⬛ConditionalBind for Db2 z/OS
Stop Unnecessary Binds

Avoid unnecessary binds

Optimize and accelerate the compile procedure

Reduce CPU load

Reduce locks in the catalog

Compatible with the Db2 Co-processor

⬛**ConditionalBind** stops unnecessary binds. ⬛**ConditionalBind** determines whether a DBRM contains changed SQL and skips the bind step for those that do not.

What happens with a pre-compile and a bind?

With a pre-compile, Db2 replaces all SQL in the source code with calls of Db2 subroutines, as well as syntax checking and INCLUDE resolutions. SQL statements, including the description of their host variables as well as DECLARE TABLE statements, are transferred into the DBRM. The assignment of the DBRMs to the program code happens using a timestamp, which is called a CONTOKEN that is written in the source listing and passed to Db2 at program execution. The individual SQL statements are assigned to the section number with their order within the source codes. If the STMTNO is self-coded in the SQL, the SQL with this number is assigned.
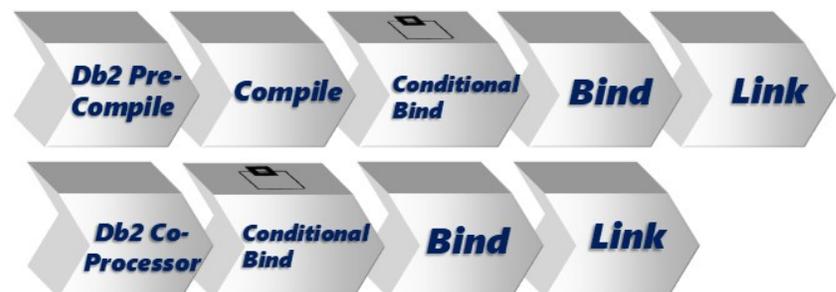
With a bind, Db2 transfers the SQL statement text, section number, statement and CONTOKEN in the catalog. The determined access paths are provided as executable code in the directory. If bound with option EXPLAIN(YES), the access paths are represented visibly in the PLAN_TABLE for the user. With a bind, S and X locks arise on different catalog tables and by doing this the isolation level is RS(read stability).
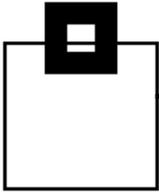
The bind is performed exclusively based on the DBRM. Db2 doesn't access the load module. The load module is merely necessary for the runtime of the program because Db2 does not need the DBRM anymore after the bind.

If no SQL changes were made in the source code, the SQL text and the SQL section numbers have also not changed. Therefore, a new bind is not necessary for the execution. ⬛**ConditionalBind** checks whether SQL changes took place and sets back the CONTOKEN if there are no SQL changes. Using condition code control, ⬛**ConditionalBind** automatically avoids the bind step.

ConditionalBind works as a subsequent step to the Db2 precompile and compile steps.

ConditionalBind supports the integrated Db2 coprocessor as well.

Related Solutions

Bind ImpactExpert

ConditionalBind belongs to the **Bind ImpactExpert** solution. This extended solution automatically determines the performance impact of BINDs and REBINDs on packages in advance.

Intégral Products of
SQL PerformanceExpert

- **Analyze+** performs expert evaluations of DML and DDL and provides tuning recommendations to promote effective application tuning and quality assurance.

- **DBRM/Checker**® determines compliance of your current quality assurance rules by verifying new programs during the compile procedure and analyzing whole DBRM libraries.

- **ProductionSimulator**® provides the user with a method to create Db2 catalog statistics in the test environment so that actual application access paths may be simulated during testing.

- **PackageManager** performs catalog house keeping and frees up superfluous packages.