

RLX for DB2

REXX Language Xtensions for DB2



- **Embed SQL statements in REXX execs**
- **Load SQL query results directly into ISPF tables and REXX stemmed arrays**
- **Issue DB2 commands, IFI requests and CAF services natively from REXX**
- **Compile and bind SQL execs to deploy industrial strength applications**
- **Develop NetView dialogs with an ISPF look and feel**

Overview

The suite of REXX Language eXtensions for DB2 provides the glue to integrate REXX, SQL, DB2, ISPF, NetView and the growing number of third party products that support REXX scripting.

- **RLX/SQL** lets you embed SQL statements, DB2 commands, CAF services and IFI requests within REXX execs and test them immediately. Without compile and link edit steps. Without preprocess and bind delays. Without even leaving ISPF Edit!
- **RLX/ISPF** provides powerful composite functions to load SQL query results directly into ISPF tables, then display and process those results on scrollable ISPF panels.
- **RLX/Compile** delivers industrial strength performance and security by compiling your REXX applications into load modules and static SQL plans. RLX/Compile enables DBAs to GRANT EXECUTE authority to specific plans rather than GRANT users access to underlying DB2 objects. This prevents unauthorized access to sensitive data through dynamic SQL facilities like QMF and SPUFI. RLX/Compile includes facilities to compile, link edit and bind 'real world' applications comprised of multiple modules.

RLX for DB2 makes REXX the ideal Rapid Application Development tool, with ample power to tackle the kinds of applications once written in COBOL, PL/1, C — even Assembler language!

- Build your own DBA utilities and developer workbenches
- Prototype high volume systems for CICS, IMS and batch
- Quickly develop small applications and one-time programs
- Deliver production applications under deadline pressure
- Automate system and network management with REXX scripts that access DB2
- Learn SQL with immediate feedback, extensive diagnostics and context sensitive help

RLX for DB2 runs everywhere REXX does including batch, TSO, ISPF, NetView and DB2 stored procedure address spaces. RLX exploits TSO, ISPF and NetView facilities when they are available but does not require them to operate. For example, RLX execs run natively in batch but use scrollable panels to display diagnostics in the TSO/ISPF foreground.

Benefits

- Provide DB2 developers with instant feedback and the expressive power of REXX
- Leverage expertise in REXX, SQL and DB2. RLX supports exactly the same 'embedded' SQL used with COBOL and PL/1, so there's no learning curve
- Shorten the development cycle to improve productivity and lower development costs
- Use CURSOR-based facilities such as OPEN, FETCH and CLOSE to develop dialogs more efficiently, flexibly and easily than with QMF or SPUIFI
- Develop applications with *far less* coding than with compiled or assembled languages
- Make use of the numerous utilities and sample execs supplied with RLX for DB2

Features

- Supports nulls, host variables and all DB2 datatypes
- Allows applications to process individual rows as well as complete SQL answer sets
- Supports new SQL and DB2 facilities as soon as they become available
- Converts between REXX and DB2 data formats automatically
- Works compatibly with all DB2 security and authorization mechanisms
- REXX execs, TSO CLISTs and compiled applications can invoke one another without restriction
- Returns feedback from the SQL Communications Area after each SQL statement executes
- Describes query result columns through a formatted SQLDA
- Full screen diagnostics pinpoint errors and speed their correction
- Sophisticated optimization techniques speed performance and maximize the reuse of SQL statements and DB2 threads
- Supports both RRSAP and CAF as DB2 attachment facilities
- RXSQL support lets you migrate REXX procedures from VM and OS/2 to MVS

RLX/SQL Facilities

You can embed SQL statements, DB2 commands, CAF services and IFI requests within your REXX execs. Subject to your authorization ID, you can:

- Issue queries (SELECT, DECLARE, OPEN, FETCH, CLOSE)
- Manipulate data (INSERT, UPDATE, DELETE)
- Manage data consistency and referential integrity (COMMIT and ROLLBACK as well as SRRCOMIT and SRRBACK)
- Call stored procedures and manage result sets (CALL, ASSOCIATE LOCATORS and ALLOCATE CURSOR)
- Define and revise DB2 objects (CREATE, ALTER, DROP)
- Administer authorizations and privileges (GRANT and REVOKE)
- Process dynamic SQL (PREPARE, DESCRIBE, EXECUTE)
- Connect to remote servers (CONNECT and RELEASE)
- Set and obtain values of SQL special registers (various forms of the SET statement)
- Copy SQL query results directly into ISPF tables and REXX stemmed arrays (RLX extensions)

In addition, you can natively issue DB2 commands and process the command output returned in REXX variables. You can manage DB2 connections and application plans explicitly through CAF services – even create and switch among multiple, concurrently active threads within the same REXX application. Lastly, RLX/SQL's native IFI support provides READ/WRITE access to DB2 trace records and all the statistics, accounting, audit, performance, monitor and global IFCIDs they contain, populated as REXX variables.

Prerequisites

RLX for DB2 is easily installed and administered.

It runs with all releases of z/OS and DB2 from z/OS whose IBM support status is "current".

Product names are the trademarks or registered trademarks of their respective holders.

```

----- Tables Created by SYSIBM ----- ROW 1 OF 30
Command ==>                               Scroll ==> HALF

```

Table Name	Owner	Type	DB Name	TS Name	DB ID	Colcount
SYSOCOPY	SYSIBM	T	DSNDB06	SYSOCOPY	6	14
SYSFIELDS	SYSIBM	T	DSNDB06	SYSDBAS	6	13
SYSTABLESPACE	SYSIBM	T	DSNDB06	SYSDBAS	6	23
SYSTABLES	SYSIBM	T	DSNDB06	SYSDBAS	6	31
SYSTABLEPART	SYSIBM	T	DSNDB06	SYSDBAS	6	21
SYSTABAUTH	SYSIBM	T	DSNDB06	SYSDBAS	6	21
SYSYNONYMS	SYSIBM	T	DSNDB06	SYSDBAS	6	6
SYSRELS	SYSIBM	T	DSNDB06	SYSDBAS	6	11
SYSLINKS	SYSIBM	T	DSNDB06	SYSDBAS	6	12
SYSKEYS	SYSIBM	T	DSNDB06	SYSDBAS	6	7
SYSINDEXPART	SYSIBM	T	DSNDB06	SYSDBAS	6	16
SYSINDEXES	SYSIBM	T	DSNDB06	SYSDBAS	6	26
SYSFOREIGNKEYS	SYSIBM	T	DSNDB06	SYSDBAS	6	7
SYSCOLUMNS	SYSIBM	T	DSNDB06	SYSDBAS	6	19
SYSCOLAUTH	SYSIBM	T	DSNDB06	SYSDBAS	6	10

Figure 1: Scrollable query result produced by the REXX execs in Figures 2 and 3

```

/* REXX */
arg createdby          /* Get creator ID as a parameter */
address RLX              (1)

"RLX declare tcreator cursor for
  select name, creator, type, dbname, tsname, dbid, colcount
  from sysibm.systables
  where createdby = :createdby" (2)
"tbcreate tblnames
  names(name, creator, type, dbname, tsname, dbid, colcount)
  nowrite" (3)

"RLX open tcreator"      /* Materialize the SQL query result */
"RLX fetch tcreator into
  :name, :creator, :type, :dbname, :tsname, :dbid, :colcount" (4)

Do while sqlcode = 0    /* While FETCHes are successful */ (5)
  "tbadd tblnames"
  "RLX fetch tcreator into
    :name, :creator, :type, :dbname, :tsname, :dbid, :colcount" (6)
End
"RLX close tcreator"    /* close the cursor */

"tbtop tblnames"
Do while rc = 0        /* until user signals end or return */
  "tbdispl tblnames panel(tblname)"
End
"tbend tblnames"
exit rc

```

Figure 2: Exec using standard SQL and ISPF dialog services

```

/* REXX */
arg createdby          /* Obtain the creator's name as a parameter */
address RLX            /* Route host commands to RLX for execution */

"RLX declare tcreator ispf table for" (1)
  select name, creator, type, dbname, tsname, dbid, colcount
  from sysibm.systables
  where createdby = :createdby"
Do while rc = 0
  "RLX tbdisp1 tcreator panel(RLX)" (2)
end
Exit rc

```

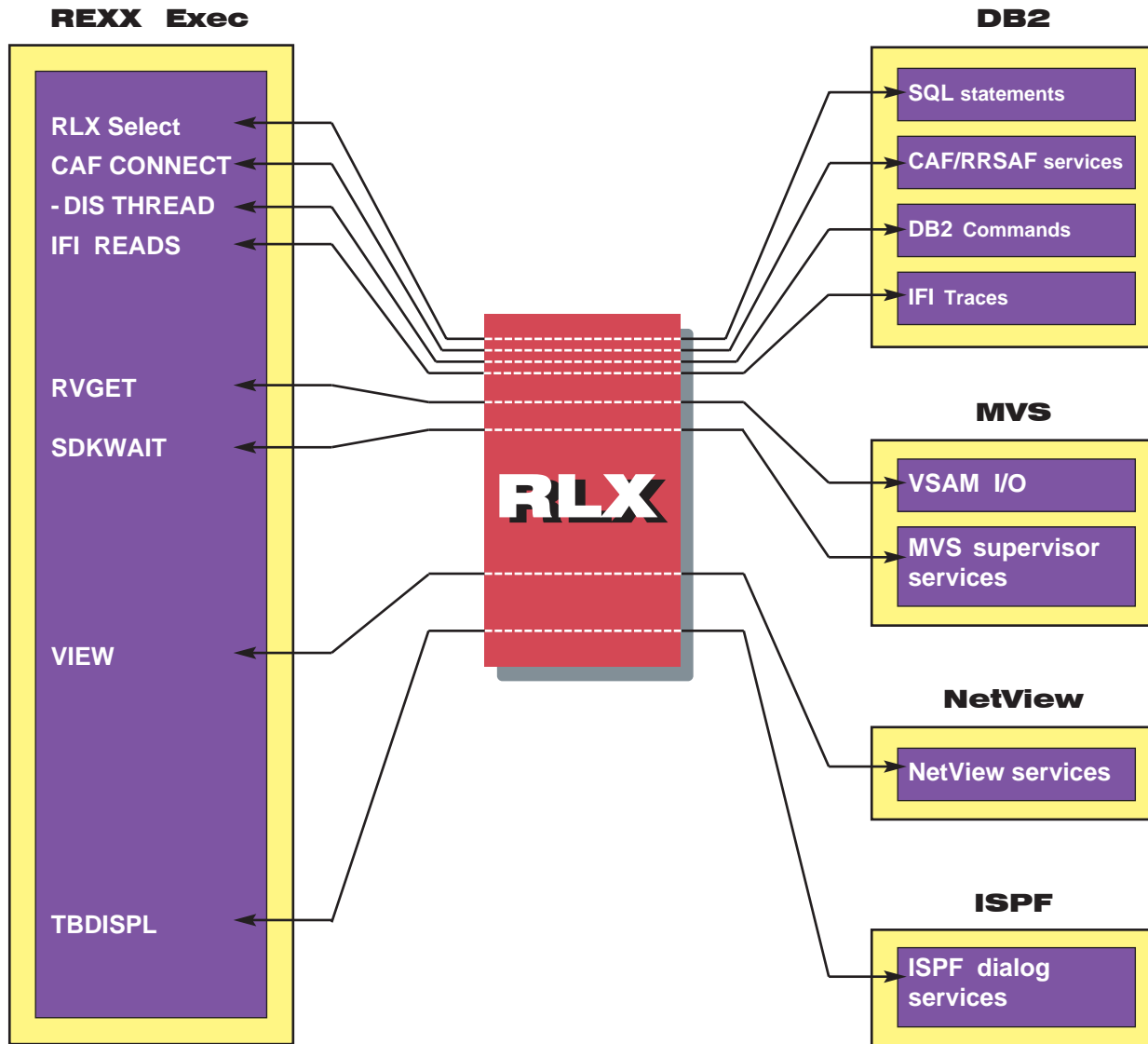
Figure 3: The expressive power and economy of RLX/ISPF. This compact RLX dialog produces the same scrollable display as the REXX exec in Figure 2.

REXX Examples

Figure 1 illustrates a scrollable display of DB2 tables created by the authid SYSIBM. The exec in Figure 2 interleaves standard SQL and ISPF dialog services to produce the display. Figure 3 illustrates a more compact solution using RLX/ISPF.

- (1) ADDRESS RLX routes host commands to RLX for execution. The RLX Host Command environment provides branch entry to both SQL and ISPF dialog services.
- (2) The colon prefix denotes SQL host variables. For example, the ARG instruction assigns a value to the REXX variable CREATEDBY while the DECLARE CURSOR statement is parameterized by the SQL host variable :CREATEDBY.
- (3) The RLX Host Command environment ADDRESS RLX recognizes all ISPF dialog service names so ADDRESS ISPEXEC need not be coded.
- (4) RLX FETCHes values directly through memory into 'host' variables shared by RLX, ISPF and REXX. Thus, the 'host' variables referenced in SQL statements are automatically ISPF dialog variables as well as variables your REXX exec can access. RLX returns SQL statement results in REXX variables and performs all data conversions automatically.
- (5) RLX returns feedback about the most recently executed SQL statement in the variables of the SQLCA.
- (6) RLX recognizes statements it has already processed to further improve performance. A reexecuted RLX SQL statement runs at 'static' SQL speed.

- (1) The RLX DECLARE ISPFTABLE service is patterned after the DECLARE CURSOR statement. It creates an ISPF table and loads it with the SQL query result.
- (2) The RLX TBDISPL service lets you display and process SQL query results on scrollable ISPF panels — without the need to design a scrollable panel beforehand or provide left/right scrolling logic.



Bottom Line:

Rapid Application Development

The RLX approach to application development provides the best of both worlds:

- Streamlined development with powerful debugging facilities and immediate execution.
- Later, deploy secure, high performance applications into production with RLX/Compile.

**Harness the power of
REXX, SQL and DB2**

**Call the REXX experts at
800 776-0771**