

# RDX

## REXX DB2 eXtensions for z/OS



- **Embed SQL/XML statements and XPath expressions natively within REXX execs**
- **Exercise SQL/XML as implemented by DB2 9 for z/OS as well as all SQL for DB2 V7, V8 and V9**
- **Publish (create) new XML data from existing data sources**
- **Exploit multi row FETCH support that outperforms static SQL within compiled applications**
- **Flow SQL/XML query results into REXX stemmed arrays in one operation**
- **Issue DB2 commands, IFI requests and RRSAP / CAF services – natively from REXX**
- **Learn pureXML with immediate execution and powerful debugging**

### Overview

**REXX DB2 eXtensions (RDX)** lets you code SQL and SQL/XML statements natively within REXX execs. RDX execs that combine SQL statements, XPath expressions, DB2 commands, IFI requests and RRSAP/CAF services can be developed rapidly and tested immediately. Without compile and link edit steps. Without preprocess and bind delays. You can edit and test RDX execs directly within ISPF Edit and get instant feedback.

What's more, there's a minimal learning curve. z/OS developers can leverage their skills with REXX, SQL and ISPF to become productive almost immediately with pureXML and Large Objects (LOBs). RDX harnesses the expressiveness and ease of use of REXX to get you started right away with the pureXML capabilities of DB2 9 for z/OS.

In addition to being a superb training tool, RDX is well suited to prototyping high volume systems and developing tools and small applications quickly – all with immediate feedback, extensive diagnostics and context sensitive help.

The combination of REXX and RDX facilitates learning and using pureXML to such a degree, every organization should consider RDX in their enablement plans for DB2 9 for z/OS.

### Benefits

- Leverage expertise in REXX, SQL and DB2
- Provide valuable "getting started" help with pureXML and DB2/z in terms of training, usage and debugging
- Provide DB2 for z/OS developers with the expressive power and instant feedback of REXX
- Streamline development with powerful debugging facilities and immediate execution
- Develop applications with *far less* coding than compiled and assembled languages
- Improve productivity and reduce the cost of training and development
- Exploit multi-row FETCH to develop REXX applications that run faster than their compiled, static SQL counterparts
- Explore and prototype new DB2 9 SQL with REXX, before programming as static SQL

## Features

- Supports all DB2 data types including XML, LOB, LOB\_FILE and LOB\_LOCATOR
- Converts automatically between DB2 and REXX datatypes
- Supports host variables within SQL and XPath expressions
- Provides granular access to XML document content
- Supports the same dynamic SQL syntax employed by DSNREXX and extends it with full support for SQL/XML, Large Objects and multi-row FETCH
- Provides REXX functions for UNICODE conversion as well as between ASCII and EBCDIC
- Augments the XML Guide for DB2 9 for z/OS with an extensive set of sample execs and RDX User Guide that parallel the DB2 XML Guide. They get you started as quickly as possible with pureXML and DB2 9 for z/OS.
- Interactive diagnostic facilities pinpoint errors and speed their correction
- A formatted SQLCA provides feedback after each SQL statement executes
- Formatted SQLDAs describe query result columns and the result sets returned by stored procedures

## Who Will Use REXX DB2 extensions?

Application Developers	✓
End users	✓
DBAs	✓
Operations	✓
Technical Support	✓
Systems Programmers	✓

## Prerequisites

RDX is easily installed and administered. It runs with all releases of z/OS and DB2 whose IBM support is current

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

## RDX Sample Exec

```

Address RDX (1)

"EXECSQL DECLARE C1 CURSOR FOR", (2)
"SELECT CID,",
"XMLQUERY(",
  "'declare default element namespace",
  "'http://posample.org';'",
  "/customerinfo/phone[@type = $type]'",
  'passing INFO, CAST(:phone_type AS VARCHAR(16)) as "type"',
  'AS "PHONE FROM INFO"',
"FROM MYCUSTOMER",
"WHERE CID between :low_value and :high_value"

If SQLCODE \= 0 then return SQLCODE (3)
"EXECSQL OPEN C2"
"EXECSQL FETCH C2 into :CID, :info"

Do while sqlcode = 0
  SAY cid info
  "EXECSQL FETCH C2 into :CID, :info" (4)
End

"EXECSQL CLOSE C2"
Return SQLCODE
  
```

Sample execs reference the XML Guide for DB2 for z/OS

- (1) ADDRESS RDX directs REXX to route host commands to RDX for execution.
- (2) The SQL/XML DECLARE statement (preceded by EXECSQL) is simply embedded within the REXX exec. The standard colon prefix denotes host variable references within SQL clauses and XPath expressions.
- (3) RDX updates all the host variables comprising the SQLCA to provide feedback about the most recently executed SQL/XML statement. For example, you can reference the current value of SQLCODE by name.
- (4) RDX FETCHes values directly thru memory into host variables shared by RDX, REXX and ISPF. Thus, the host variables referenced by SQL/XML statements are automatically REXX variables as well as dialog variables accessible to ISPF.

**Harness the power of  
DB2 9 for z/OS  
and its pureXML capabilities**

**Call the REXX/DB2 experts at  
800 776-0771**