



**DBA**  
SISTEMI

SOFTWARE PER I SISTEMI INFORMATICI

 QuickSelect



# QuickSelect® for Db2 \*

**A performance improvement tool for DB2 on z/OS  
Installs easily and saves immediately SQL, I/O and CPU**

\* QuickSelect supports DB2, SQL for Batch, TSO, CICS and IMS/TM.

## **Do you want to save CPU?**

Quick Select will save significant amounts of CPU and IO and let DB2 perform other tasks

## **Do you want to lower your CPU peaks?**

QuickSelect will save CPU and IO when it counts most – during your peak hours

## **Do you want to do this without changing ANYTHING in your DB2 applications?**

QuickSelect works WITHOUT having to change anything in your applications or DB2

## QuickSelect® for DB2

- Why QuickSelect?
- With QuickSelect you'll save!
- What Does QuickSelect Do?
- How does QuickSelect work?
- Implementation highlights

## **Why QuickSelect?**

**Most of our customers face the following challenges:**

- You want to reduce the costs of your computer resources
- You want to improve the performance of your IT systems while avoiding the risks of touching the applications and databases themselves
- Database knowledge gets lost due to employee retirement and turnaround, and through mergers and acquisitions - leaving your organization with difficult and expensive choices for improvements and maintenance
- QuickSelect provides risk-free performance improvements
- QuickSelect works transparently, concentrating exclusively on HOW the data is used, rather than on what the application does
- QuickSelect is installed dynamically and makes **zero changes** in existing systems

## With QuickSelect you will save!

- **MONEY !!!** – start saving money immediately after QuickSelect goes online – no wait time, no tuning time, no risk to your data's integrity, no other negative impact on system performance
- **I/O** – reduce database I/O – QuickSelect loads DB2 result sets of frequently executed SQL's into its own memory, thereby greatly reducing repetitive I/O access and spent CPU on those SQL's
- **CPU** – reduce your peak (lower your high watermark!) and average CPU utilization levels, thereby reducing the footprint of your applications
- **Elapsed time** - shortens the time needed for batch jobs to run
- Improves response times in online systems
- **License costs for existing software** – get improved performance out of your existing resources – reduce the cost for existing licenses
- **Upgrade costs for existing software** – since QuickSelect improves the software performance, upgrading may be deferred
- Save **development efforts** for improving applications: Bypass the classic ailments of software improvement by installing a plug-in that will expedite the performance and throughput of your existing software without touching the existing code
- Get more throughput from current system – improved CPU utilization, improved software performance – means improved overall system throughput without investing in hardware or software upgrades
- Savings also in Test and QA environment – get better performance from the limited (capped) resources allocated to these environments
  - Improve the veracity of the tests by running more lifelike tests
  - Consume less resources
  - Run more tests in the same given timeframe
- Free up the teams for their routine work, reduce the risky, time-consuming process of trying to make improvements to existing code
- QuickSelect is an incredibly quick solution – a plug-in: quick installation, quick start-up, quick payback
- Bypass fixing “old, unknown, inherited” applications – QuickSelect rejuvenates system performance and frees you from having to confront the following challenges and the risks involved in:
  - Aging software systems contain old, inherited portions that are difficult to maintain or improve
  - Knowledge has been lost due to personnel turn-around and retirement
  - Old undocumented code cannot be deciphered
  - Archaic software languages in legacy systems stymie maintenance and improvement efforts

## What Does QuickSelect Do?

- Saves on CPU utilization and on database I/O for programs which access the same table data very frequently – up to 85% CPU and I/O. Frequently accessed reference tables and policy tables are loaded into memory, freeing up the CPU and DB2 for other work
- Frees DB2 to serve other requests and use buffer pools and CPU for other tasks
- Built-in LEARNMODE - automatically learns which tables and SQL are frequently accessed and should therefore be cached. The customer does not need to study and prepare a list of tables and SQL's to be cached: **QuickSelect** does this for you highly accurately and quickly. It outputs a list of tables, and from this list the customer can confirm or reject tables to be cached
- Built-in table update sensitivity in order to ensure data integrity, guaranteeing that **QuickSelect** never runs on outdated tables: triggers are defined for cached tables based on the customer-approved list of tables to be cached. Whenever a table is changed, the database informs QuickSelect that a change has been made and QuickSelect invalidates the cache. A table without a trigger defined is never loaded to cache, protecting against the risk of running on outdated data
- Full support for data sharing across DB2Plex: **QuickSelect** is installed on each of the servers in the DB2Plex and communicate with each other through a standard XCF facility. Instead of DB2 giving the answer, QuickSelect does it faster, saving on DB2 times, elapsed time
- QuickSelect always returns the same answer and/or behaves exactly as DB2 would – it is transparent to the rest of the system

## **How does QuickSelect work ?**

- QuickSelect **returns the same data DB2 would** – but from its cache – cutting CPU, I/O and elapsed time!
- QuickSelect interfaces between the application and DB2, but is totally transparent to the application and DB2. QuickSelect front-ends program calls to DB2 with **minimal overhead** and checks eligibility of the SQL for caching: only relevant SQLs, which are beneficial from the performance standpoint, are cached
- QuickSelect retains the RESULT of a SQL call in memory **above the bar** (64 bit addressing mode) – it does not interfere with any other existing data – it incurs zero overhead on DB2

## **QuickSelect implementation highlights:**

- **Minimal installation time:** end-to-end installation within **1-2 workdays**
- **Requires no application changes**, no upfront and tedious application analysis, and works with a minimal overhead
- **Non-intrusive installation** – QuickSelect is installed via SMP/E, including an IVP for Batch, CICS and IMS/TM – **no compile or re-link is required** – QuickSelect has a dynamic hook for all environments and it is possible to roll-out QuickSelect in a staggered fashion (Job, Jobstep, CICS Transaction(s), PSB's, Programs & Tables!)
- Confirm suggested tables for caching from a automatically proposed list – and define QuickSelect Trigger definitions for selected tables in order to activate QuickSelect and ensure **update sensitivity**
- Want to compare a job's performance with and without QuickSelect? - Run the job / Transaction with and without QuickSelect and compare the results
- QuickSelect works in a **multiple LPAR environment**

For information please visit [www.dbasistemi.it](http://www.dbasistemi.it) or email [info@dbasistemi.it](mailto:info@dbasistemi.it)

**DBA Sistemi Srl** - Software per i Sistemi Informatici

Head Office: Via della Casanova 3 - 50012 Bagno a Ripoli (Firenze); Tel.+39 055 600636

North Italy Branch: Via Baranzate 47A - 20026 Novate Milanese (Milano)

e-mail: [info@dbasistemi.it](mailto:info@dbasistemi.it); web: [www.dbasistemi.it](http://www.dbasistemi.it)