

00101 01011000  
10000 01000101  
10010 01001001

IDUG® Europe

01000101 01011000 01010000 01000101 01010010 01001001 01000101 0100111  
01000100 01010101 01000111 00100001 00100000 01000101 01011000 0101000  
01001110 01000011 01000101 00100000 01001001 01000100 01000101 0101100

Experience IDUG

**Session: F14**

# DB2 and the Joy of Rexx

Ron Brown  
*Ronek Consulting*

Wednesday 7<sup>th</sup> October 2009  
17.00 – 18.00



**Platform: DB2 for z/OS and LUW**

# DB2 and the Joy of Rexx



- Agenda

- What is Rexx?
- Rexx – DB2 interfaces
- Advanced SQL invocation/debugging on z/OS
- SQL invocation on LUW
- Sample Rexx-DB2 code

# DB2 and the Joy of Rexx



- History of Rexx

- 1979 – 1982: creator Mike Cowlshaw on VM
- 1988: Ported to TSO & all other IBM platforms
- 1996: ANSI standard TRL-2 for “classic” Rexx
- 1997: Object Rexx for Windows & OS/2, then ported to AIX, Linux & SUN Solaris
- 2004: Open Object Rexx replaced Object Rexx

# DB2 and the Joy of Rexx



- Features of Rexx
  - English-like language with few format rules
  - Rich in Built-in Functions and Methods
  - Typeless variables and compound variables
  - Powerful string handling and decimal arithmetic
  - Clear error messages & powerful debugging
  - Interpreted or compiled
  - Portability

# DB2 and the Joy of Rexx



- DB2 Interfaces
  - Commands
  - IFI
  - Utilities
  - File access
  - Storage access
  - SQL
  - Stored Procedures

# DB2 and the Joy of Rexx



- **DB2 Commands**

- **On LUW, call the Administrative APIs or the DB2 Command Line Processor or ADMIN\_CMD procedure**

- Call SQLDBS 'CATALOG GLOBAL DATABASE  
/.../cell1/subsys/database/DB3 AS dbtest USING DIRECTORY DCE  
WITH "Sample Database"

- **On z/OS (TSO or batch) call DSN program**

- Queue command\_text
- Queue "END"
- Queue ""
- x = Outtrap(outline.)
- Address TSO "DSN SYSTEM("ssid")"
- x = Outtrap("OFF")
- /\* say "The number of lines trapped is" outline.0 \*/
- Do i = 1 to outline.0
- Say outline.i
- End



# DB2 and the Joy of Rexx



- DB2 Commands
  - **In z/OS Rexx Stored Procedure**
    - Queue "DSNE"
    - Queue command\_text
    - Queue "END"
    - Queue ""
    - x = Outtrap(outline.)
    - Address ATTCHMVS "DSNESM71"
    - x = Outtrap("OFF")
    - Do i = 1 to outline.0
      - Say outline.i
    - End
  - **On z/OS call IFI program DSNWLI2**

# DB2 and the Joy of Rexx



- **IFI**
- **Issuing DB2 commands**

- `IFCA = '00B4'X|| IFCA '||COPIES('00'X,24)' ' ||'0000'X||COPIES('`  
`' ,34)||'0000'X,`
- `||COPIES(' ',18)||'0000'X||COPIES(' ',82)`
- `CMD = "-STA TRACE(MON) CLASS(1)"`
- `COMMAND = SUBSTR("COMMAND",1,18)`
- `RTRNAREASIZE = 512`
- `RTRNAREA = D2C(RTRNAREASIZE+4,4)LEFT(' ',RTRNAREASIZE,' ')`
- `OUTPUT = D2C(LENGTH(CMD)+4,2)||'0000'X||CMD`
- `BUFFER = '00000000'X||'WBUF'||'0000000000000000'X`
- **ADDRESS LINKPGM "DSNWL12 COMMAND IFCA RTRNAREA OUTPUT**  
**BUFFER"**
- `RETCB = C2D(SUBSTR(IFCA,13,4))`
- `REASCB = D2X(C2D(SUBSTR(IFCA,17,4)))`



# DB2 and the Joy of Rexx



- **IFI**

- **Reading DB2 Trace records**

- `/* Issue READS for IFCID225 */`
- `TOTLEN = C2D(SUBSTR(IFCA,20+1,4))`
- `READS = SUBSTR('READS',1,8)`
- `RTRNAREA = '00001004'X||COPIES(' ',4096)`
- `IFCID = '0006'X||' '||'0081'X`
- `WQUAL = '00A8'X||'`  
`WQAL'||COPIES('00'X,88)'002B'X||COPIES('00',42)'CIR'||'0001'X||'N '`
- `||COPIES('00'X,8)' '||'0001'X||' '`
- `IFCIDAREA = '0006000000E1'X`
- **ADDRESS LINKPGM "DSNWLI2 READS IFCA RTRNAREA  
IFCIDAREA"**
- `RETCB = C2D(SUBSTR(IFCA,13,4))`
- `REASCD = D2X(C2D(SUBSTR(IFCA,17,4)))`

# DB2 and the Joy of Rexx



- DB2 Utilities
- On LUW
  - calling the Administrative APIs
    - Call SQLDBS “EXPORT :stmt TO datafile OF filetype MESSAGES msgfile
  - calling the ADMIN\_CMD stored procedure
    - Call SQLEXEC “CALL SYSPROC.ADMIN\_CMD(‘RUNSTATS ON TABLE RON.MYTABLE WITH DISTRIBUTION AND INDEXES ALL’)

# DB2 and the Joy of Rexx



- DB2 Utilities
- On z/OS
  - stand-alone utilities or DSNTIAUL can be directly invoked
  - DSNUTILB cannot be invoked from TSO (batch or online)
  - DSNUTILS Stored Procedure can be used to invoke utilities

# DB2 and the Joy of Rexx



- File Access
- Rexx can be used to read many different types of files which hold DB2 data. Some examples:
  - a) DB2 journal files
  - b) DB2MSTR message output
  - c) DBRMLIB library members
- Some sample output is on the next slide from a program in DB2 Update, November 2004

# DB2 and the Joy of Rexx



```
• CONTENTION REPORT - OBJECTS
• Object                DLocks TOuts
• =====
• DCS001P1.SFINPLC      23    133
• DCS001P1.SFINPLP      1     12
• DCS001P1.SFINFSS      0     2
•
• CONTENTION REPORT - TRANSACTIONS/JOBS
• Victim      DLocks      TOuts
• =====
• F417         4         25
• F420         2         12
• F421         3         8
•
• Holder      DLocks      TOuts
• =====
• E123         23        111
• F420         1         0
• N008         0         1
•
• Victim      Holder      DLocks      TOuts
• =====
• F417        E123         4         25
• F420        E123         2         11
• PFF10946    PFF10945      1         0
•
• Deadlocks
• =====
• 08.39.01 Plan PCSP1A Trans E123 in PA31CICS with Plan PCSP1A Trans F420
•           in PA41CICS on (0000210) DCS001P1.SFINPLC .00000001
• 09.51.32 Plan PCSP1A Trans E123 in PA32CICS with Plan PCSP1A Trans E123
•           in PA31CICS on Tablespace DCS001P1.SFINPLC page X'05E010A7
• 09.52.25 Plan PCSP1A Trans E134 in PA32CICS with Plan PCSP1A Trans E124
•           in PA41CICS on Tablespace DCS001P1.SFINCSH page X'000172
```

# DB2 and the Joy of Rexx



## • Storage Access

```
• Do i=1 to No-1 By 1
•   If SSCTSUE.i = '00000000' ,
•     | SSCTSUE.i = 'FFFFFFFF' Then Iterate i
•   ERLY_Ptr = SSCTSUE.i
•   ERLY_Eye_P = LA(ERLY_Ptr,4)
•   ERLY_Eye_P = LA(SSCTSUE.i,4)
•   ERLY_Eye   = MVC(ERLY_Eye_P,4)
•   If ERLY_Eye = 'ERLY' Then Do
•     ERLY_Module_P = LA(ERLY_Ptr,84)      /* Is this MQ or DB2? */
•     ERLY_Module   = MVC(ERLY_Module_P,8)
•     If Left(ERLY_Module,3) <> 'DSN' Then Iterate i /* CSQ is MQ! */
•     SSID#         = SSID# + 1
•     ERLY_SSID     = LA(ERLY_Ptr,8)      /* Extract DB2 SSID */
•     DB2SSID.SSID# = MVC(ERLY_SSID,4)
•     ERLY_EEPL    = LA(ERLY_Ptr,76)     /* Extract EEPL Addr */
•     DB2EEPL.SSID# = MVC(ERLY_EEPL,4)
•     ERLY_CHAR_P  = LA(ERLY_Ptr,112)    /* Extract DB2 CmdPref */
•     DB2CRC.SSID# = MVC(ERLY_CHAR_P,8)  /* .from DB2 V4.1 on */
•     ERLY_SCOM_P  = LA(ERLY_Ptr,56)    /* SCOM pointer set? */
•     ERLY_SCOM    = MVC(ERLY_SCOM_P,4)
•     If C2X(ERLY_SCOM) == '00000000' Then
•       DB2STAT.SSID# = 'Stopped'
•     Else DB2STAT.SSID# = 'Active'
```



# DB2 and the Joy of Rexx



- SQL Interfaces
  - DSNREXX
  - SQLEXEC
  - Self-written interfaces
  - Third-party Rexx-DB2 products

# DB2 and the Joy of Rexx



- **SQL with DSNREXX**
  - Only DYNAMIC SQL
    - a) EXECUTE IMMEDIATE
    - b) PREPARE then EXECUTE
    - c) PREPARE, DECLARE (cursor), OPEN (cursor), FETCH
    - d) UPDATE, INSERT or DELETE
    - e) CALL (stored procedure)
  - SQLCA automatically created by DSNREXX
  - SQLDA can be used
  - DESCRIPTOR and LOCATORS allowed
  - Singleton SELECT not allowed
  - Fixed names for cursors and statements

# DB2 and the Joy of Rexx



- Stored Procedures

- Rexx execs can call Stored Procedures

- PROCOUT = Left(' ',32000) /\* DSNWZP output \*/
    - Address DSNREXX "EXECSQL CALL SYSPROC.DSNWZP (:PROCOUT)"

- Rexx Stored Procedures on z/OS

- JCL Procedure must have NUMTCB=1
    - //SYSEXEC DD DSN=rexx\_library,DISP=SHR
    - Isolation depends on COLLID specified in CREATE PROCEDURE
    - CONNECT / DISCONNECT not required
    - Can use TSO commands

- Rexx Stored Procedures not allowed on AIX

# DB2 and the Joy of Rexx



- **Stored Procedures**

- Handling a result set

- `ADDRESS DSNREXX /* all EXECSQL statements to DSNREXX */`
- `"EXECSQL CALL :PROC"`
- `IF SQLCODE < 0 THEN SIGNAL SQL_ERROR`
- `"EXECSQL ASSOCIATE LOCATOR (:LOC1) WITH PROCEDURE :PROC"`
- `IF SQLCODE < 0 THEN SIGNAL SQL_ERROR`
- `"EXECSQL ALLOCATE C110 CURSOR FOR RESULT SET :LOC1"`
- `IF SQLCODE < 0 THEN SIGNAL SQL_ERROR`
- `DO UNTIL(SQLCODE <> 0)`
- `"EXECSQL FETCH C110 INTO :PARM1, :PARM2, :PARM3"`
- `IF SQLCODE < 0 THEN SIGNAL SQL_ERROR`
- `. . . .`
- `END`
- `. . . .`
- `"EXECSQL CLOSE C110"`
- `. . . . .`

# DB2 and the Joy of Rexx

- SQL invocation on z/OS
  - DSNREXX by another name?
    - Address TSO "SUBCOM DB2"
    - If rc Then
      - rc = RXSUBCOM('ADD','DB2','DSNREXX')
    - Address DB2 "EXECSQL" sql\_statement
  - DSNREXX verses SQLEXEC
    - Call SQLEXEC sql\_statement
      - a) performance is the same
      - b) SQLEXEC can be used unchanged on LUW
      - c) different connecting to DB2
      - d) SQLCA output structure is different

# DB2 and the Joy of Rexx



- Accessing remote DB2 objects
  - DSNREXX & SQLEXEC can use 3 part object names (eg. *“location.schema.table”*)
  - Sometimes a WAIT is required, like:
    - Address DSNREXX “CONNECT DB2M”
    - Address TSO “Call \*(WAITER) ‘500’ /\* wait 500 ms \*/”
    - sqltxt = “SELECT \* FROM” loc”.SYSIBM.LUNAMES”
    - Address DSNREXX “DECLARE C1 CURSOR FOR S1”
    - Address DSNREXX “PREPARE S1 INTO :SQLDA1 FROM :SQLTXT”
    - Address DSNREXX “OPEN C1”
    - Address DSNREXX “FETCH C1 INTO DESCRIPTOR :SQLDA1”



# DB2 and the Joy of Rexx



- SQL Errors

- Return Code

- Call `SQLEXEC sql_statement`

- Say `rc`

- +1 Warning (SQLCODE > 0)

- 0 OK (SQLCODE = 0)

- 1 Error (SQLCODE < 0)

- SQL Code / SQL State

- Address `DSNREXX "EXECSQL" sql_statement`

- Say ``SQLCODE =` sqlcode`

- Say ``SQLSTATE =` sqlstate`

# DB2 and the Joy of Rexx



- SQL debugging on z/OS

- **SQLCA variables returned by DSNREXX**

- ```
Say 'Sqlcode =' SQLCODE ' ',
```
- ```
      ' Sqlstate =' SQLSTATE '      Sqlerrp =' SQLERRP
```
- ```
Say 'Tokens =' Translate(SQLERRMC, ',', 'FF'x)
```
- ```
Say 'Sqlerrd =' 'SQLERRD.1', 'SQLERRD.2', 'SQLERRD.3', ', ',
```
- ```
      | 'SQLERRD.4', 'SQLERRD.5', 'SQLERRD.6
```
- ```
Say 'Sqlwarn =' 'SQLWARN.0', 'SQLWARN.1', 'SQLWARN.2', 'SQLWARN.3', ', ',
```
- ```
      | 'SQLWARN.4', 'SQLWARN.5', 'SQLWARN.6', 'SQLWARN.7', ', ',
```
- ```
      | 'SQLWARN.8', 'SQLWARN.9', 'SQLWARN.10
```

- **Sqlcode = -904 Sqlstate = 57011 Sqlerrp = DSNLVCLM**
- **Tokens = 00D31024,00001004,NABNETDB2D0001.P682861.DSNREXX**
- **Sqlerrd = 9,0,0,-1,0,0**
- **Sqlwarn = , , , , , , , , , ,**

# DB2 and the Joy of Rexx



- SQL debugging on z/OS

- Direct DSNTIAR invocation

```
/* Build normal SQLCA for DSNTIAR from Rexx variables */
SQLCA = 'SQLCA  '||D2C(136,4),          /* 136 = length of SQLCA */
      ||D2C(SQLCODE,4),
      ||D2C(70,2)||Left(SQLERRMC,70),   /* message tokens */
      ||Left(SQLERRP,8), /* (or set to 'DSN  ' to suppress) */
      ||D2C(SQLERRD.1,4)||D2C(SQLERRD.2,4)||D2C(SQLERRD.3,4),
      ||D2C(SQLERRD.4,4)||D2C(SQLERRD.5,4)||D2C(SQLERRD.6,4),
      ||Left(SQLWARN.0,1)||Left(SQLWARN.1,1)||Left(SQLWARN.2,1),
      ||Left(SQLWARN.3,1)||Left(SQLWARN.4,1)||Left(SQLWARN.5,1),
      ||Left(SQLWARN.6,1)||Left(SQLWARN.7,1)||Left(SQLWARN.8,1),
      ||Left(SQLWARN.9,1)||Left(SQLWARN.10,1)||Left(SQLSTATE,5)
msglen = 80          /* Message line length ( 72 to 240 allowed) */
msglen12 = msglen * 12          /* up to 12 lines of msgs */
sqlerrmsg = D2C(msglen12,2)||Copies(' ',msglen12)
sqlerrrlen = D2C(msglen,4)
/* run DSNTIAR passing SQLCA to it & getting message text returned */
Address LINKPGM "DSNTIAR SQLCA sqlerrmsg sqlerrrlen"
errmsg. = ''
Parse Var SQLERRMSG 2 errmsg.1 81 82 errmsg.2 161 162 errmsg.3 241,
      242 errmsg.4 321 322 errmsg.5 401 402 errmsg.6 481,
      482 errmsg.7 561 562 errmsg.8 641 642 errmsg.9 721
Do m = 1 to 9
  If errmsg.m <> '' Then Say errmsg.m
End
```

# DB2 and the Joy of Rexx



- SQL debugging on z/OS
  - The direct DSNTIAR invocation might result in error messages like:

```
DSNT408I  SQLCODE = -904, ERROR:  UNSUCCESSFUL EXECUTION CAUSED BY AN  
UNAVAILABLE RESOURCE. REASON 00D31024, TYPE OF RESOURCE 00001004,  
AND  
RESOURCE NAME NABNETDB2D0001.P682861.DSNREXX  
DSNT418I  SQLSTATE = 57011 SQLSTATE RETURN CODE  
DSNT415I  SQLERRP = DSNLVCLM SQL PROCEDURE DETECTING ERROR  
DSNT416I  SQLERRD = 9 0 0 -1 0 0 SQL DIAGNOSTIC INFORMATION  
DSNT416I  SQLERRD = X'00000009' X'00000000' X'00000000' X'FFFFFFFF'  
X'00000000' X'00000000' SQL DIAGNOSTIC INFORMATION
```

- compared to

```
Sqlcode = -904      Sqlstate = 57011      Sqlerrp = DSNLVCLM  
Tokens = 00D31024,00001004,NABNETDB2D0001.P682861.DSNREXX  
Sqlerrd = 9,0,0,-1,0,0  
Sqlwarn = , , , , , , , , , .
```

# DB2 and the Joy of Rexx



- SQL and Abends
  - Code an explicit ROLLBACK for abends if SQL is doing UPDATE, INSERT, CREATE etc.
  - To protect against errors:

```
• signal on syntax name error          /* turn on syntax trap          */
• signal on halt name error            /* trap this only on LUW        */
• . . . . .
• ERROR:
• rexx = sysvar(sysicmd)
• condition = condition('C')
• If condition = 'HALT' Then Call SQLDBS "INTERRUPT" /* only for LUW */
• source = strip(sourceline(sigl),"B")
• Call SQLEXEC "ROLLBACK" /* make sure we roll back */
• say copies('*',79)
• say left('* 'condition' CONDITION ON LINE 'sigl' OF REXX 'rexx,78)''
• say left('* 'source,78)''
• say left('* RETURN CODE 'rc,78)''
• say left('* 'errortext(rc),78)''
• say copies('*',79)
• exit 100
```

# DB2 and the Joy of Rexx



- RUNSQL
- Code for invoking SQL (with optional SQL tracing) and comprehensive diagnosis of any SQL errors
- Get it from the IDUG Code Place
  - `rc = RUNSQL("sql_statement")`



# DB2 and the Joy of Rexx

```
File Edit Ed_Settings Menu Utility Compile Test Help P682861 on MSYS
EDIT          P682861.ISPEXEC(TESTSQL6) - 01.10          TGEN20  cols 00001 00072
Command ==> exe          Scroll ==> CSR
000008  runsql_trace = 'NO'
000009
000010  x = RUNSQL("CONNECT DB2M")
000011
000012  locn = 'NABNETDB2T0001.';  tabl = 'LUNAMES'
000013  sqlstmt = "SELECT *"
000014             " FROM" locn"SYSIBM."tabl ;
000015             " WHERE LUNAME LIKE ?" ;
000016             " AND SECURITY_IN <= ?" ;
000017             " AND SECURITY_OUT = ?" ;
000018             " WITH UR"
000019
000020  x = RUNSQL("PREPARE S2 INTO :S2SQLDA FROM :SQLSTMT")
000021
000022  x = RUNSQL("DECLARE C2 CURSOR FOR S2")
000023
000024  lu = 'T3%';  si = 'Z';  so = 'A'
000025  x = RUNSQL("OPEN C2 USING :LU, :SI, :SO")
000026
000027  x = RUNSQL("FETCH C2 INTO DESCRIPTOR :S2SQLDA")
000028
000029  Do i = 2 to 200 Until sqlcode <> 0
000030      x = RUNSQL("FETCH C2 INTO :LUNAME.i, :SYSM, :SEC_IN, :SEC_OUT.i," ,
000031              " VAR5, :VAR6, :VAR7, :VAR8, :VAR9")
000032  End
000033
000034  x = RUNSQL("CLOSE C2")
000035
000036  x = RUNSQL("DISCONNECT")
000037
000038  Say 'S2SQLDA.1.SQLDATA = ' S2SQLDA.1.SQLDATA ,
000039             COLUMN = ' S2SQLDA.1.SQLNAME ,
000040             TYPE = ' S2SQLDA.1.SQLTYPE ,
000041             LENGTH = ' S2SQLDA.1.SQLEN
000042  Do j = 2 To i-1
000043      Say ' LUNAME.'j '=' LUNAME.j
000044  End
000045  Say ' '
000046  Return
```

# DB2 and the Joy of Rexx



```
===== SQL Error =====
P682861.ISPEXEC(TESTSQL6) Line 30                10 Sep 2009  17:53:12

Running on system MSYS          Connected to DB2M

FETCH C2 INTO :LUNAME.i, :SYSM, :SEC_IN, :SEC_OUT.i,
              VAR5, :VAR6, :VAR7, :VAR8, :VAR9

DSNT408I  SQLCODE = -104, ERROR:  ILLEGAL SYMBOL "VAR5".  SOME SYMBOLS THAT MIGHT
          BE LEGAL ARE: <HOST-VARIABLE>
DSNT418I  SQLSTATE   = 42601 SQLSTATE RETURN CODE
DSNT415I  SQLERRP    = DSNTZNT0 SQL PROCEDURE DETECTING ERROR
DSNT416I  SQLERRD    = 0 0 0 -1 0 0 SQL DIAGNOSTIC INFORMATION
DSNT416I  SQLERRD    = X'00000000' X'00000000' X'00000000' X'FFFFFFFF'
          X'00000000' X'00000000' SQL DIAGNOSTIC INFORMATION

Failing SQL statement text
-----
SELECT *
  FROM NABNETDB2T0001.SYSIBM.LUNAMES
 WHERE LUNAME LIKE ?
       AND SECURITY_IN <= ?
       AND SECURITY_OUT = ?
 WITH UR

REXX host variable values
-----
          LU = 'T3%'
          SI = 'Z'
          SO = 'A'

=====

***
```

# DB2 and the Joy of Rexx

```
----- Start Trace -----
P682861.ISPEXEC(TESTSQL6) 11 Sep 2009 16:17:37

CONNECT DB2M
Line 10  Sqlcode = 0  Elapsed = 0.234707 sec

PREPARE S2 INTO :S2SQLDA FROM :SQLSTMT
Line 20  Sqlcode = 0  Elapsed = 0.597638 sec
-----
SELECT *
  FROM NABNETDB2T0001.SYSIBM.LUNAMES
  WHERE LUNAME LIKE ?
        AND SECURITY_IN <= ?
        AND SECURITY_OUT = ?
  WITH UR
-----
SQLDA data in S2SQLDA:
VARIABLE          LUNAME          = ' '
VARIABLE          SYSMODENAME     = ' '
CHARACTER         SECURITY_IN     = ' '
CHARACTER         SECURITY_OUT    = ' '
CHARACTER         ENCRYPTPSWDS    = ' '
CHARACTER         MODESELECT     = ' '
CHARACTER         USERNAMES      = ' '
CHARACTER         GENERIC        = ' '
CHARACTER         IBMREQD       = ' '

DECLARE C2 CURSOR FOR S2
Line 22  Sqlcode = 0  Elapsed = 0.000120 sec

OPEN C2 USING :LU, :SI, :SO
Line 25  Sqlcode = 0  Elapsed = 0.000610 sec
-----
                LU = 'T3%'
                SI = 'Z'
                SO = 'A'

FETCH C2 INTO DESCRIPTOR :S2SQLDA
Line 27  Sqlcode = 0  Fetch 1  Elapsed = 0.028270 sec
-----
SQLDA data in S2SQLDA:
***
```

# DB2 and the Joy of Rexxx

```

VARCHA(24)          LUNAME      = 'T3130100'
VARCHA(24)          SYSMODENAME = '
CHAR(1)             SECURITY_IN  = 'V'
CHAR(1)             SECURITY_OUT = 'A'
CHAR(1)             ENCRYPTPSWDS  = 'N'
CHAR(1)             MODESELECT   = 'N'
CHAR(1)             USERNAMES    = '
CHAR(1)             GENERIC      = 'N'
CHAR(1)             IBMREQD      = 'N'

FETCH C2 INTO :LUNAME.i, :SYSM, :SEC_IN, :SEC_OUT.i,
              :VAR5, :VAR6, :VAR7, :VAR8, :VAR9
  Line 30      sqlcode = 0      Fetch 2      Elapsed = 0.005282 sec
-----
      LUNAME.2 = 'T3130201'
      SYSM     = '
      SEC_IN   = 'A'
      SEC_OUT.2 = 'A'
      VAR5    = 'N'
      VAR6    = 'N'
      VAR7    = '
      VAR8    = 'N'
      VAR9    = 'N'

FETCH C2 INTO :LUNAME.i, :SYSM, :SEC_IN, :SEC_OUT.i,
              :VAR5, :VAR6, :VAR7, :VAR8, :VAR9
  Line 30      sqlcode = 0      Fetch 3      Elapsed = 0.005413 sec
-----
      LUNAME.3 = 'T3169200'
      SYSM     = '
      SEC_IN   = 'V'
      SEC_OUT.3 = 'A'
      VAR5    = 'N'
      VAR6    = 'N'
      VAR7    = '
      VAR8    = 'N'
      VAR9    = 'N'

FETCH C2 INTO :LUNAME.i, :SYSM, :SEC_IN, :SEC_OUT.i,
              :VAR5, :VAR6, :VAR7, :VAR8, :VAR9
  Line 30      sqlcode = +100    sqlstate = 02000    Elapsed = 0.004686 sec
***

```

# DB2 and the Joy of Rexx

```
CLOSE C2
  Line 34  Sqlcode = 0  Elapsed = 0.008760 sec
-----
  There were 3 successful fetches from this cursor

DISCONNECT
  Line 36  Sqlcode = 0  Elapsed = 0.011280 sec

S2SQLDA.1.SQDATA = T3130100  COLUMN = LUNAME  TYPE = 448  LENGTH = 24
                LUNAME.2 = T3130201
                LUNAME.3 = T3169200

***
```



# DB2 and the Joy of Rexx



- Invocation on LUW
  - **AIX: Initialise environments before connecting**
    - Rc = SysAddFuncPkg('db2rex')
      - If rc <> 0 Then Return "Unable to initialise DB2REXX"
  - **Windows: Initialise environments before connecting**
    - If RxFuncquery('SQLDBS') Then
      - If RxFuncAdd('SQLDBS','DB2AR','SQLDBS')
      - Then Return "Unable to register SQLDBS"
    - If RxFuncquery('SQLDB2') Then /\* CLP \*/
      - If RxFuncAdd('SQLDB2','DB2AR','SQLDB2')
      - Then Return "Unable to register SQLDB2"
    - If RxFuncquery('SQLEXEC') Then /\* SQL \*/
      - If RxFuncAdd('SQLEXEC','DB2AR','SQLEXEC')
      - Then Return "Unable to register SQLEXEC"



# DB2 and the Joy of Rexx



- SQL invocation on LUW
  - Connect to DB2
    - Call SQLDBS "ATTACH TO" db2name
    - Call SQLEXEC "CONNECT" db2name
  - SQLEXEC must be used
    - Call SQLEXEC sql\_statement
  - SQLCA is written to stem variables called SQLCA.xxxxx (where 'xxxxx' are the same names as the variables created by DSNREXX)
    - Say `SQLCODE =` sqlca.sqlcode
    - Say `SQLSTATE =` sqlca.sqlstate
  - Disconnect from DB2
    - Call SQLDBS "DETACH"
    - Call SQLEXEC "CONNECT RESET"

# DB2 and the Joy of Rexx



- **Debugging on LUW**

- Return Code, SQL Code & SQL State are the same as on z/OS, and SQLCA is updated after every SQL statement or API call.
- Error messages can be produced by either of the following APIs, which use the contents of the SQLCA structure to obtain information.

- The REXX API syntax for Get Error Message is:

```
Call SQLDBS "GET MESSAGE INTO :msg [LINEWIDTH width]"
```

The REXX API syntax for Get the SQLSTATE Message is:

```
Call SQLDBS
```

- "GET MESSAGE FOR SQLSTATE sqlstate INTO :msg . . . "

# DB2 and the Joy of Rexx



- **Setting the Isolation Level**
- **DSNREXX and SQLEXEC default to CS**
- **z/OS has 4 packages**
  - DSNREXCS - Cursor Stability
  - DSNREXRR - Repeatable Read
  - DSNREXRS - Read Stability
  - DSNREXUR - Uncommitted Read
- **Call EXECSQL “SET CURRENT PACKAGESET=‘DSNREXUR’ “**
- **LUW has 5 packages**
  - DB2ARXCS.BND - Cursor Stability
  - DB2ARXRR.BND - Repeatable Read
  - DB2ARXRS.BND - Read Stability
  - DB2ARXUR.BND - Uncommitted Read
  - DB2ARXNC.BND - No Commit (on some AS/400 systems)
- **Call SQLDBS “CHANGE SQLISL TO UR“**

# DB2 and the Joy of Rexx



- Unicode Considerations
  - This is usually only an issue on DB2 for z/OS, starting with version 8.1
  - SELECT from DB2 Catalog (or other tables with CCSID UNICODE)
  - Interpret DBRMLIB members

# DB2 and the Joy of Rexx



```
Menu Utilities Compilers Help P682861 on MSYS
BROWSE SYS1.QMF.DSQDBRM(DSQESDB8) MDB2A1 0000000 col 001 080
Command ==> Scroll ==> CSR
***** Top of Data *****
DBRM...µw98COMP DSQESDB8.(...152..B.....4..1..ØLL
..UK15152
DBRM...µ.....Aäää< éá.ääéäää.äíêê|é.ä|é.éá<áää.ä|<+|...ä|<èß&á...+í<<
ë...äää í<è...äää í<èí <íá...éä <á...<á+ääç.äê|ç.éßèñäç...éßèä|<íç+è.íçáéá.éääéá
è|é.....ç. +à.éä+ (á.....ç. +à.ä|<+|...ä|é.ääéäç. |+<ß.....°Ø.....Ø.{...
...èä|í+áé.....~Ø.....éä+ (á
DBRM...u.....içá+áíáé.éé<áéê|é.ä|+èñ+íá.
DBRM...v.....içá+áíáé.éé<í è+ñ+á.ä|+èñ+íá.
DBRM...w.....içá+áíáé.+|é.ä|í+à.ä|+èñ+íá.
DBRM...é.....[.....éáè.äíêéá+è.éé<ñä.....ç.....Ø.....Ø.{.....äíêéé<
ñä
DBRM...é.....T.....éáè.äíêéá+è.äääéáá.....ç.....Ø.....D.....äíêéäá
ä
DBRM...ê.....éáè.äíêéá+è.éí<áé.....ç.....Ø.....D.....äíêéí<
áé
DBRM...í.....éáè.äíêéá+è.&éáñèñ|+.....ç.....Ø.....D.....äí
é&éáä
DBRM...í.....éáè.äíêéá+è.|&èñ(ñ! èñ|+çñ+è.....ç.....Ø.....
D.....äíê|&èç
DBRM...*.....F.....éáè.äíêéá+è.<|ä <á.<ä-äèß&á.....ç.....Ø.....D.
...äíê<|ä <
DBRM...ç.....O.....éáè.äíêéá+è. &&<ñä èñ|+á+ä|àñ+ä.éäçá(á.....ç.....
Ø.....D.....äíéá+ä|ä
DBRM...|.....I.....áíááíéá.ñç(áàñ éá.....ç.....Ø.....{.....éáèèèç(è
DBRM...U.....=.....èéá<áää.èä+ (á.....éääéá è|é...ñ+è|...ç.....ç.äê|ç.éßèñäç
...éßèèß+|+ß(è..içáéá.+ (á.....ç. +à..äéá è|é.....ç.....Ø.....{...
..éé+ (á.....Ø.....{.....éê|í+áé.....ßØ.....Ø.....{.....ñ++ (á....._Ø.....Ø
..{.....ñ+|í+áé
DBRM...A.....|&á+.ääéäää.....Ø.....Ø.....Ø.....èä|í+áé.....Ø...
...Ø.....éä+ (á
DBRM...ø.....ääéäç.ääéäää..ñ+è|...ç.....ç.....ç.....ç.....ç.....ç.....
.ç.....4.....ä|<+|.....D.....ä|<èß&á.....D
...+í<<è.....D.....äää í<è.....Ù.....äääí <íá.....
4.....éä <á.....4.....ä|<<á+
DBRM...6.....ä<|éá.ääéäää
DBRM...5.....Oéá<áää.èß&á...<|ä èñ|+...éääéá è|é...éä+ (á.ñ+è|...ç.....
.ç.....ç.....ç.äê|ç.éßèñäç...éßèè ä<áé.içáéá.+ (á.....ç. +à.äéá è|é.....ç.í+ñ|+.
```

# DB2 and the Joy of Rexx



- DBRMMAP

```
Menu  Functions  Confirm  Utilities  Help                                P682861 on MSYS
VIEW                                     SYS1.QMF.DSQDBRM                                MDB2A1  Row 00030 of 00046
Command ==>                               Scroll ==> CSR
Name      Prompt      Size  Created      Changed      ID
dbrmmap__ DSQESDB8  *Browsed
_____  DSQESDGN
_____  DSQESDTA
_____  DSQESDT7
_____  DSQESDT8
_____  DSQESQ44
_____  DSQESQ45
_____  DSQESSDS
_____  DSQESSQL
_____  DSQESSQ2
_____  DSQESSQ5
_____  DSQESUSR
_____  DSQESV
_____  DSQESV2
_____  DSQETMGT
_____  DSQETSQL
_____  DSQEUPRF
**End**
```



# DB2 and the Joy of Rexxx



```
Menu Utilities Compilers Help P682861 on MSYS
BROWSE P682861.DBRMAP.T76967 TGEN08 00000000 Col 001 080
Command ==> Scroll ==> CSR
***** Top of Data *****
SOURCE DBRM MEMBER: SYS1.QMF.DSQDBRM(DSQESDB8)

Precompile Userid = W98COMP
Program Name = DSQESDB8
DBRM Contoken = 0E4D2F1F05F1F5F2
Loadlib Token = 05F1F5F20E4D2F1F
DB2 Version = V8
Program Version Id = UK15152
Precompile Options:
  DEC(15)
  SQLFLAG(STD) (a.k.a SQLFLAG(86))
  DATE(JIS) or DATE(ISO)
  TIME(EUR) or TIME(ISO)
  NOGRAPHIC
  APOSTSQL
  APOST
  SQL(DB2)

Section Number = 1
Statement number = 6304
Statement Text = DECLARE GETCDF4 CURSOR FOR SELECT COLNO , COLTYPE , NULLS
                DEFAULTVALUE , SCALE , LENGTH FROM SYSIBM . SYSCOLUMNS WH
                = : H AND TBNAME = : H AND COLNO > 0 FOR FETCH ONLY

  HOST VAR#  HOST VARIABLE NAME  IN/OUT  HOST VARIAB
  -----  -
  1          TBOWNER            INPUT   VARCHAR(128)
  2          TBNAME             INPUT   VARCHAR(128)

Section Number = 0
Statement number = 1956
Statement Text = WHENEVER SQLERROR CONTINUE

Section Number = 0
```

# DB2 and the Joy of Rexx

## • Unicode Considerations

```
• tablei = XRANGE('00'x,'FF'x)
• tableo = /* conversion table for Unicode UTF-8 to EBCDIC */,
• '00'x'01'x'02'x'03'x'37'x'2D'x'2E'x'2F'x'16'x'05'x'25'x'0B'x |,
• '0C'x'0D'x'0E'x'0F'x'10'x'11'x'12'x'13'x'3C'x'3D'x'32'x'26'x |,
• '18'x'19'x'3F'x'27'x'1C'x'1D'x'1E'x'1F'x'40'x'4F'x'7F'x'7B'x |,
• '5B'x'6C'x'50'x'7D'x'4D'x'5D'x'5C'x'4E'x'6B'x'60'x'4B'x'61'x |,
• 'F0'x'F1'x'F2'x'F3'x'F4'x'F5'x'F6'x'F7'x'F8'x'F9'x'7A'x'5E'x |,
• '4C'x'7E'x'6E'x'6F'x'7C'x'C1'x'C2'x'C3'x'C4'x'C5'x'C6'x'C7'x |,
• 'C8'x'C9'x'D1'x'D2'x'D3'x'D4'x'D5'x'D6'x'D7'x'D8'x'D9'x'E2'x |,
• 'E3'x'E4'x'E5'x'E6'x'E7'x'E8'x'E9'x'4A'x'E0'x'5A'x'5F'x'6D'x |,
• '79'x'81'x'82'x'83'x'84'x'85'x'86'x'87'x'88'x'89'x'91'x'92'x |,
• '93'x'94'x'95'x'96'x'97'x'98'x'99'x'A2'x'A3'x'A4'x'A5'x'A6'x |,
• 'A7'x'A8'x'A9'x'C0'x'BB'x'D0'x'A1'x'07'x'20'x'21'x'22'x'23'x |,
• '24'x'15'x'06'x'17'x'28'x'29'x'2A'x'2B'x'2C'x'09'x'0A'x'1B'x |,
• '30'x'31'x'1A'x'33'x'34'x'35'x'36'x'08'x'38'x'39'x'3A'x'3B'x |,
• '04'x'14'x'3E'x'FF'x'41'x'AA'x'B0'x'B1'x'9F'x'B2'x'6A'x'B5'x |,
• 'BD'x'B4'x'9A'x'8A'x'BA'x'CA'x'AF'x'BC'x'90'x'8F'x'EA'x'FA'x |,
• 'BE'x'A0'x'B6'x'B3'x'9D'x'DA'x'9B'x'8B'x'B7'x'B8'x'B9'x'AB'x |,
• '64'x'65'x'62'x'66'x'63'x'67'x'9E'x'68'x'74'x'71'x'72'x'73'x |,
• '78'x'75'x'76'x'77'x'AC'x'69'x'ED'x'EE'x'EB'x'EF'x'EC'x'BF'x |,
• '80'x'FD'x'FE'x'FB'x'FC'x'AD'x'AE'x'59'x'44'x'45'x'42'x'46'x |,
• '43'x'47'x'9C'x'48'x'54'x'51'x'52'x'53'x'58'x'55'x'56'x'57'x |,
• '8C'x'49'x'CD'x'CE'x'CB'x'CF'x'CC'x'E1'x'70'x'DD'x'DE'x'DB'x |,
• 'DC'x'8D'x'8E'x'DF'x |,
• text = TRANSLATE(text,tableo,tablei) /* Unicode -> EBCDIC */
```

# DB2 and the Joy of Rexx



- **SPUFI**

- SPUFI is designed to run under the DSN program, but only online in TSO/ISPF
- Rexx can use its ISPEXEC interface to seed some ISPF variables, then invoke SPUFI with a modified SPUFI panel to run SQL immediately

```
• Address TSO "NEWSTACK" /* start a new (empty) stack */
• Parse Value '; 2500 4092 4096 VB SYSDA',
• '33 256 NAMES C SECOND TIME',
• With DSNESV2B DSNESV2D DSNESV2C DSNESV21 DSNESV22 DSNESV2E,
• DSNESV24 DSNESV25 DSNESV26 DSNESV3Z DSNESV1W
• "VPUT (DSNESV2B DSNESV2D DSNESV2C DSNESV21 DSNESV22 DSNESV2E",
• "DSNESV24 DSNESV25 DSNESV26 DSNESV3Z DSNESV1W) PROFILE"
• Push ''
• Push 'END'
• Push 'SPUFI'
• "SELECT CMD(DSN SYSTEM("dsneov01") TEST(0) RETRY(0))"
• Address TSO "DELSTACK" /* finished with that stack */
```

# DB2 and the Joy of Rexx



- Unsupported SQL
  - Scrollable Cursors
  - Multi-row Fetch
  - Multi-row Insert
  - GET DIAGNOSTICS
- They won't ever be supported unless enough users request them!

# DB2 and the Joy of Rexx

- Disadvantages of Rexx
  - performance
  - no static SQL
  - some SQL not supported
  - possible “write-only code”
  - not widely used on LUW

# DB2 and the Joy of Rexx



- Why should you use Rexx?
  - Quick development
  - Can use many types of interfaces
  - Runs on many different platforms
  - Widely used by DBAs and DB2 Systems Programmers on z/OS



# Session F14

## DB2 and the Joy of Rexx



**Ron Brown**  
Ronek Consulting  
Ronek.Consulting@gmail.com