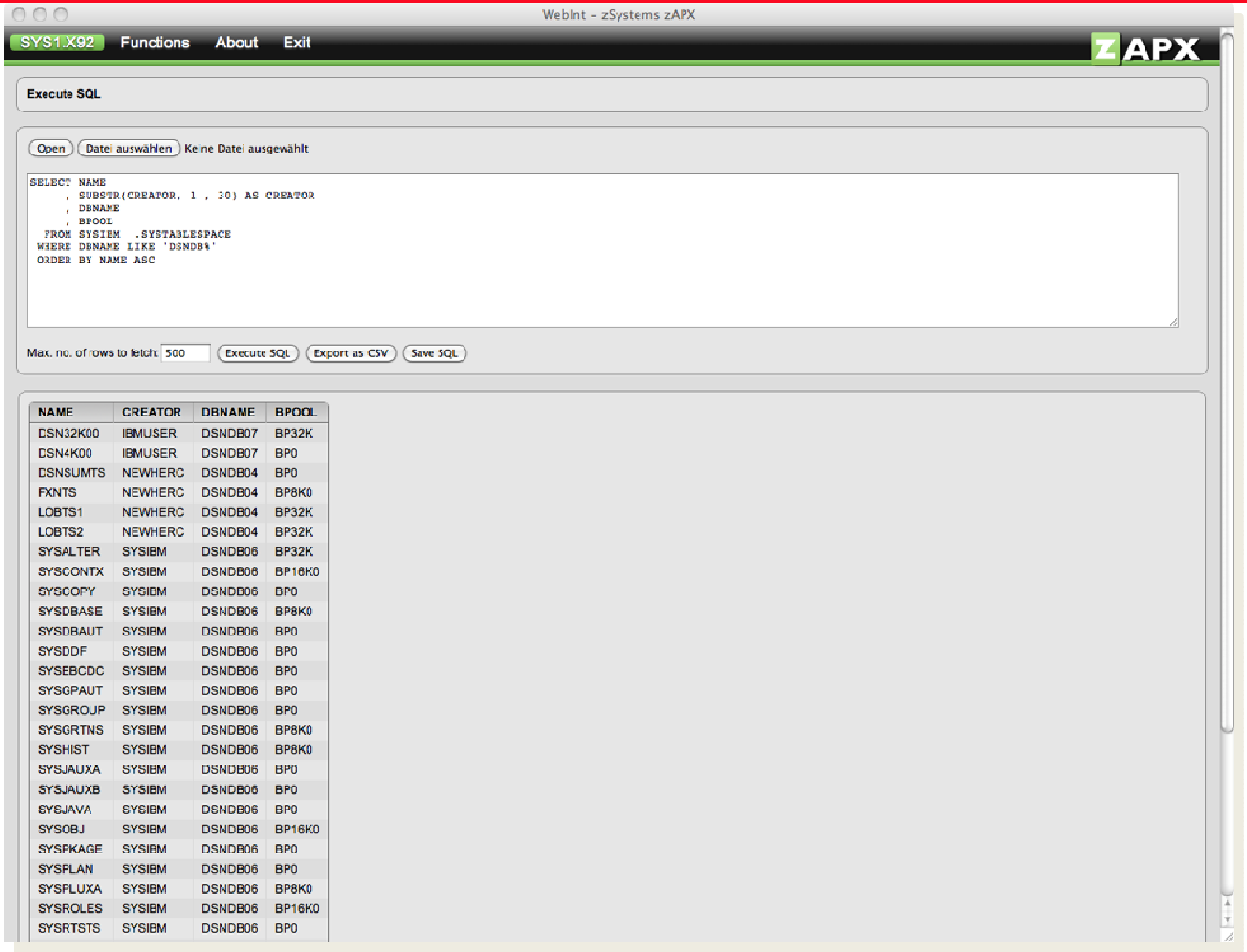
Filter

Search in Table SYSIBM.SYSTABLESPACE

☑	Field	Type	Operator	Value
☑	NAME	VARCHAR(24)	LIKE	
☑	CREATOR	VARCHAR(128)	LIKE	
☑	DBNAME	VARCHAR(24)	LIKE	
☑	DBID	SMALLINT(2)	=	
☑	OBID	SMALLINT(2)	=	
☑	PSID	SMALLINT(2)	=	
☑	BPOOL	CHAR(8)	LIKE	
☑	PARTITIONS	SMALLINT(2)	=	
☑	LOCKRULE	CHAR(1)	LIKE	
☑	PGSIZE	SMALLINT(2)	=	
☑	ERASERULE	CHAR(1)	LIKE	
☑	STATUS	CHAR(1)	LIKE	
☑	IMPLICIT	CHAR(1)	LIKE	
☑	NTABLES	SMALLINT(2)	=	
☑	NACTIVE	INTEGER(4)	=	
☑	DSETPASS	VARCHAR(24)	LIKE	
☑	CLOSERULE	CHAR(1)	LIKE	
☑	SPACE	INTEGER(4)	=	
☑	IBMREQD	CHAR(1)	LIKE	
☑	ROOTNAME	VARCHAR(54)	LIKE	
☑	ROOTCREATOR	VARCHAR(24)	LIKE	
☑	SEGSIZE	SMALLINT(2)	=	
☑	CREATEDBY	VARCHAR(128)	LIKE	
☑	STATSTIME	TIMESTMP(10)	LIKE	
☑	LOCKMAX	INTEGER(4)	=	

zSystems
Database Solutions GmbH

Execute SQL



```
SELECT NAME
     , SUBSTR(CREATOR, 1 , 30) AS CREATOR
     , DBNAME
     , BPOOL
  FROM SYSIBM  .SYSTABLESPACE
 WHERE DBNAME LIKE 'DSNDB%'
 ORDER BY NAME ASC
```

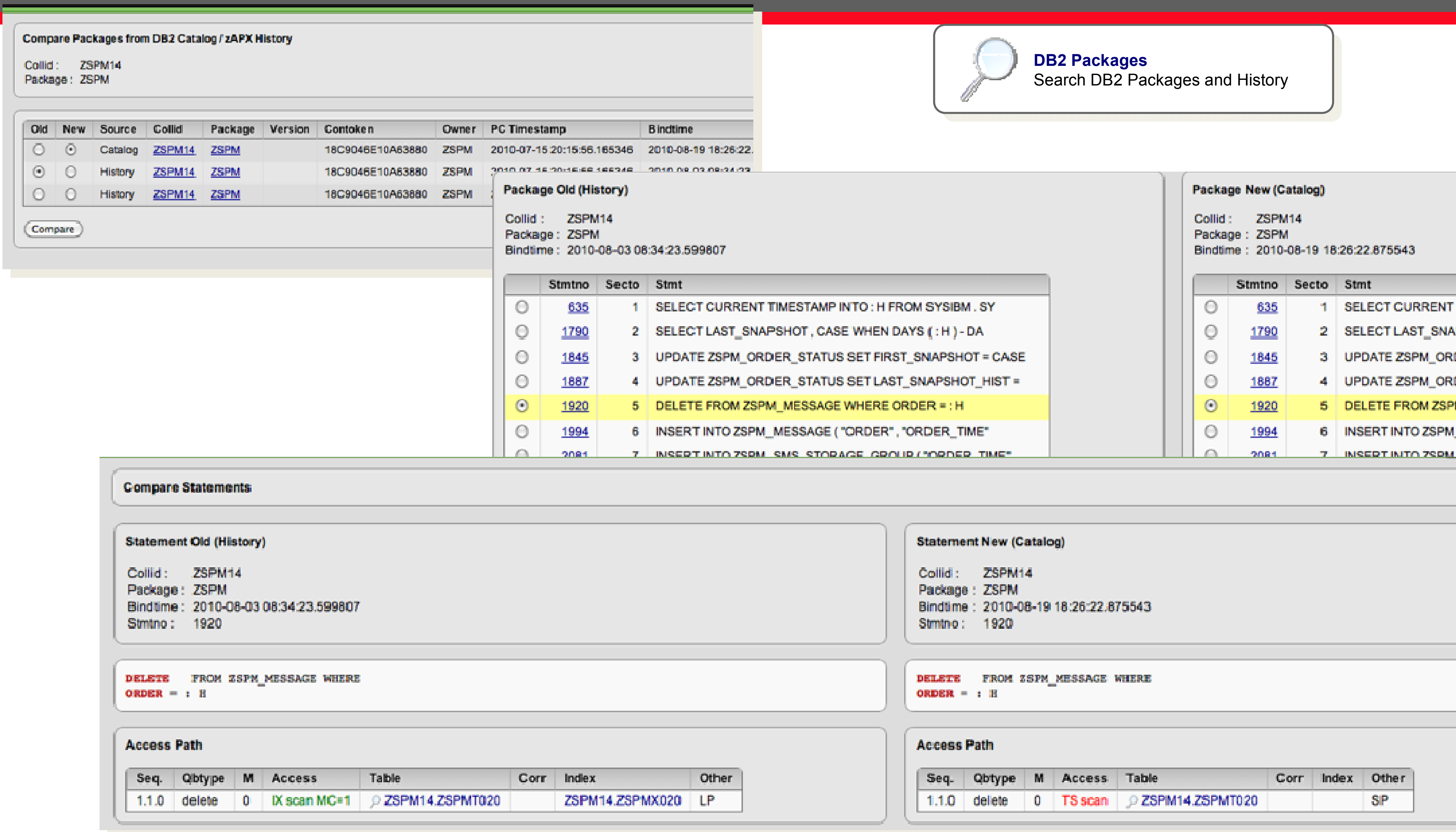
Max. no. of rows to fetch: 500

NAME	CREATOR	DBNAME	BPOOL
DSN32K00	IBMUSER	DSNDB07	BP32K
DSN4K00	IBMUSER	DSNDB07	BP0
DSNSUMTS	NEWHERC	DSNDB04	BP0
FXNTS	NEWHERC	DSNDB04	BP8K0
LOBTS1	NEWHERC	DSNDB04	BP32K
LOBTS2	NEWHERC	DSNDB04	BP32K
SYSALTER	SYSIBM	DSNDB06	BP32K
SYSCONTX	SYSIBM	DSNDB06	BP16K0
SYSCOPY	SYSIBM	DSNDB06	BP0
SYSDBASE	SYSIBM	DSNDB06	BP8K0
SYSDBAUT	SYSIBM	DSNDB06	BP0
SYSDDF	SYSIBM	DSNDB06	BP0
SYSEBCDC	SYSIBM	DSNDB06	BP0
SYSGPAUT	SYSIBM	DSNDB06	BP0
SYSGROUP	SYSIBM	DSNDB06	BP0
SYSGRTNS	SYSIBM	DSNDB06	BP8K0
SYSHIST	SYSIBM	DSNDB06	BP8K0
SYSJAUXA	SYSIBM	DSNDB06	BP0
SYSJAUXB	SYSIBM	DSNDB06	BP0
SYSJAVA	SYSIBM	DSNDB06	BP0
SYSOBJ	SYSIBM	DSNDB06	BP16K0
SYSPKAGE	SYSIBM	DSNDB06	BP0
SYSPLAN	SYSIBM	DSNDB06	BP0
SYSPLUXA	SYSIBM	DSNDB06	BP8K0
SYSROLES	SYSIBM	DSNDB06	BP16K0
SYSRTSTS	SYSIBM	DSNDB06	BP0

Packages & History

softwareonz
...a better way on z

DB2 Packages
Search DB2 Packages and History



Compare Packages from DB2 Catalog / zAPX History

Collid : ZSPM14
Package : ZSPM

Old	New	Source	Collid	Package	Version	Contoken	Owner	PC Timestamp	Bindtime
○	●	Catalog	ZSPM14	ZSPM		18C9046E10A63880	ZSPM	2010-07-15 20:15:56.165346	2010-08-19 18:26:22.
●	○	History	ZSPM14	ZSPM		18C9046E10A63880	ZSPM	2010-07-15 20:15:56.165346	2010-08-03 08:34:23
○	○	History	ZSPM14	ZSPM		18C9046E10A63880	ZSPM		

Package Old (History)

Collid : ZSPM14
Package : ZSPM
Bindtime : 2010-08-03 08:34:23.599807

	Stmtno	Secto	Stmt
○	635	1	SELECT CURRENT TIMESTAMP INTO : H FROM SYSIBM . SY
○	1790	2	SELECT LAST_SNAPSHOT , CASE WHEN DAYS (: H) - DA
○	1845	3	UPDATE ZSPM_ORDER_STATUS SET FIRST_SNAPSHOT = CASE
○	1887	4	UPDATE ZSPM_ORDER_STATUS SET LAST_SNAPSHOT_HIST =
●	1920	5	DELETE FROM ZSPM_MESSAGE WHERE ORDER = : H
○	1994	6	INSERT INTO ZSPM_MESSAGE ("ORDER" , "ORDER_TIME"
○	2081	7	INSERT INTO ZSPM_SMS_STORAGE_GROUP ("ORDER_TIME"

Package New (Catalog)

Collid : ZSPM14
Package : ZSPM
Bindtime : 2010-08-19 18:26:22.875543

	Stmtno	Secto	Stmt
○	635	1	SELECT CURRENT
○	1790	2	SELECT LAST_SNA
○	1845	3	UPDATE ZSPM_ORI
○	1887	4	UPDATE ZSPM_ORI
●	1920	5	DELETE FROM ZSPI
○	1994	6	INSERT INTO ZSPM
○	2081	7	INSERT INTO ZSPM

Compare Statements

Statement Old (History)

Collid : ZSPM14
Package : ZSPM
Bindtime : 2010-08-03 08:34:23.599807
Stmtno : 1920

```
DELETE  FROM ZSPM_MESSAGE WHERE
ORDER = : H
```

Access Path

Seq.	Qbtype	M	Access	Table	Corr	Index	Other
1.1.0	delete	0	IX scan MC=1	ZSPM14.ZSPMT020		ZSPM14.ZSPMX020	LP

Statement New (Catalog)

Collid : ZSPM14
Package : ZSPM
Bindtime : 2010-08-19 18:26:22.875543
Stmtno : 1920

```
DELETE  FROM ZSPM_MESSAGE WHERE
ORDER = : H
```

Access Path

Seq.	Qbtype	M	Access	Table	Corr	Index	Other
1.1.0	delete	0	TS scan	ZSPM14.ZSPMT020			SP

zSystems
Database Solutions GmbH

To Summarize

- Understand your BIND | REBIND parameters
  - And their impact on optimization & performance
- Move from the 3 R's to the 5 R's
  - RTS, REORG, RUNSTATS, REBIND, Recheck
- Control access path changes
  - Analysis and checking is important
  - Avoid binding when access paths will degrade
- Adopt automated solutions
- Questions?
  - Craig S. Mullins: Craig.Mullins@softwareonz.com
  - Ralf Neumann: r.neumann@z-dbs.com

For More Information

softwareonz
...a better way on z

- To learn more about Z APX or to start a trial

In the USA & Canada

softwareonz
...a better way on z

http://www.softwareonz.com
info@softwareonz.com

In Europe and ROW

Z Systems
Database Solutions GmbH

http://www.z-dbs.com
info@z-dbs.com

Don’t forget to register for the remaining three Tuesday webinars in this series!
http://www.softwareonz.com/index_files/Page560.htm