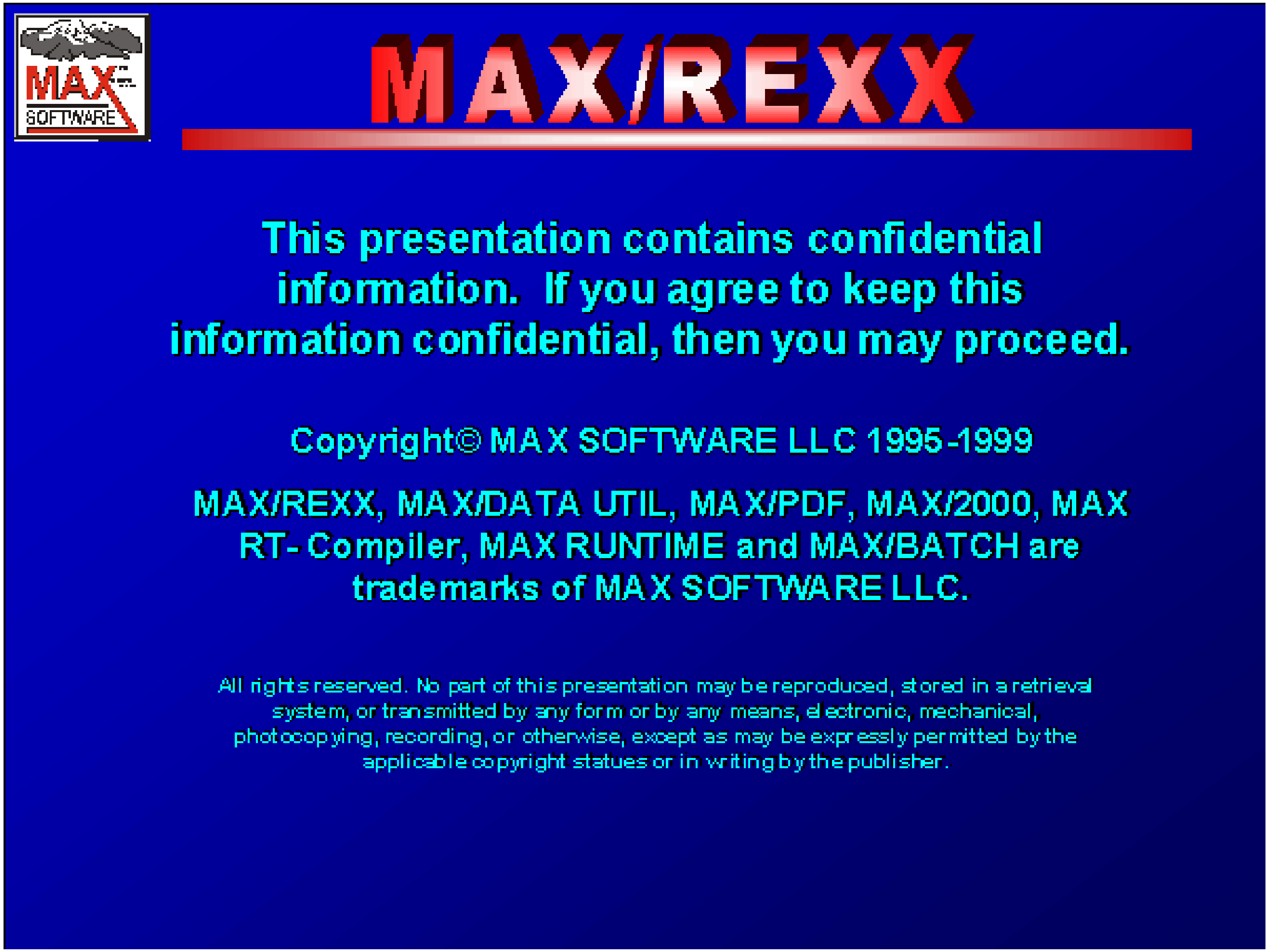


MAX/REXX

This presentation contains confidential information. If you agree to keep this information confidential, then you may proceed.

Welcome to MAX/REXX

Currently available on OS/390 where a large percentage of the legacy business data resides

MAX/REXX

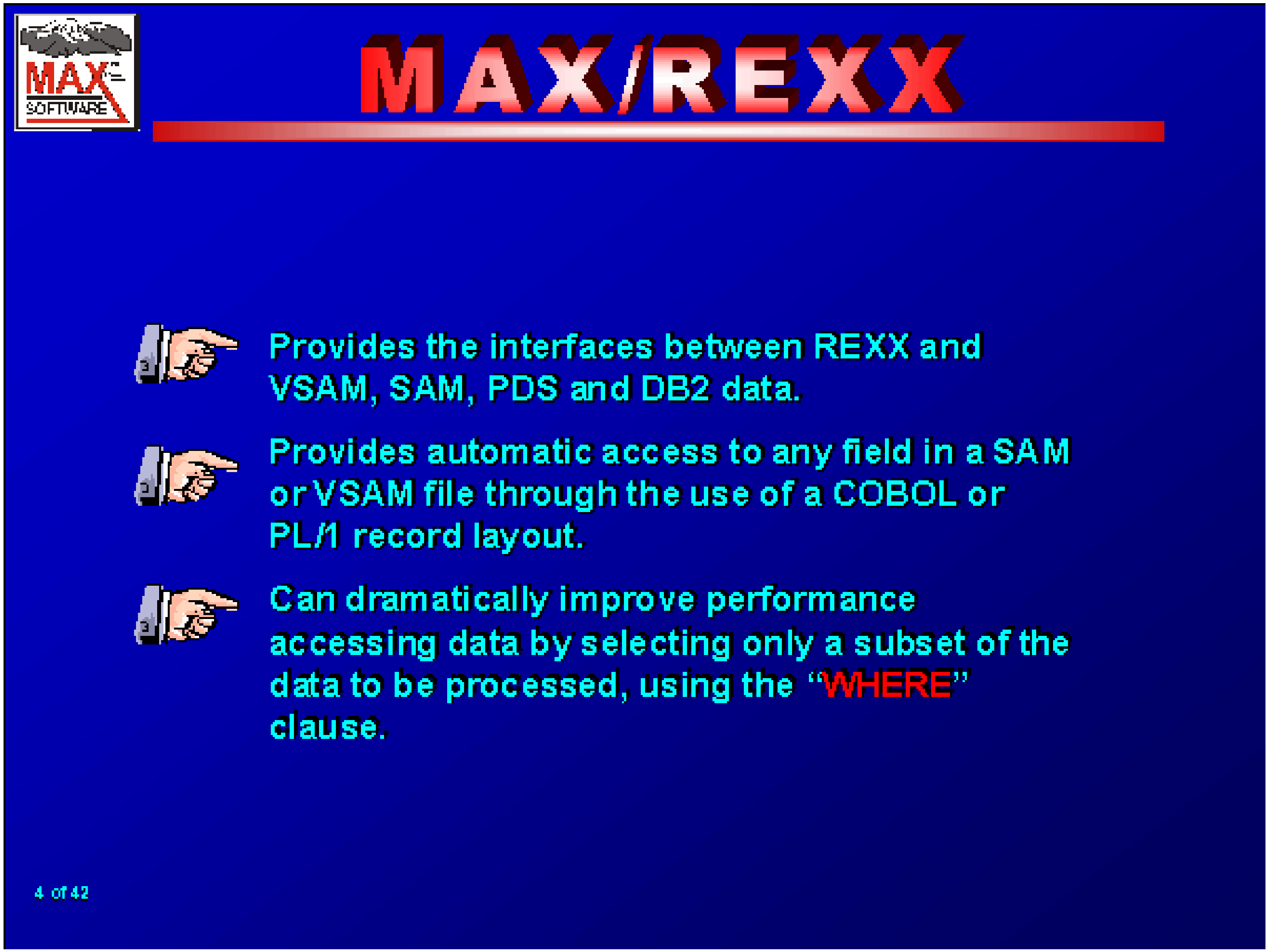
- Allows the REXX programming language to be used to solve business problems.
- Provides the capabilities to process business data with the performance needed in a comprehensive language environment.

MAX/REXX

The competitiveness of an organization can be directly linked to its ability to respond quickly to changes and opportunities.

Responsiveness in an information driven business requires a standardized set of tools that can be used to rapidly develop or revise both applications and data.

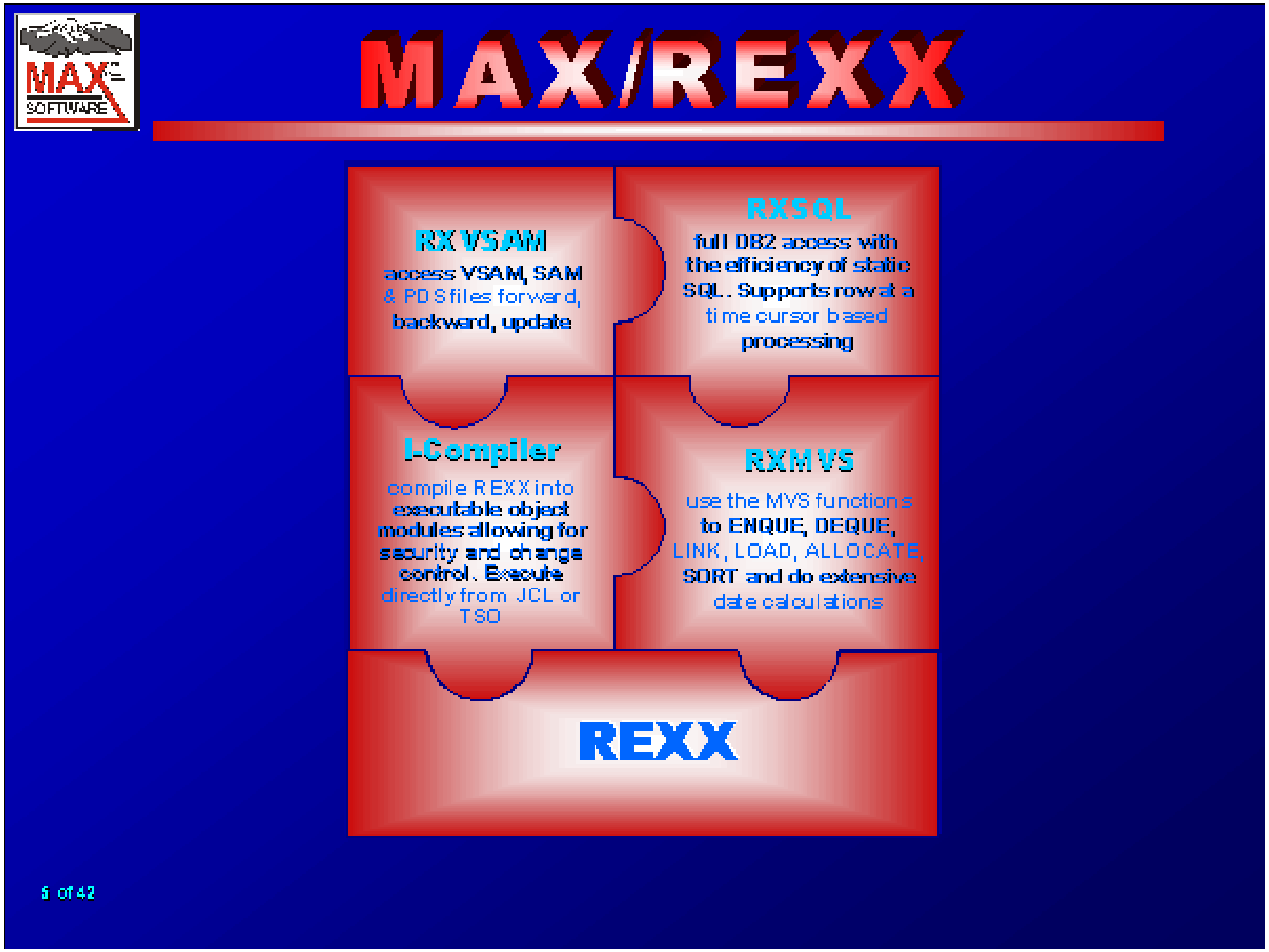
REXX is the language that allows for quick response. However, classic REXX cannot manipulate all the data types and file formats that typify today's applications.



MAX/REXX

- Provides the interfaces between REXX and VSAM, SAM, PDS and DB2 data.
- Provides automatic access to any field in a SAM or VSAM file through the use of a COBOL or PL/1 record layout.
- Can dramatically improve performance accessing data by selecting only a subset of the data to be processed, using the "WHERE" clause.

4 of 42



MAX/REXX

RX VSAM

access VSAM, SAM & PDS files forward, backward, update

RXSQL

full DB2 access with the efficiency of static SQL. Supports row at a time cursor based processing

I-Compiler

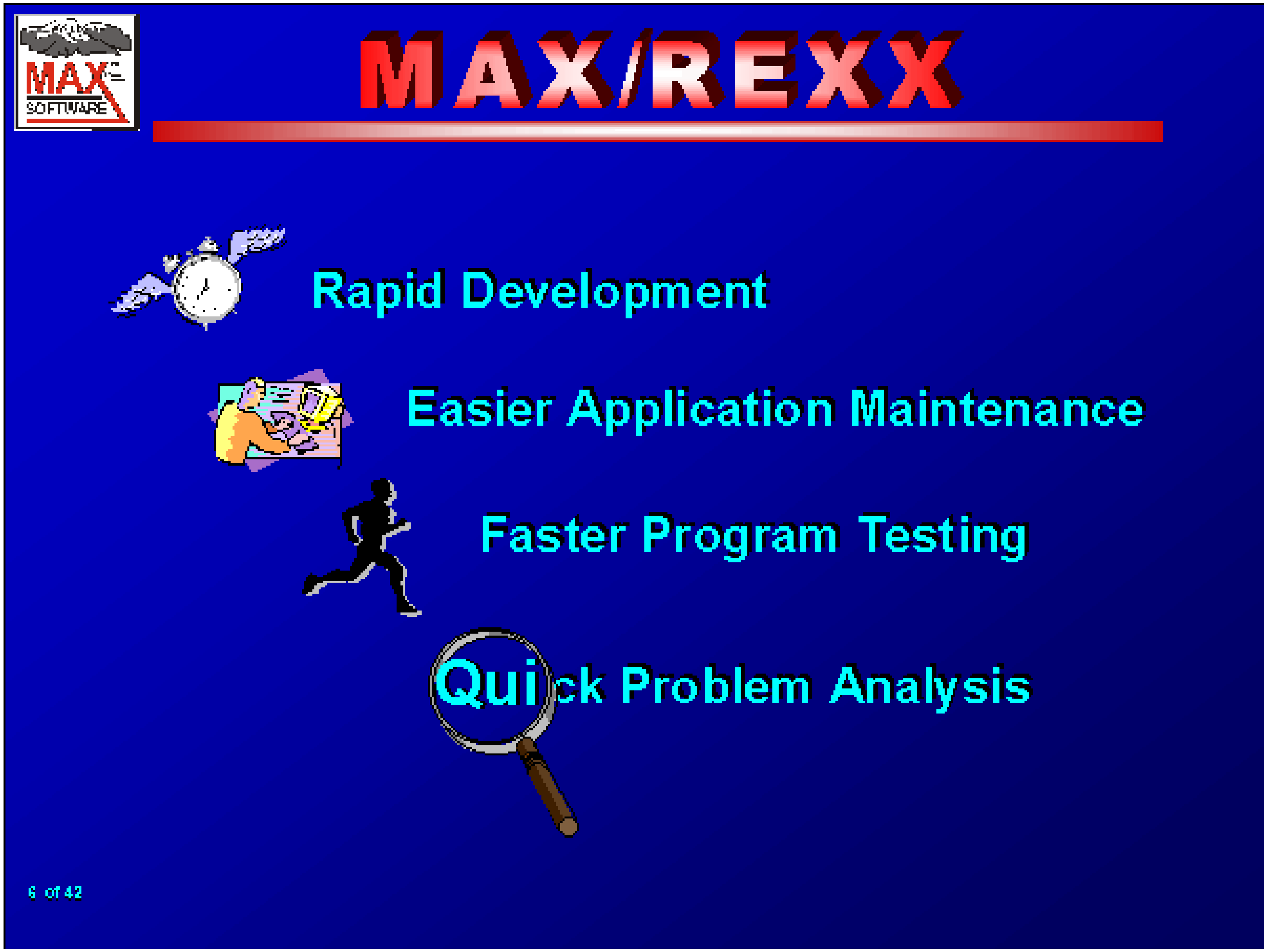
compile REXX into executable object modules allowing for security and change control. Execute directly from JCL or TSO

RXMVS

use the MVS functions to ENQUE, DEQUE, LINK, LOAD, ALLOCATE, SORT and do extensive date calculations

REXX

5 of 42



MAX/REXX

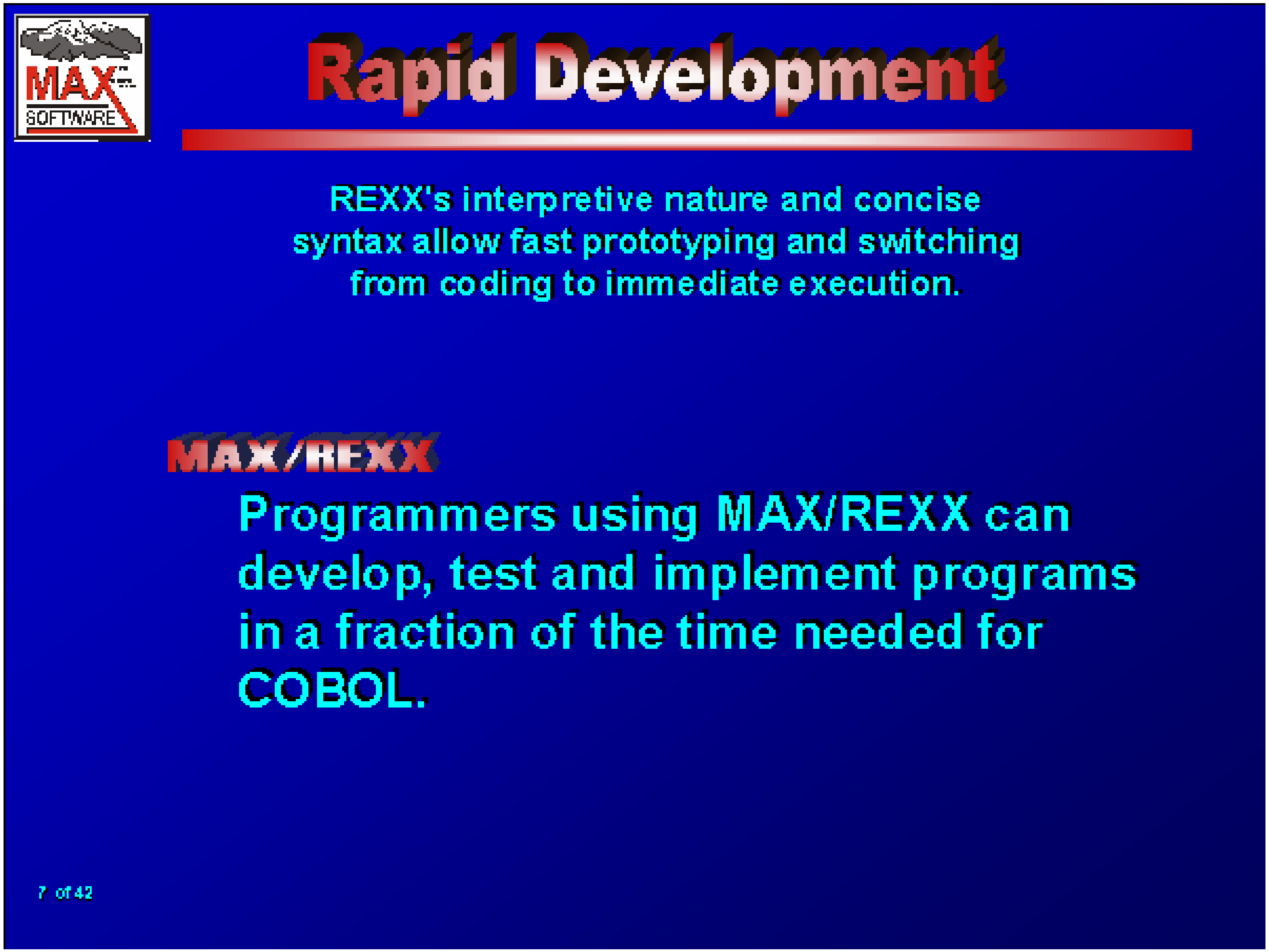
Rapid Development

Easier Application Maintenance

Faster Program Testing

Quick Problem Analysis

6 of 42



Rapid Development

REXX's interpretive nature and concise syntax allow fast prototyping and switching from coding to immediate execution.

MAX/REXX

Programmers using MAX/REXX can develop, test and implement programs in a fraction of the time needed for COBOL.

Easier Application Maintenace

- MAX/REXX programs have a structured, top-down syntax, and programs are smaller and easier to read.
- MAX/REXX uses an externally defined file layout. Changes to file format do not necessarily require modification or recompilation of the associated programs.

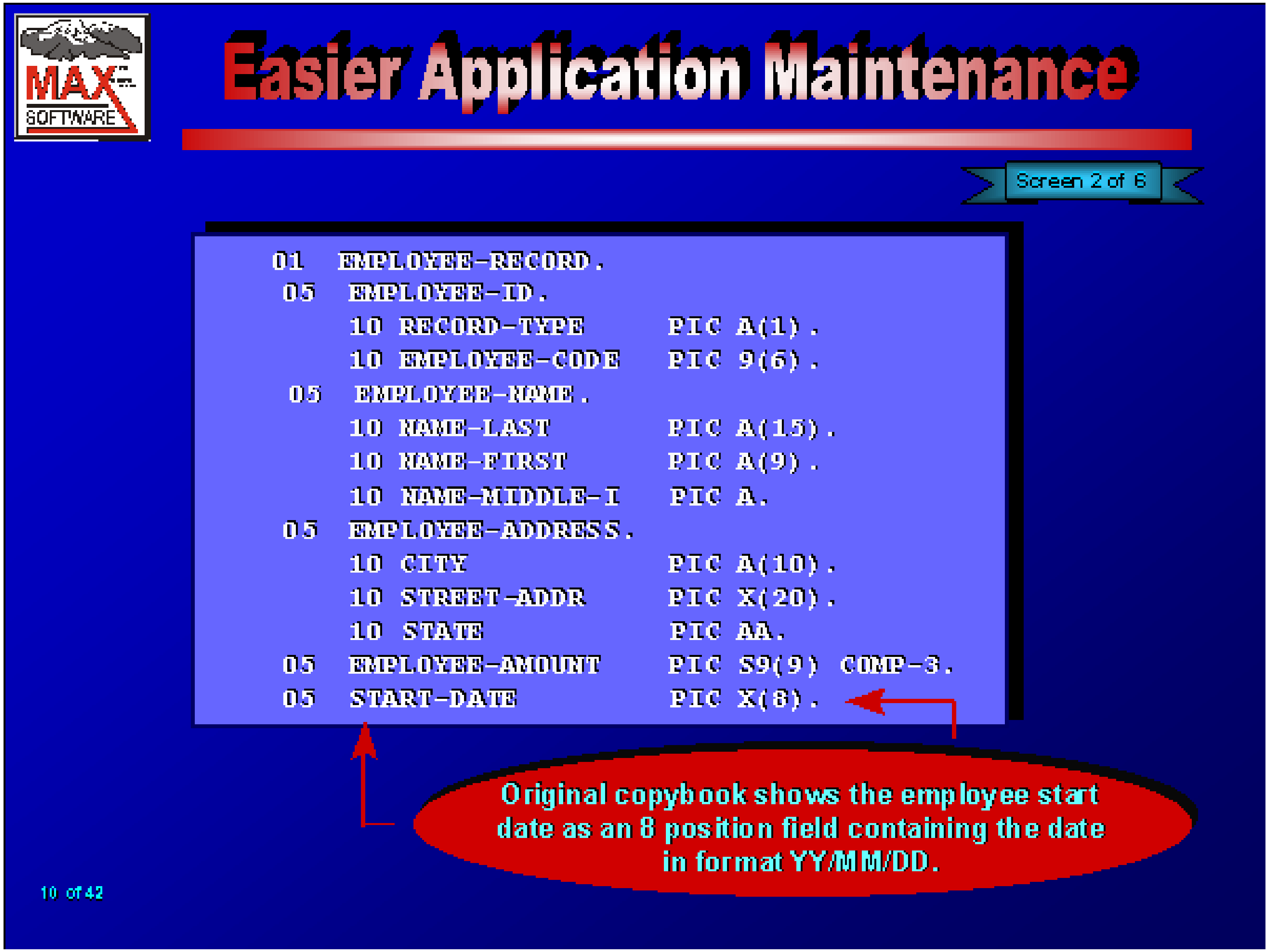
MAX SOFTWARE

Easier Application Maintenance

The following is an entire MAX/REXX Program

Screen 1 of 6

```
/* REXX */
/* DOC: produce a report of employee and their start dates          */

  IF ADDRESS()="TSO" THEN "ALLOC FI(SYSUT1) DA('MXS.USER.FEDS') SHR"
                                                        /*           */
/* OPEN the file using a copybook and the field names can be used   */
/*       to access the data in each field of the record              */
                                                        /*           */
  IF 'RXVSAM'("OPEN FILE(SYSUT1) COPYBOOK(MXS.P390.COPYLIB(CEHUR2))",
              " SEQ")<>0 THEN DO
     SAY 'RC='VSAMCODE 'MSG='VSAMMSG
     CALL CLEANUP
  END
                                                        /*           */
/* use MAX/REXX to provide a report of employee name and start date  */
                                                        /*           */
  DO WHILE 'RXVSAM'("READNEXT FILE(SYSUT1)")=0
    SAY 'EMPLOYEE:'NAME_LAST','NAME_FIRST '        ',
        'STARTING DATE:'START_DATE              /* show start date    */
  END
  CALL CLEANUP


CLEANUP:
   CALL 'RXVSAM' "CLOSE FILE(SYSUT1)"
   IF ADDRESS()="TSO" THEN "FREE FI(SYSUT1)"
   EXIT 0
```



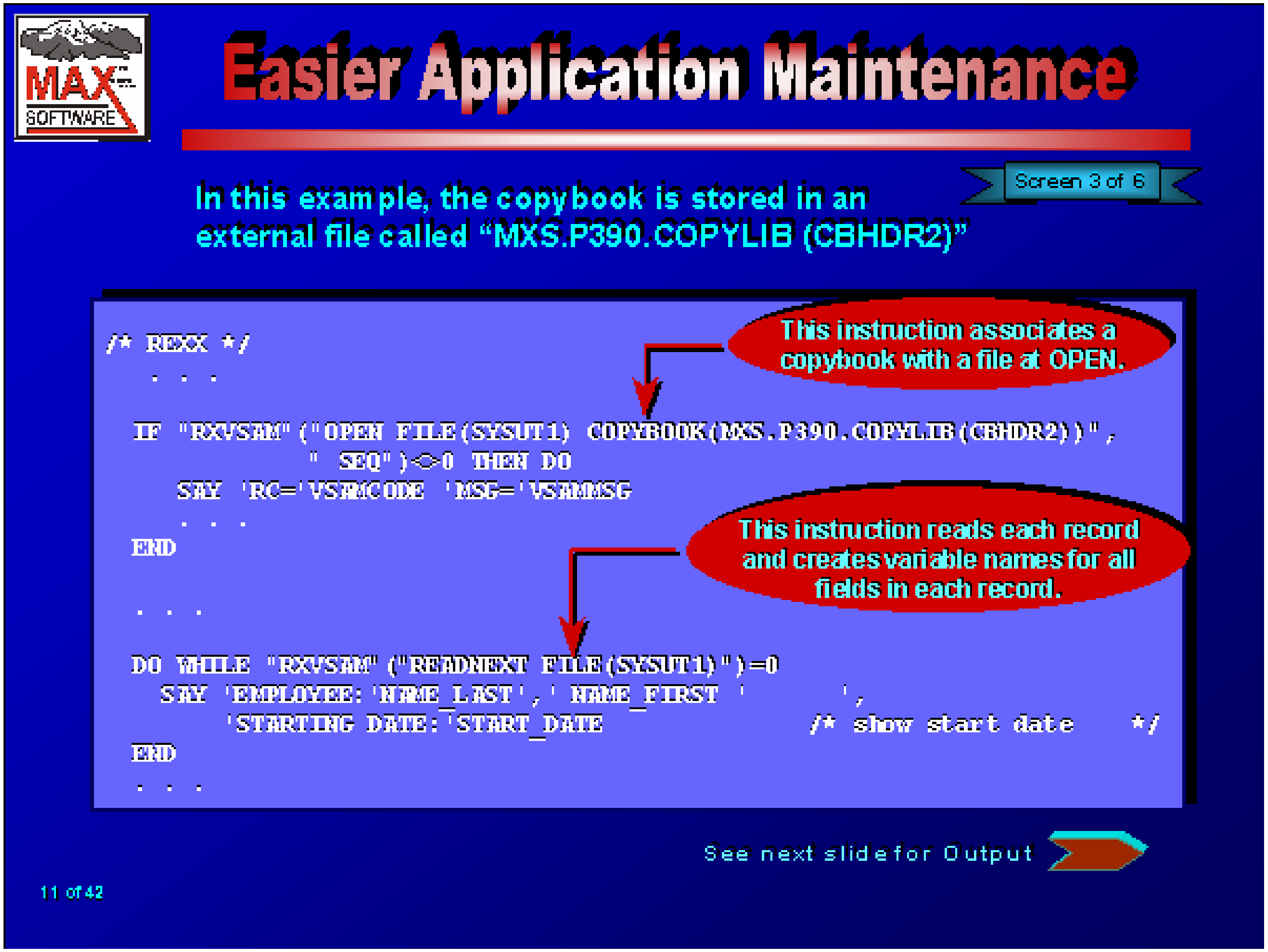
Easier Application Maintenance

Screen 2 of 6

```
   01  EMPLOYEE-RECORD.
    05  EMPLOYEE-ID.
        10 RECORD-TYPE        PIC A(1).
        10 EMPLOYEE-CODE      PIC 9(6).
     05  EMPLOYEE-NAME.
        10 NAME-LAST          PIC A(15).
        10 NAME-FIRST         PIC A(9).
        10 NAME-MIDDLE-I      PIC A.
    05  EMPLOYEE-ADDRESS.
        10 CITY               PIC A(10).
        10 STREET-ADDR        PIC X(20).
        10 STATE              PIC AA.
    05  EMPLOYEE-AMOUNT       PIC S9(9) COMP-3.
    05  START-DATE            PIC X(8).
```

Original copybook shows the employee start date as an 8 position field containing the date in format YY/MM/DD.

10 of 42



MAX SOFTWARE

Easier Application Maintenance

Screen 3 of 6

In this example, the copybook is stored in an external file called "MXS.P390.COPYLIB (CBHDR2)"

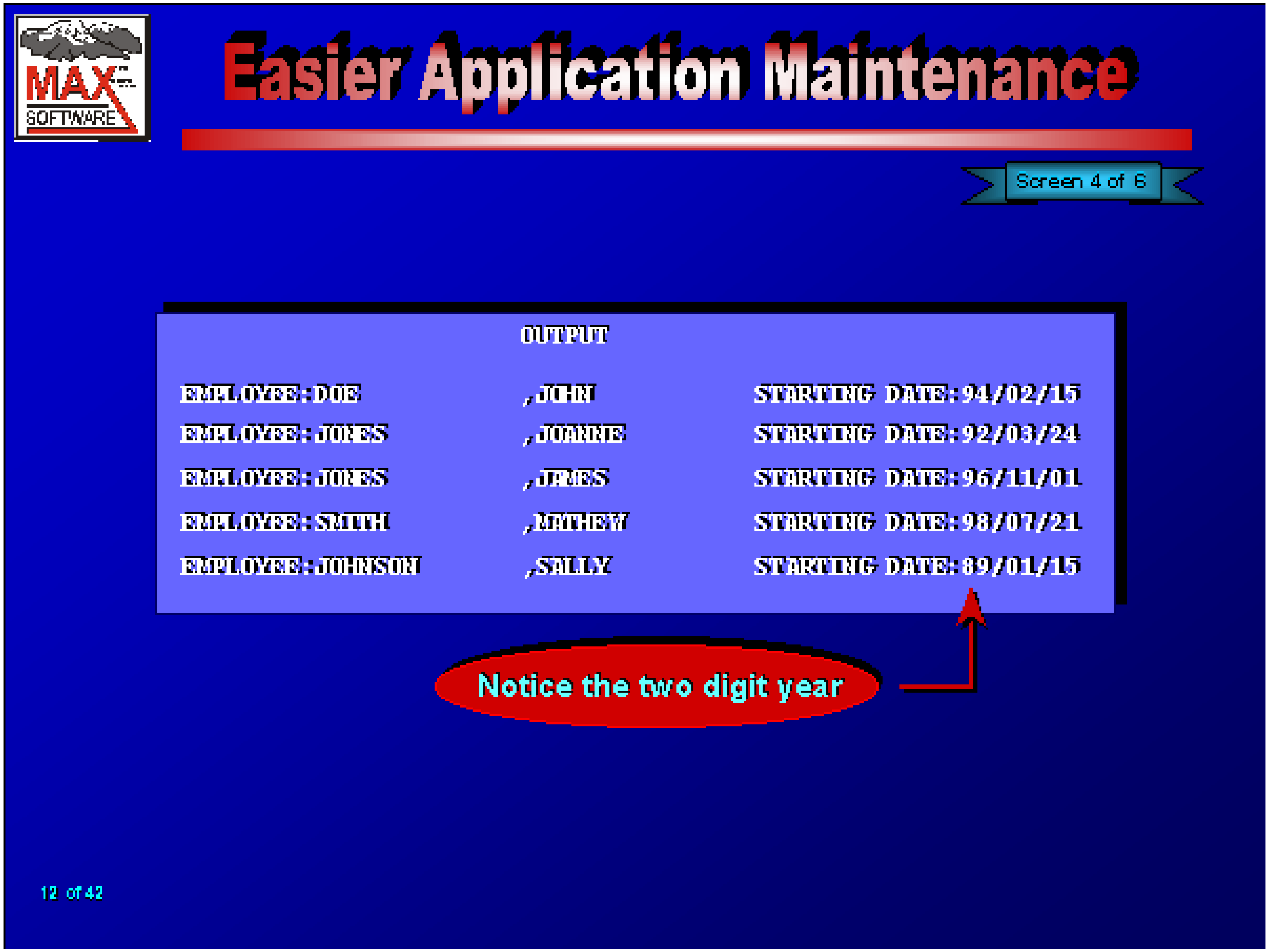
```
/* REXX */
   . . .

  IF "RXVSAM"("OPEN FILE(SYSUT1) COPYBOOK(MXS.P390.COPYLIB(CBHDR2))",
              " SEQ")<>0 THEN DO
     SAY 'RC='VSAMCODE 'MSG='VSAMMSG
     . . .
  END

  . . .

  DO WHILE "RXVSAM"("READNEXT FILE(SYSUT1)")=0
    SAY 'EMPLOYEE:'NAME_LAST','NAME_FIRST '        ',
        'STARTING DATE:'START_DATE              /* show start date    */
  END
  . . .
```

This instruction associates a copybook with a file at OPEN.

This instruction reads each record and creates variable names for all fields in each record.

See next slide for Output

11 of 42



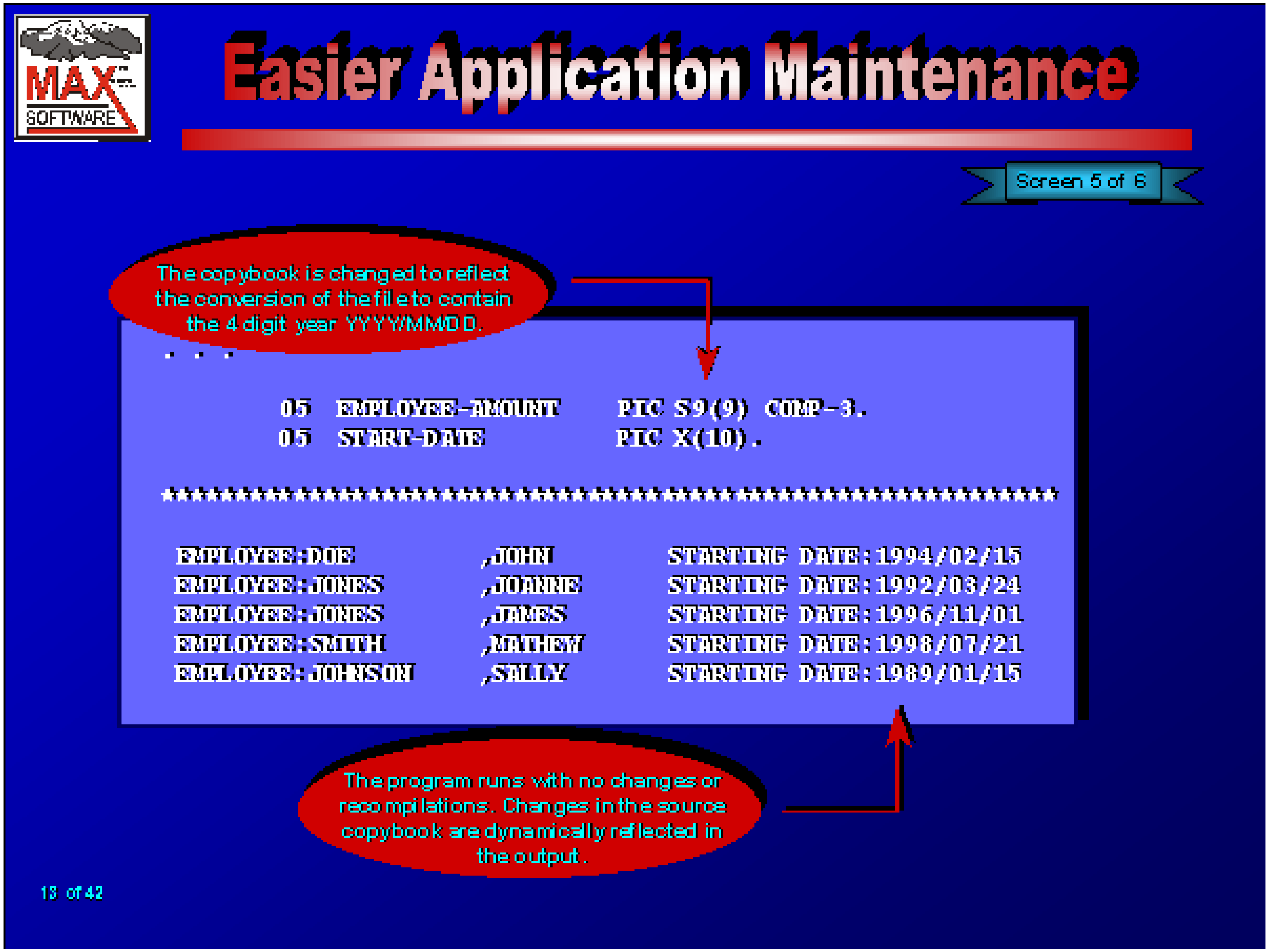
Easier Application Maintenance

Screen 4 of 6

```
                          OUTPUT

EMPLOYEE:DOE             ,JOHN         STARTING DATE:94/02/15
EMPLOYEE:JONES           ,JOANNE       STARTING DATE:92/03/24
EMPLOYEE:JONES           ,JAMES        STARTING DATE:96/11/01
EMPLOYEE:SMITH           ,MATHEW       STARTING DATE:98/07/21
EMPLOYEE:JOHNSON         ,SALLY        STARTING DATE:89/01/15
```

Notice the two digit year

12 of 42



Easier Application Maintenance

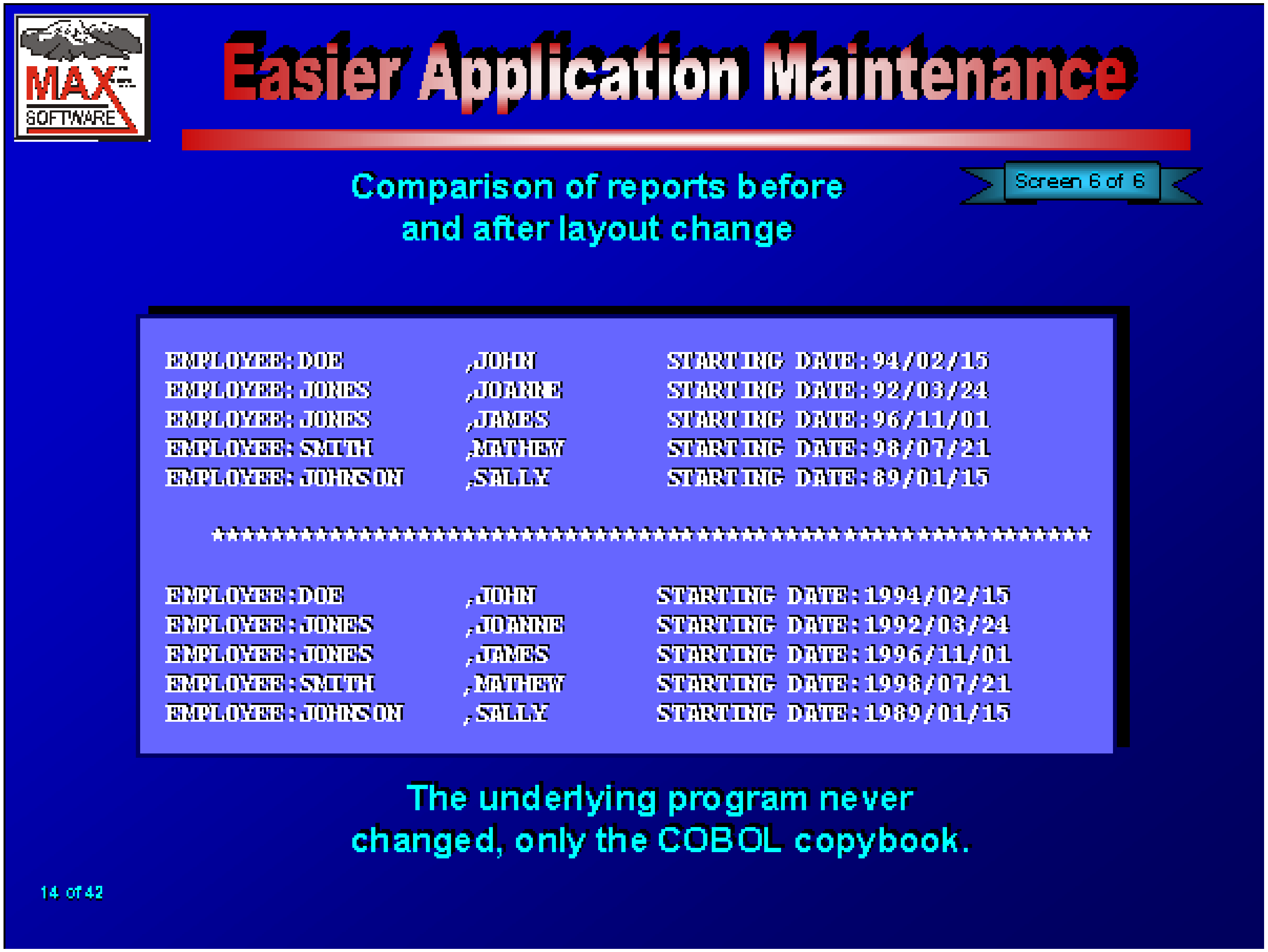
Screen 5 of 6

The copybook is changed to reflect the conversion of the file to contain the 4 digit year YYYY/MM/DD.

```
. . .

        05  EMPLOYEE-AMOUNT        PIC S9(9) COMP-3.
        05  START-DATE             PIC X(10).

*****************************************************************

EMPLOYEE:DOE            ,JOHN          STARTING DATE:1994/02/15
EMPLOYEE:JONES          ,JOANNE        STARTING DATE:1992/03/24
EMPLOYEE:JONES          ,JAMES         STARTING DATE:1996/11/01
EMPLOYEE:SMITH          ,MATHEW        STARTING DATE:1998/07/21
EMPLOYEE:JOHNSON        ,SALLY         STARTING DATE:1989/01/15
```

The program runs with no changes or recompilations. Changes in the source copybook are dynamically reflected in the output.

13 of 42



Easier Application Maintenance

Screen 6 of 6

Comparison of reports before and after layout change

```
EMPLOYEE: DOE          ,JOHN          STARTING DATE: 94/02/15
EMPLOYEE: JONES        ,JOANNE        STARTING DATE: 92/03/24
EMPLOYEE: JONES        ,JAMES         STARTING DATE: 96/11/01
EMPLOYEE: SMITH        ,MATHEW        STARTING DATE: 98/07/21
EMPLOYEE: JOHNSON      ,SALLY         STARTING DATE: 89/01/15

   ************************************************************

EMPLOYEE :DOE          ,JOHN         STARTING DATE: 1994/02/15
EMPLOYEE :JONES        ,JOANNE       STARTING DATE: 1992/03/24
EMPLOYEE :JONES        ,JAMES        STARTING DATE: 1996/11/01
EMPLOYEE :SMITH        ,MATHEW       STARTING DATE: 1998/07/21
EMPLOYEE :JOHNSON      ,SALLY        STARTING DATE: 1989/01/15
```

The underlying program never changed, only the COBOL copybook.

14 of 42

Faster Program Testing

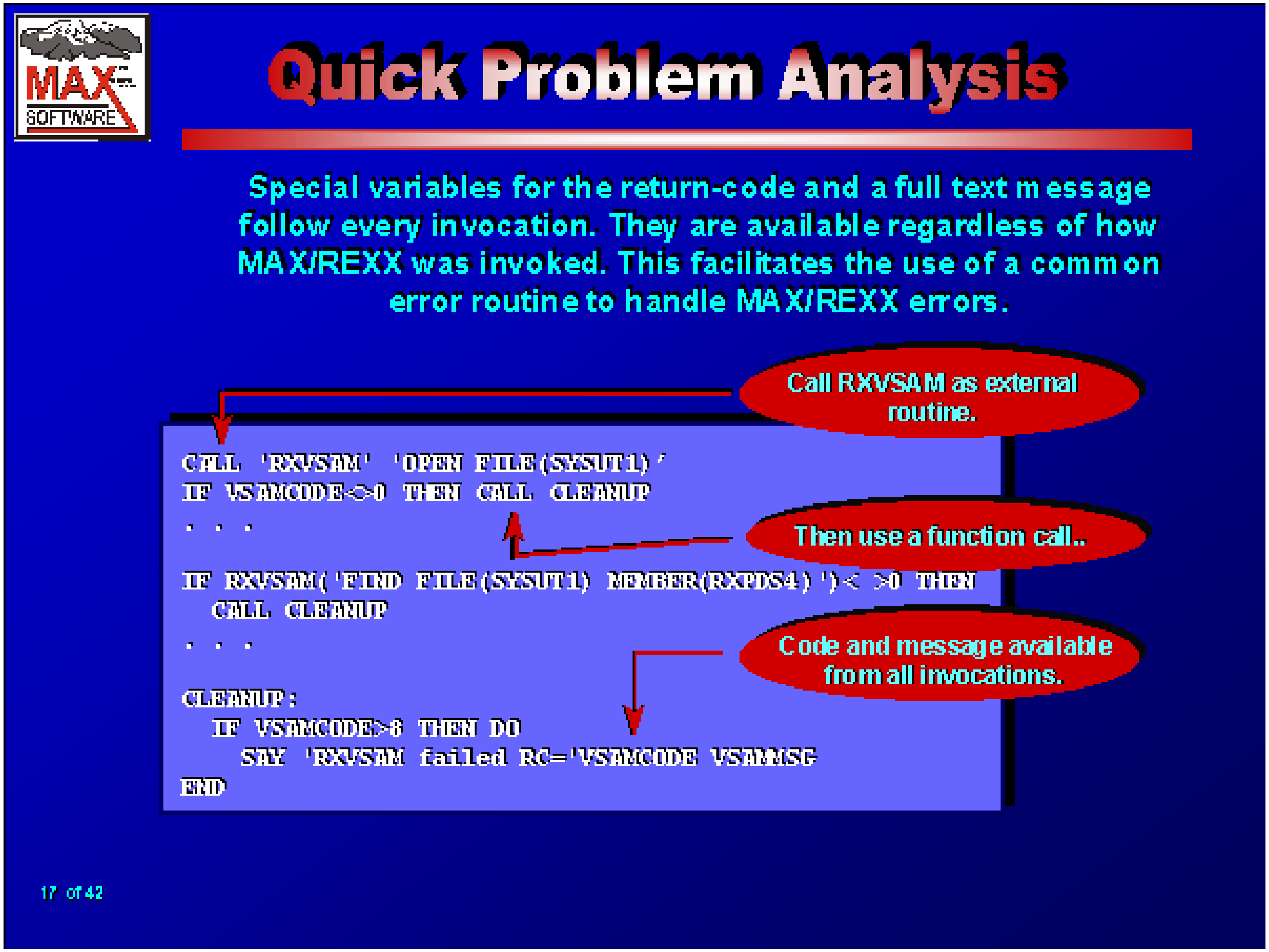
- MAX/REXX programs are much smaller than traditional programs, resulting in fewer opportunities for failure.
- Built-in interactive trace capabilities facilitate faster testing and debugging.

Faster Program Testing

All MAX/REXX components optionally provide trace feedback.

```
OPEN FILE(SYSUT1) COPYBOOK(MXS.P390.COPYLIB(CBHDR))
SEQUPD
RC=00000000 RXV0004I SUCCESSFUL COMPLETION
READNEXT FILE(SYSUT1) WHERE(10,EQ,'Y')
RC=00000000 RXV0004I SUCCESSFUL COMPLETION
REWRITE FILE(SYSUT1)
RC=00000000 RXV0004I SUCCESSFUL COMPLETION
READNEXT FILE(SYSUT1) WHERE(10,EQ,'Y')
RC=00000000 RXV0004I SUCCESSFUL COMPLETION
REWRITE FILE(SYSUT1)
RC=00000000 RXV0004I SUCCESSFUL COMPLETION

. . .

READNEXT FILE(SYSUT1) WHERE(10,EQ,'Y')
RC=00000008 RXV0046I VSAM OPERATION FAILED OPER=NXT RC=EOF
CLOSE FILE(SYSUT1)
RC=00000000 RXV0004I SUCCESSFUL COMPLETION
. . .
```

TRACE provides both passed command and returned results

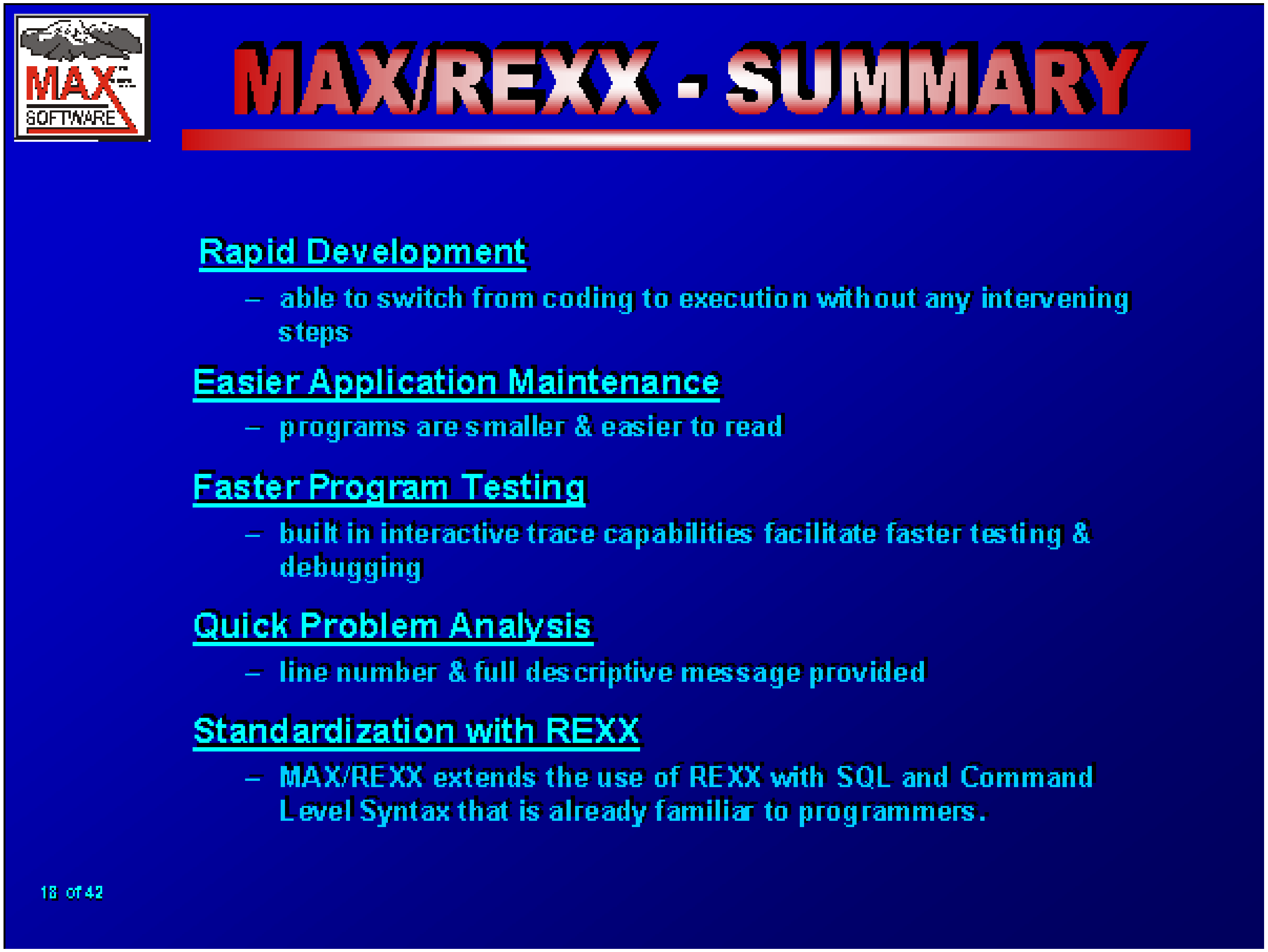


Quick Problem Analysis

Special variables for the return-code and a full text message follow every invocation. They are available regardless of how MAX/REXX was invoked. This facilitates the use of a common error routine to handle MAX/REXX errors.

```
CALL 'RXVSAM' 'OPEN FILE(SYSUT1)'
IF VSAMCODE<>0 THEN CALL CLEANUP
. . .

IF RXVSAM('FIND FILE(SYSUT1) MEMBER(RXPDS4)')< >0 THEN
  CALL CLEANUP
. . .

CLEANUP:
  IF VSAMCODE>8 THEN DO
    SAY 'RXVSAM failed RC='VSAMCODE VSAMMSG
END
```

Call RXVSAM as external routine.
Then use a function call..
Code and message available from all invocations.



MAX/REXX - SUMMARY

Rapid Development

- able to switch from coding to execution without any intervening steps

Easier Application Maintenance

- programs are smaller & easier to read

Faster Program Testing

- built in interactive trace capabilities facilitate faster testing & debugging

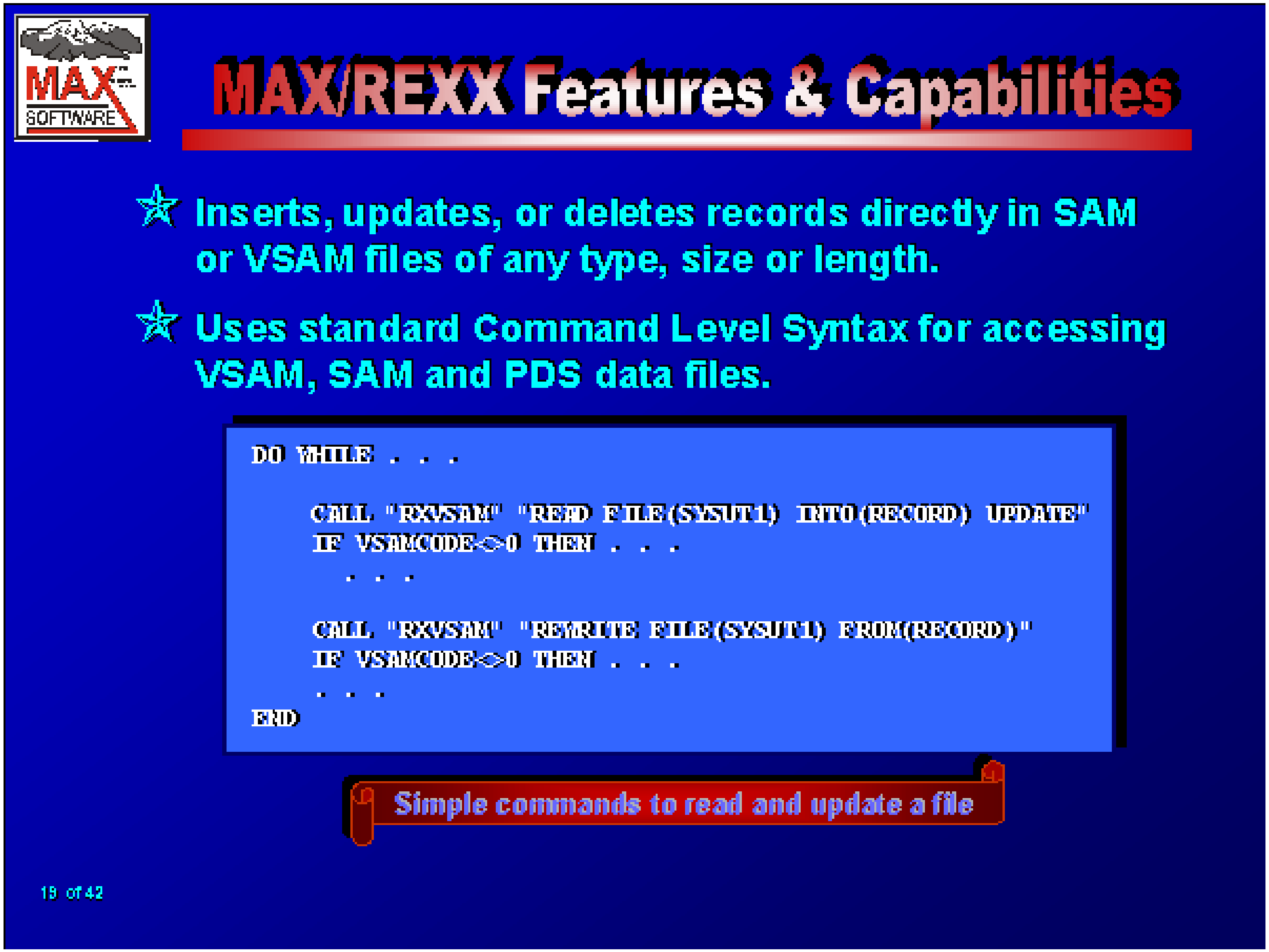
Quick Problem Analysis

- line number & full descriptive message provided

Standardization with REXX

- MAX/REXX extends the use of REXX with SQL and Command Level Syntax that is already familiar to programmers.

18 of 42

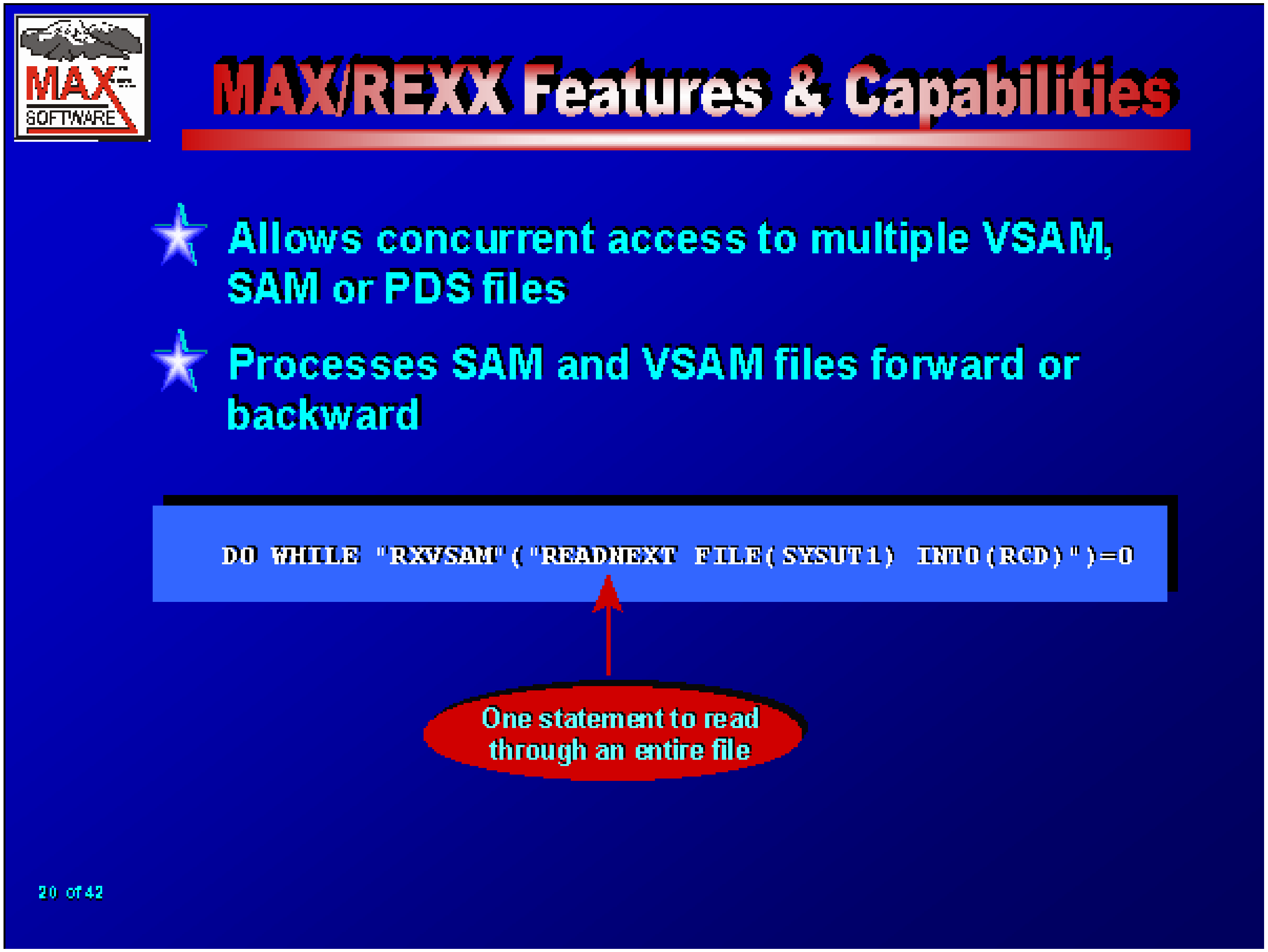


MAX/REXX Features & Capabilities

- ✯ Inserts, updates, or deletes records directly in SAM or VSAM files of any type, size or length.
- ✯ Uses standard Command Level Syntax for accessing VSAM, SAM and PDS data files.

```
DO WHILE . . .

     CALL "RXVSAM" "READ FILE(SYSUT1) INTO(RECORD) UPDATE"
     IF VSAMCODE<>0 THEN . . .
        . . .

     CALL "RXVSAM" "REWRITE FILE(SYSUT1) FROM(RECORD)"
     IF VSAMCODE<>0 THEN . . .
      . . .
END
```

Simple commands to read and update a file

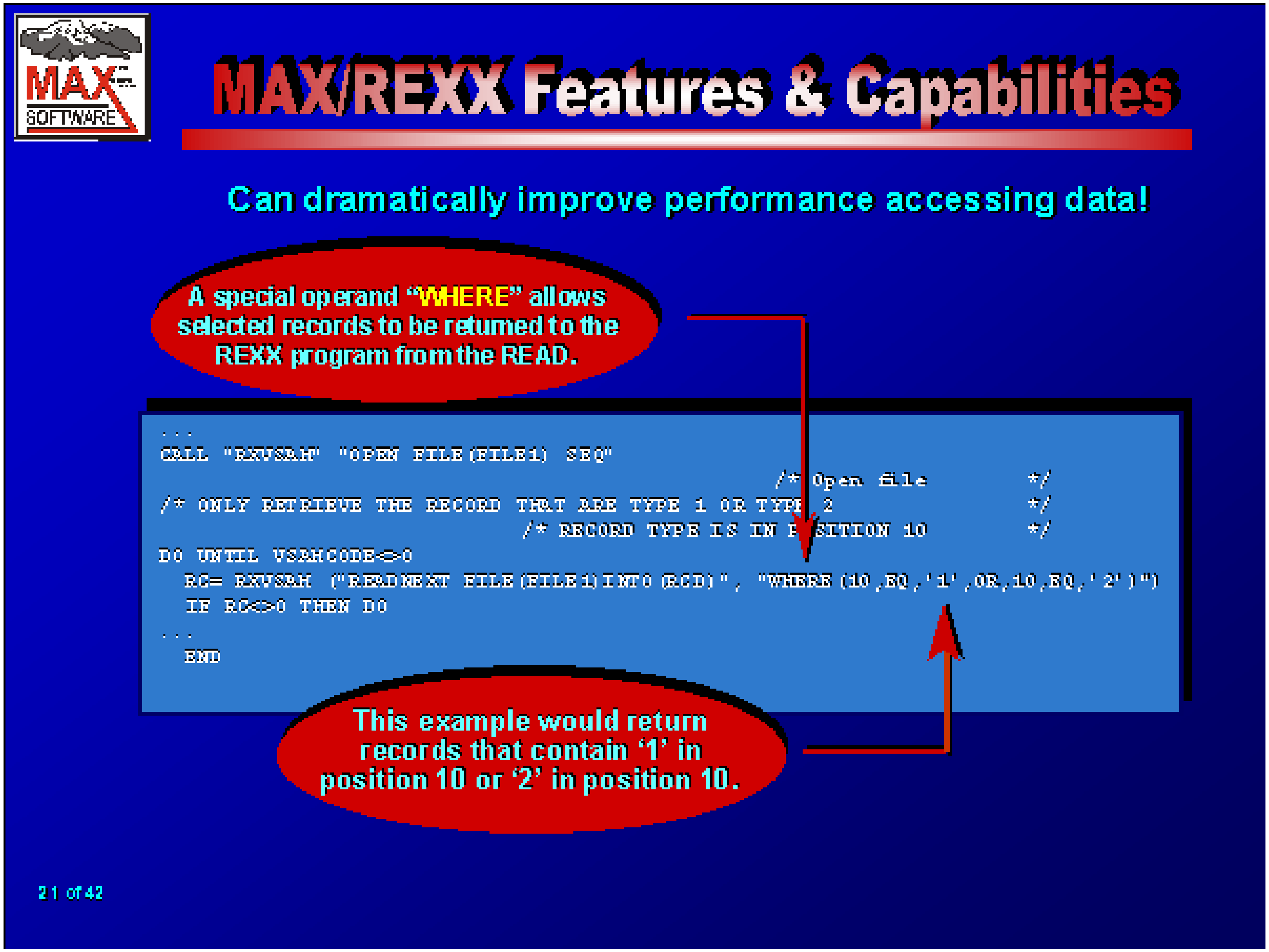


MAX/REXX Features & Capabilities

- Allows concurrent access to multiple VSAM, SAM or PDS files
- Processes SAM and VSAM files forward or backward

```
DO WHILE "RXVSAM"("READNEXT FILE(SYSUT1) INTO(RCD)")=0
```

One statement to read through an entire file



MAX/REXX Features & Capabilities

Can dramatically improve performance accessing data!

A special operand "WHERE" allows selected records to be returned to the REXX program from the READ.

```
...
CALL "RXVSAM" "OPEN FILE(FILE1) SEQ"
                                                            /* Open file         */
/* ONLY RETRIEVE THE RECORD THAT ARE TYPE 1 OR TYPE 2                            */
                                   /* RECORD TYPE IS IN POSITION 10              */
DO UNTIL VSAMCODE<>0
  RC= RXVSAM ("READNEXT FILE(FILE1)INTO(RCD)", "WHERE(10,EQ,'1',OR,10,EQ,'2')")
  IF RC<>0 THEN DO
...
  END
```

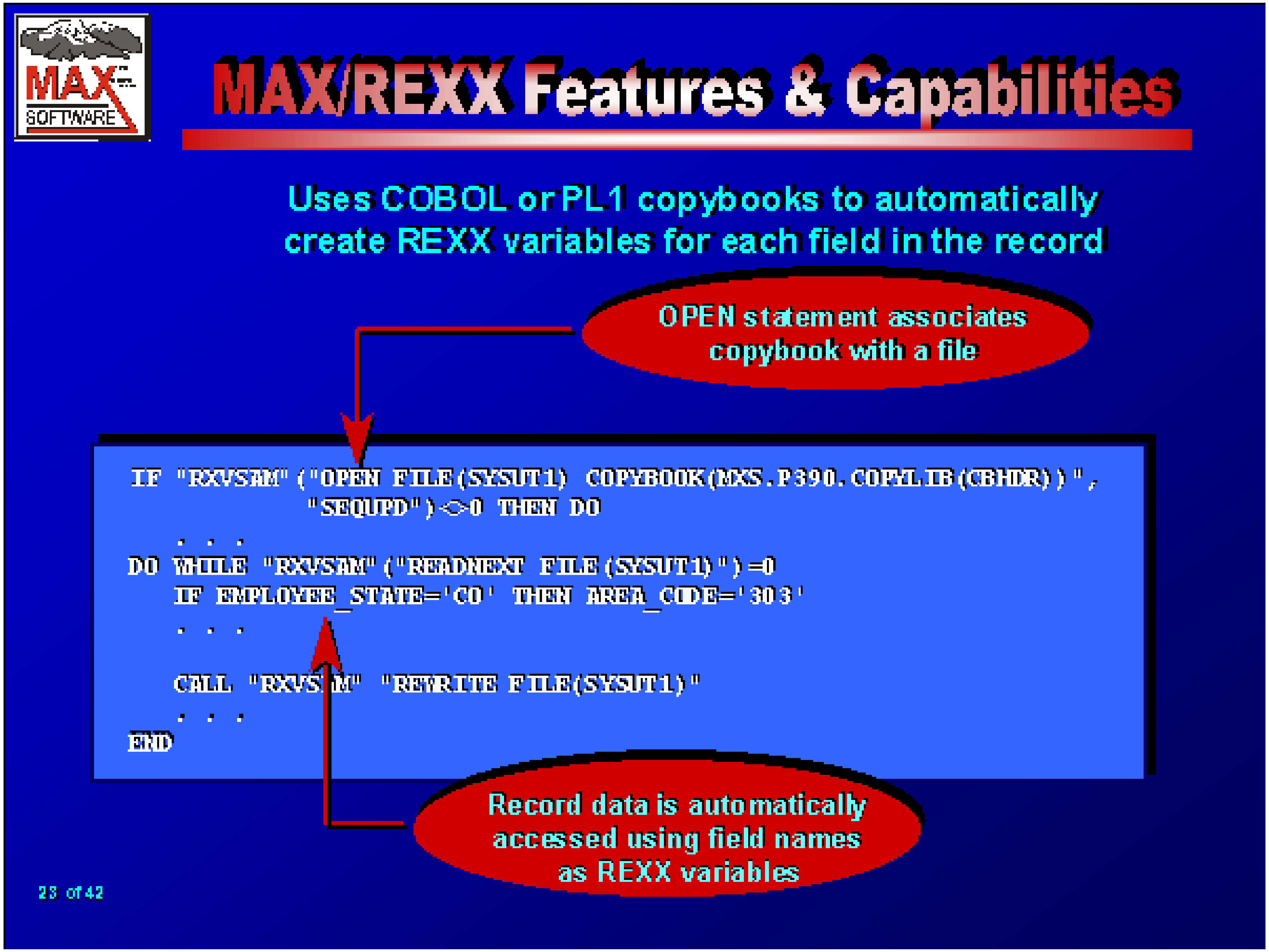
This example would return records that contain '1' in position 10 or '2' in position 10.

MAX/REXX Features & Capabilities

- Processes PDS directory information
- Process PDS members

```
MEM_NAME='TESTPGM1'
CALL 'RXVSAM' 'FIND FILE(SYSUT1) MEMBER('MEM_NAME')'
IF VSAMCODE<>0 THEN . . .
```

This statement will position to start of member



MAX/REXX Features & Capabilities

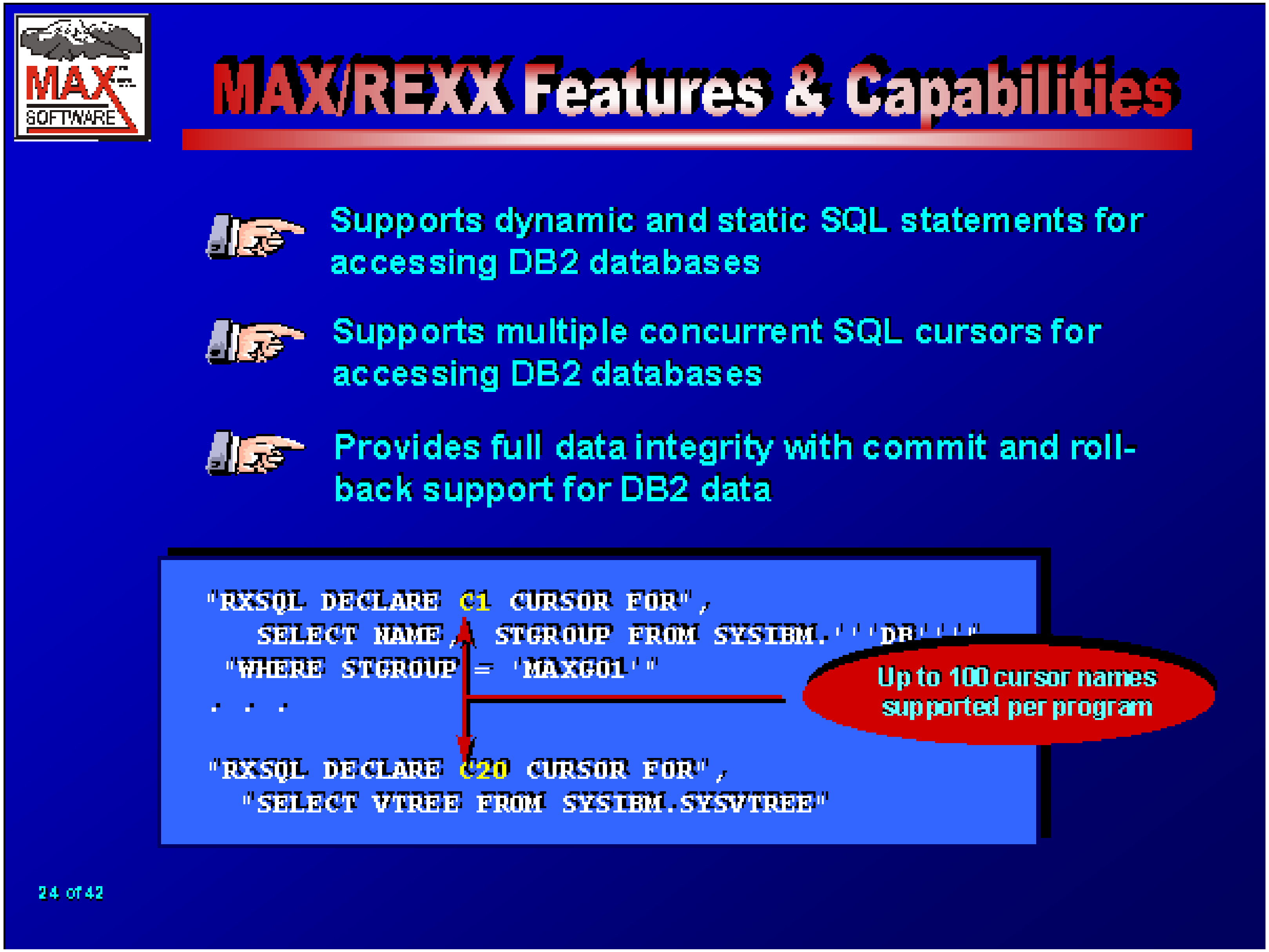
Uses COBOL or PL1 copybooks to automatically create REXX variables for each field in the record

OPEN statement associates copybook with a file

```
IF "RXVSAM"("OPEN FILE(SYSUT1) COPYBOOK(MXS.P390.COPYLIB(CBHDR))",
            "SEQUPD")<>0 THEN DO
   . . .
DO WHILE "RXVSAM"("READNEXT FILE(SYSUT1)")=0
   IF EMPLOYEE_STATE='CO' THEN AREA_CODE='303'
   . . .

   CALL "RXVSAM" "REWRITE FILE(SYSUT1)"
   . . .
END
```

Record data is automatically accessed using field names as REXX variables

23 of 42



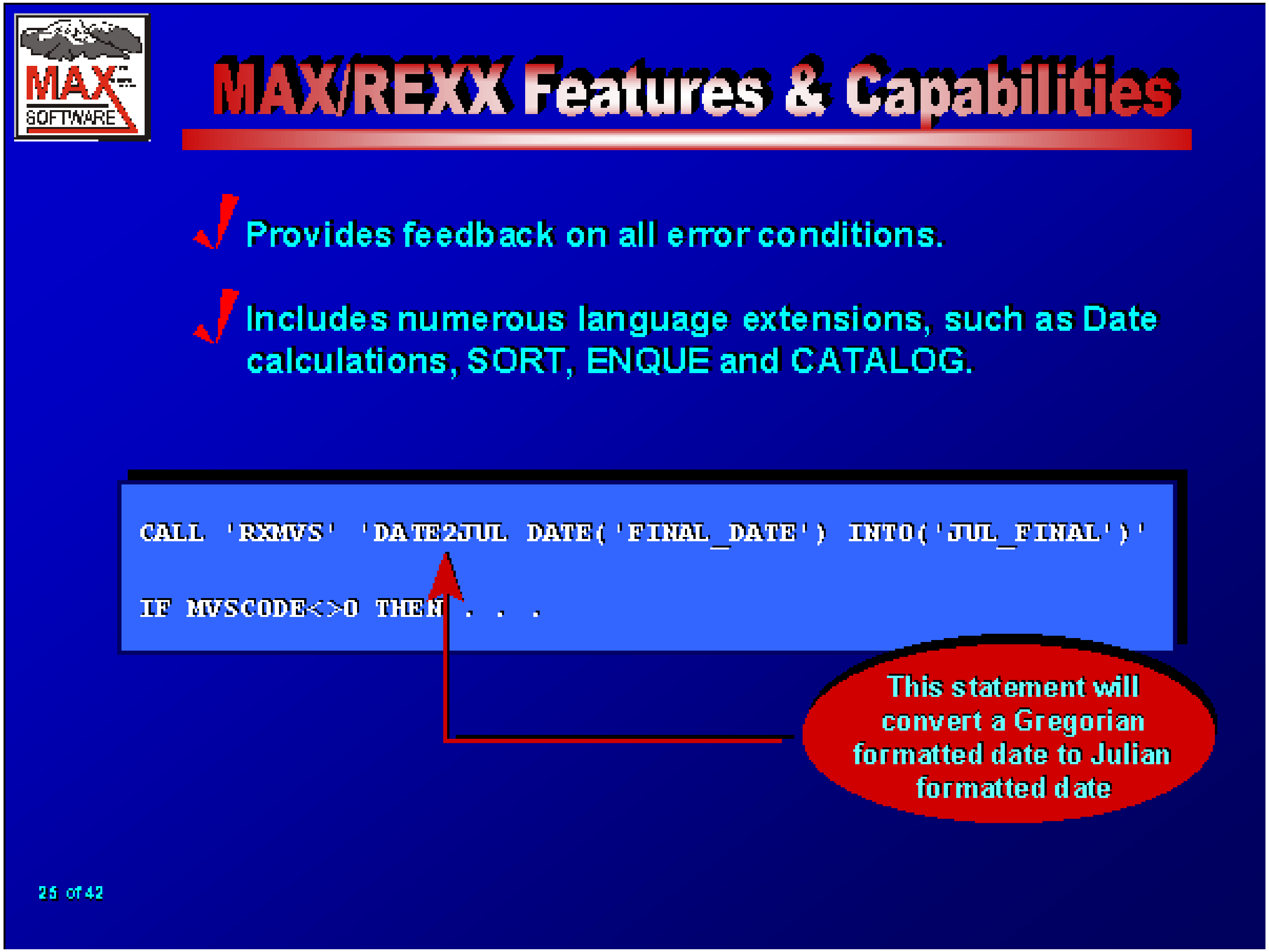
MAX/REXX Features & Capabilities

- Supports dynamic and static SQL statements for accessing DB2 databases
- Supports multiple concurrent SQL cursors for accessing DB2 databases
- Provides full data integrity with commit and roll-back support for DB2 data

```
"RXSQL DECLARE C1 CURSOR FOR",
   SELECT NAME,  STGROUP FROM SYSIBM.'''DB'''"
 "WHERE STGROUP = 'MAXG01''"
. . .

"RXSQL DECLARE C20 CURSOR FOR",
  "SELECT VTREE FROM SYSIBM.SYSVTREE"
```

Up to 100 cursor names supported per program

24 of 42



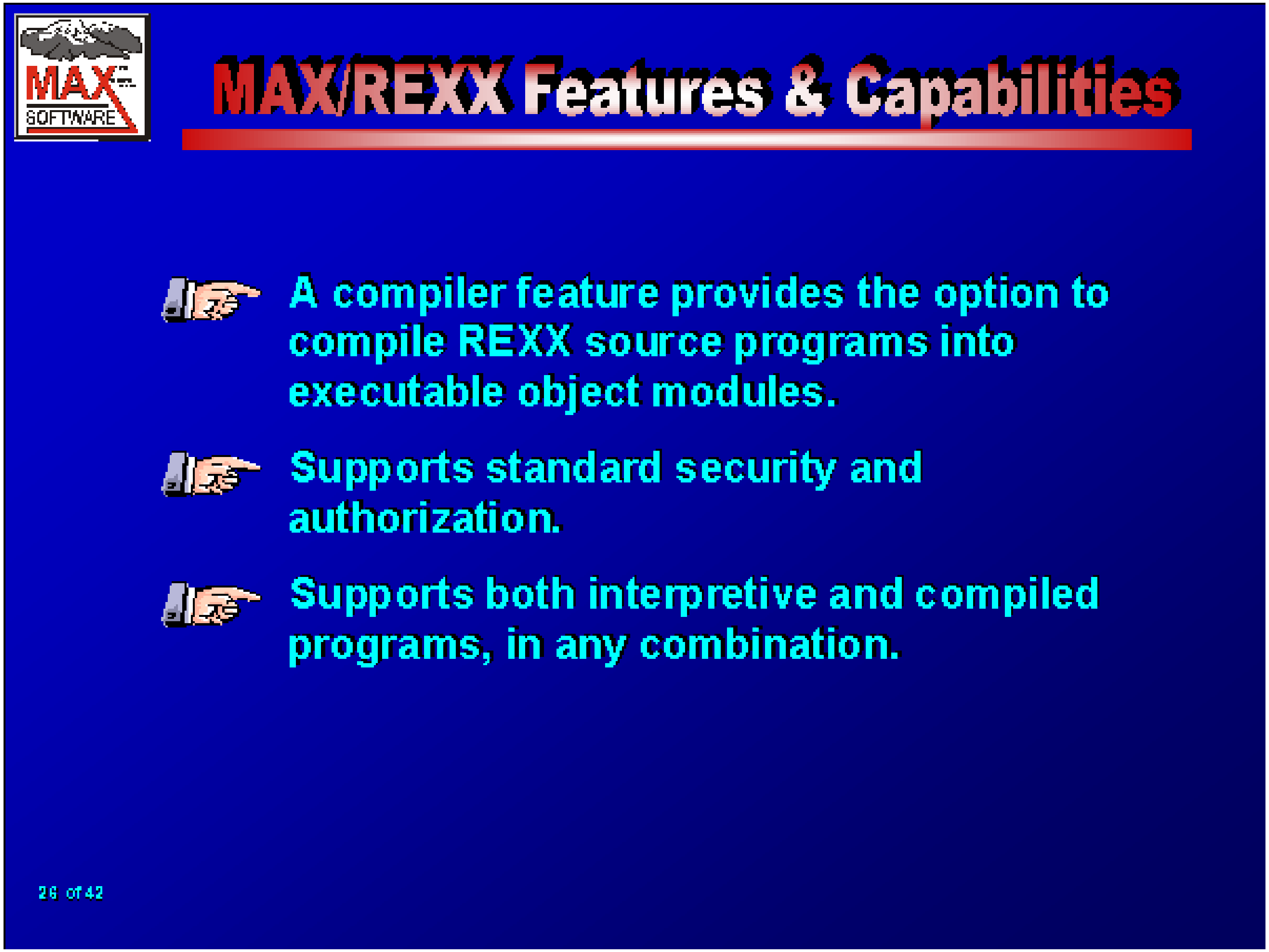
MAX/REXX Features & Capabilities

- Provides feedback on all error conditions.
- Includes numerous language extensions, such as Date calculations, SORT, ENQUE and CATALOG.

```
CALL 'RXMVS' 'DATE2JUL DATE('FINAL_DATE') INTO('JUL_FINAL')'

IF MVSCODE<>0 THEN . . .
```

This statement will convert a Gregorian formatted date to Julian formatted date

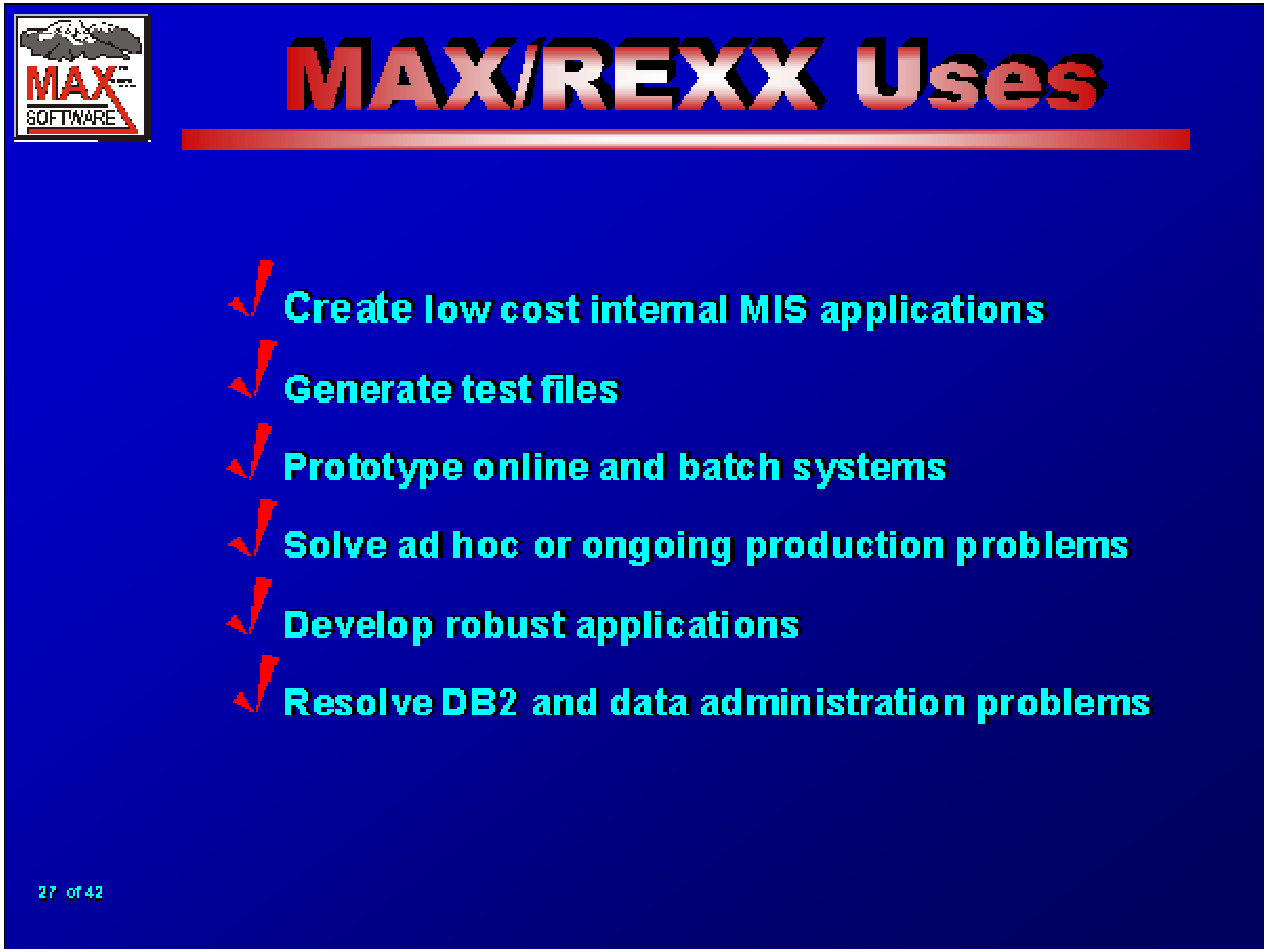
25 of 42



MAX/REXX Features & Capabilities

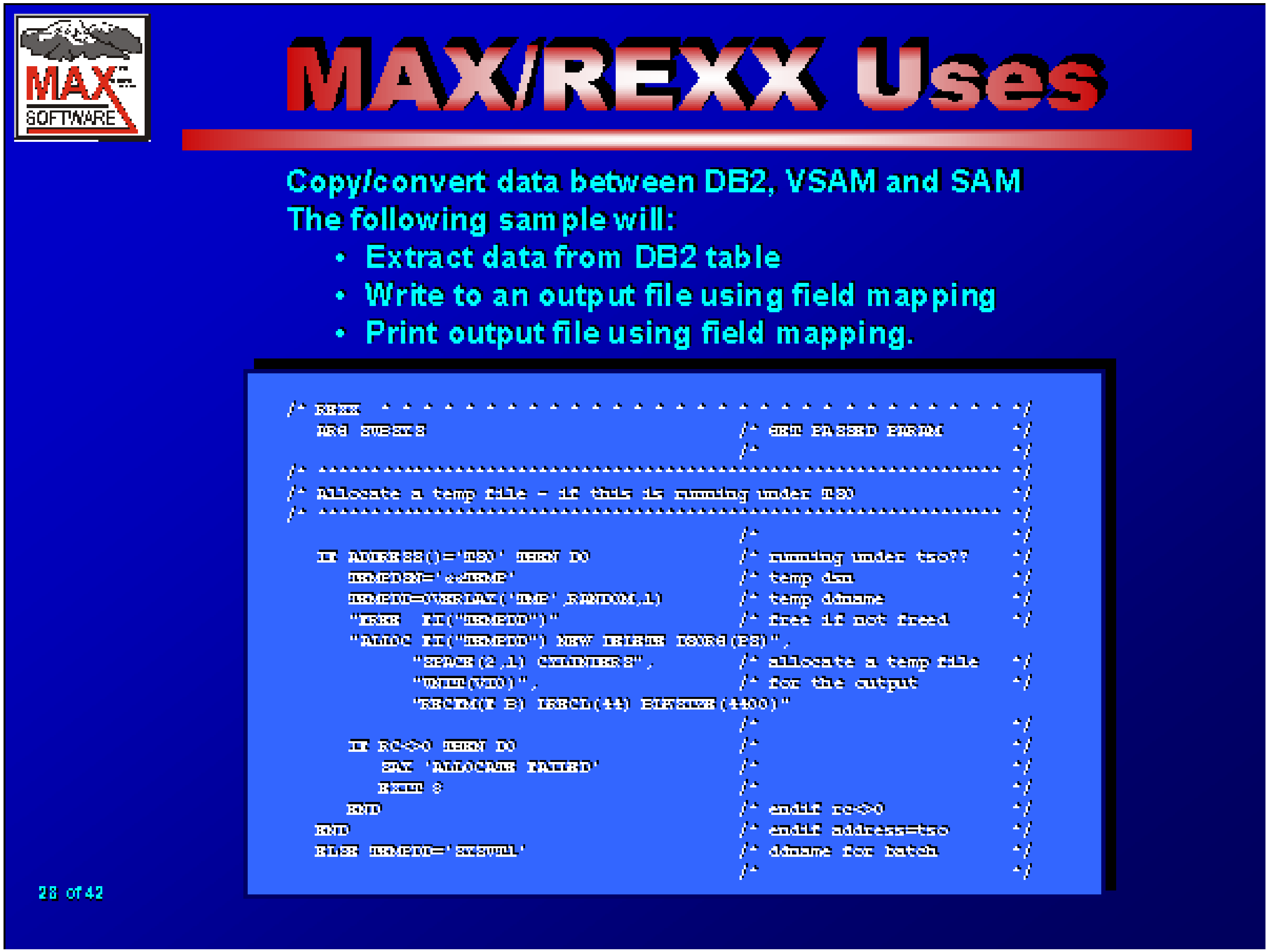
- A compiler feature provides the option to compile REXX source programs into executable object modules.
- Supports standard security and authorization.
- Supports both interpretive and compiled programs, in any combination.

26 of 42



MAX/REXX Uses

- Create low cost internal MIS applications
- Generate test files
- Prototype online and batch systems
- Solve ad hoc or ongoing production problems
- Develop robust applications
- Resolve DB2 and data administration problems



MAX/REXX Uses

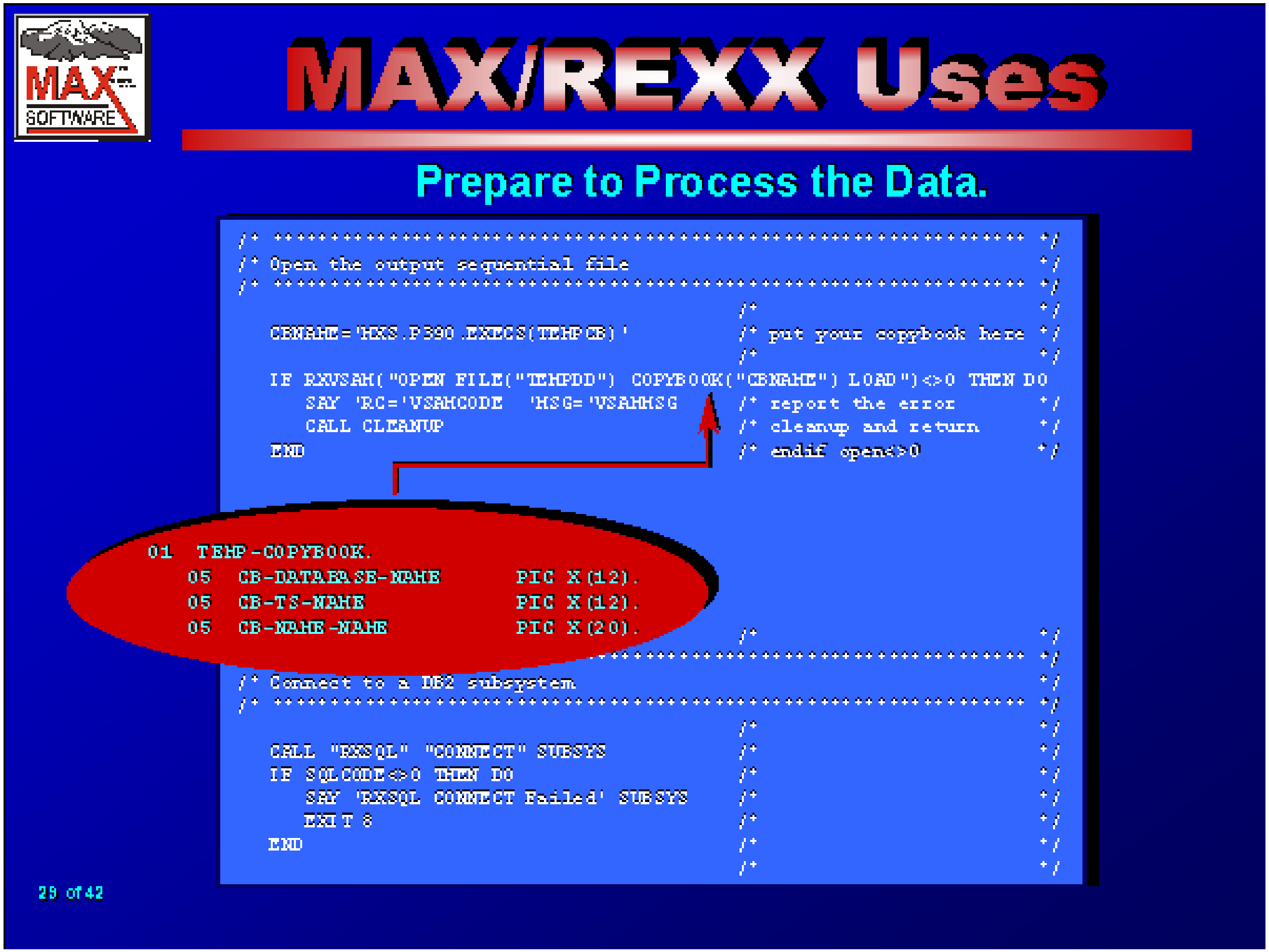
Copy/convert data between DB2, VSAM and SAM

The following sample will:

- Extract data from DB2 table
- Write to an output file using field mapping
- Print output file using field mapping.

```
/* REXX  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
   ARG SUBSYS                                     /* GET PASSED PARM     */
                                                  /*                     */
/* ****************************************************************** */
/* Allocate a temp file - if this is running under TSO                */
/* ****************************************************************** */
                                                  /*                     */
   IF ADDRESS()='TSO' THEN DO                     /* running under tso?? */
      TEMPDSN='&&TEMP'                            /* temp dsn            */
      TEMPDD=OVERLAY('TMP',RANDOM,1)              /* temp ddname         */
      "FREE  FI("TEMPDD")"                        /* free if not freed   */
      "ALLOC FI("TEMPDD") NEW DELETE DSORG(PS)",
            "SPACE(2,1) CYLINDERS",               /* allocate a temp file  */
            "UNIT(VIO)",                          /* for the output      */
            "RECFM(F B) LRECL(44) BLKSIZE(4400)"
                                                  /*                     */
      IF RC<>0 THEN DO                            /*                     */
         SAY 'ALLOCATE FAILED'                    /*                     */
         EXIT 8                                   /*                     */
      END                                         /* endif rc<>0         */
   END                                            /* endif address=tso   */
   ELSE TEMPDD='SYSUT1'                           /* ddname for batch    */
                                                  /*                     */
```

28 of 42



MAX/REXX Uses

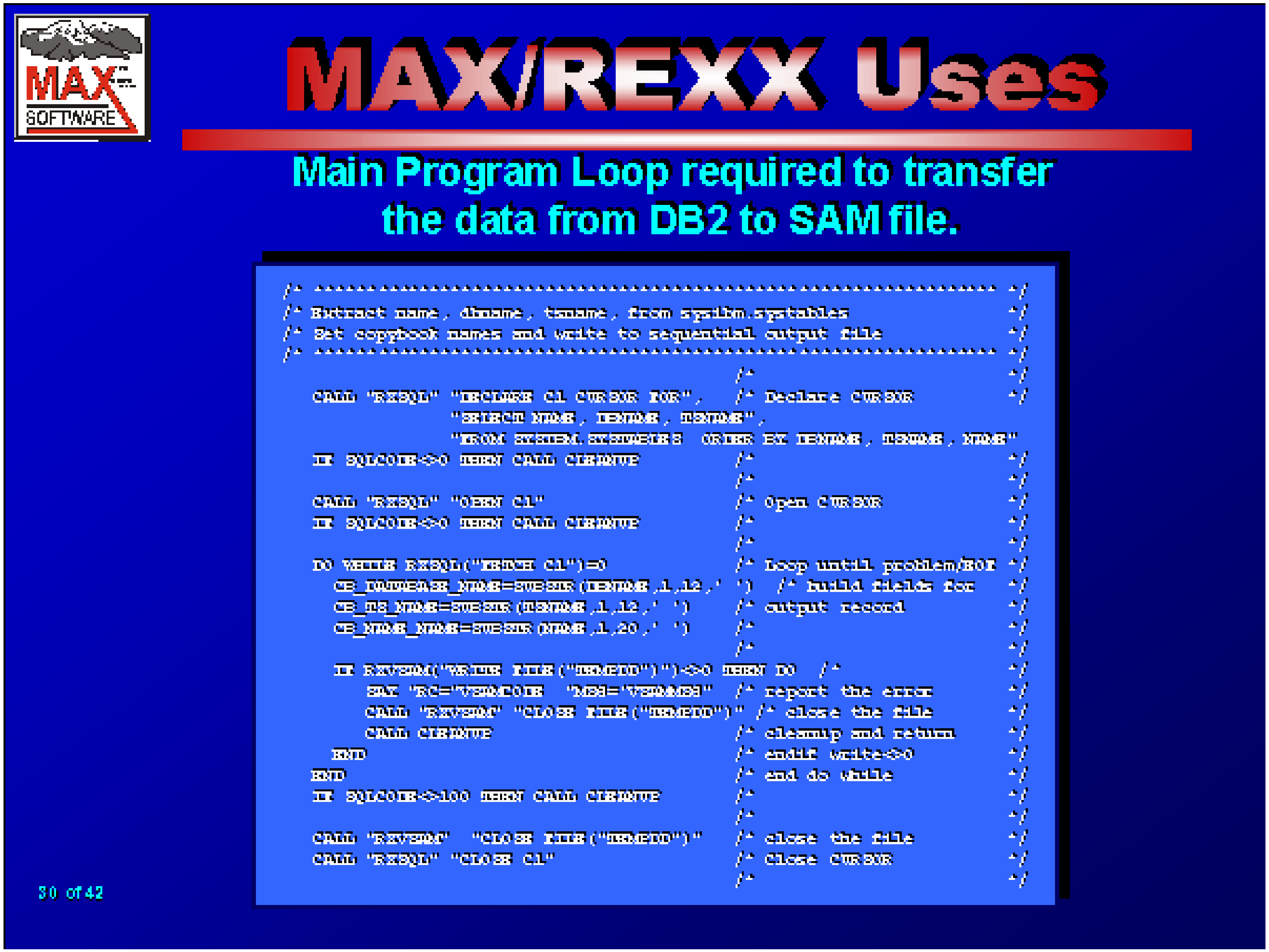
Prepare to Process the Data.

```
/* ************************************************************* */
/* Open the output sequential file                                */
/* ************************************************************* */
                                                /*                 */
   CBNAME='MXS.P390.EXECS(TEMPCB)'              /* put your copybook here */
                                                /*                 */
   IF RXVSAM("OPEN FILE("TEMPDD") COPYBOOK("CBNAME") LOAD")<>0 THEN DO
      SAY 'RC='VSAMCODE  'MSG='VSAMMSG          /* report the error      */
      CALL CLEANUP                              /* cleanup and return    */
   END                                          /* endif open<>0         */
```

```
01  TEMP-COPYBOOK.
    05  CB-DATABASE-NAME        PIC X(12).
    05  CB-TS-NAME              PIC X(12).
    05  CB-NAME-NAME            PIC X(20).
```

```
                                                /*                 */
                                ************************************ */
/* Connect to a DB2 subsystem                                     */
/* ************************************************************* */
                                                /*                 */
   CALL "RXSQL" "CONNECT" SUBSYS                /*                 */
   IF SQLCODE<>0 THEN DO                        /*                 */
      SAY 'RXSQL CONNECT Failed' SUBSYS         /*                 */
      EXIT 8                                    /*                 */
   END                                          /*                 */
                                                /*                 */
```

29 of 42



MAX/REXX Uses

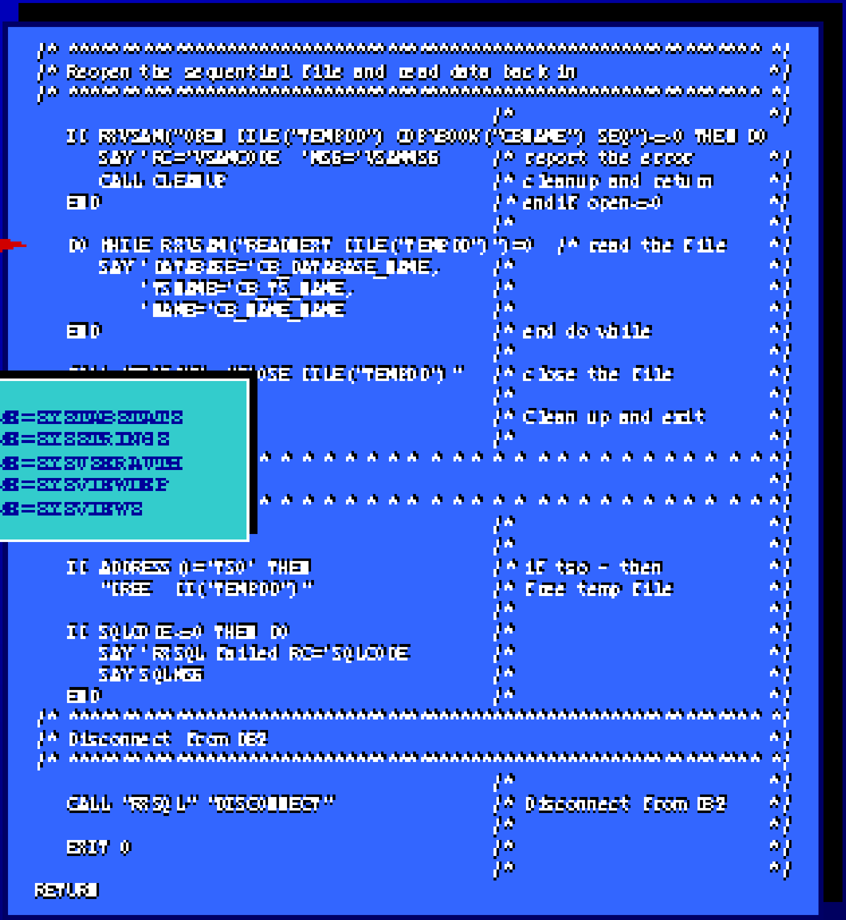
Main Program Loop required to transfer the data from DB2 to SAM file.

```
/* ******************************************************************** */
/* Extract name, dbname, tsname, from sysibm.systables                  */
/* Set copybook names and write to sequential output file               */
/* ******************************************************************** */
                                             /*                         */
   CALL "RXSQL" "DECLARE C1 CURSOR FOR",     /* Declare CURSOR          */
                "SELECT NAME, DBNAME, TSNAME",
                "FROM SYSIBM.SYSTABLES  ORDER BY DBNAME, TSNAME, NAME"
   IF SQLCODE<>0 THEN CALL CLEANUP           /*                         */
                                             /*                         */
   CALL "RXSQL" "OPEN C1"                    /* Open CURSOR             */
   IF SQLCODE<>0 THEN CALL CLEANUP           /*                         */
                                             /*                         */
   DO WHILE RXSQL("FETCH C1")=0              /* Loop until problem/EOF  */
     CB_DATABASE_NAME=SUBSTR(DBNAME,1,12,' ')  /* build fields for      */
     CB_TS_NAME=SUBSTR(TSNAME,1,12,' ')      /* output record           */
     CB_NAME_NAME=SUBSTR(NAME,1,20,' ')      /*                         */
                                             /*                         */
     IF RXVSAM("WRITE FILE("TEMPDD")")<>0 THEN DO  /*                   */
        SAY "RC="VSAMCODE  "MSG="VSAMMSG"    /* report the error        */
        CALL "RXVSAM" "CLOSE FILE("TEMPDD")" /* close the file          */
        CALL CLEANUP                         /* cleanup and return      */
     END                                     /* endif write<>0          */
   END                                       /* end do while            */
   IF SQLCODE<>100 THEN CALL CLEANUP         /*                         */
                                             /*                         */
   CALL "RXVSAM"  "CLOSE FILE("TEMPDD")"     /* close the file          */
   CALL "RXSQL" "CLOSE C1"                   /* Close CURSOR            */
                                             /*                         */
```

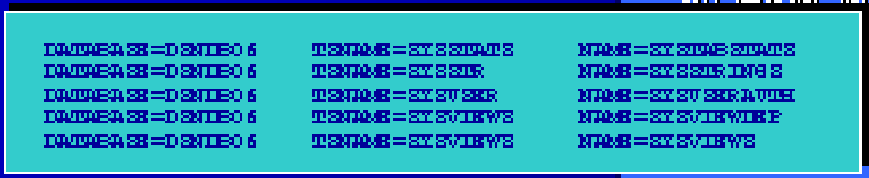
30 of 42

MAX/REXX Uses

Display Output Results



```
/* ************************************************************ */
/* Reopen the sequential file and read data back in             */
/* ************************************************************ */
                                                  /*            */
   IF RXVSAM("OPEN FILE('TEMPDD') COPYBOOK('CBNAME') SEQ")¬=0 THEN DO
      SAY 'RC='VSAMCODE  'MSG='VSAMMSG            /* report the error   */
      CALL CLEANUP                                /* cleanup and return */
   END                                            /*endif open¬=0       */
                                                  /*            */
   DO WHILE RXVSAM("READNEXT FILE('TEMPDD')")=0   /* read the file */
      SAY 'DATABASE='CB_DATABASE_NAME,            /*            */
          'TSNAME='CB_TS_NAME,                    /*            */
          'NAME='CB_NAME_NAME                     /*            */
   END                                            /* end do while */
                                                  /*            */
   CALL "RXVSAM" "CLOSE FILE('TEMPDD')"           /* close the file */
                                                  /*            */
                                                  /* Clean up and exit */
                                                  /*            */
 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
                                                                */
 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
                                                  /*            */
                                                  /*            */
   IF ADDRESS()='TSO' THEN                        /* if tso - then  */
      "FREE  FI('TEMPDD')"                        /* free temp file */
                                                  /*            */
   IF SQLCODE¬=0 THEN DO                          /*            */
      SAY 'RXSQL failed RC='SQLCODE               /*            */
      SAY SQLMSG                                  /*            */
   END                                            /*            */
/* ************************************************************ */
/* Disconnect from DB2                                          */
/* ************************************************************ */
                                                  /*            */
   CALL "RXSQL" "DISCONNECT"                      /* Disconnect from DB2 */
                                                  /*            */
   EXIT 0                                         /*            */
                                                  /*            */
RETURN
```



```
DATABASE=DSNDB06    TSNAME=SYSSTATS    NAME=SYSTABSTATS
DATABASE=DSNDB06    TSNAME=SYSSTR      NAME=SYSSTRINGS
DATABASE=DSNDB06    TSNAME=SYSUSER     NAME=SYSUSERAUTH
DATABASE=DSNDB06    TSNAME=SYSVIEWS    NAME=SYSVIEWDEP
DATABASE=DSNDB06    TSNAME=SYSVIEWS    NAME=SYSVIEWS
```

MAX/REXX Uses

Use for System Reporting

MAX/REXX can be used to quickly provide reports on system information such as RACF data.

Use a Copybook to define RACF data

```
*  RACF RECORD TYPE 100
        01  RACF-0100.
            05  RECORD-TYPE-0100              PIC X(4).
            05  FILLER                        PIC X(1).
            05  GPBD-NAME                     PIC X(8).
            05  FILLER                        PIC X(1).
            05  GPBD-SUPGRP-ID                PIC X(8).
            05  FILLER                        PIC X(1).
            05  GPBD-CREATE-DATE              PIC X(10).
            05  FILLER                        PIC X(1).
            05  GPBD-OWNER-ID                 PIC X(8).
            05  FILLER                        PIC X(1).
            05  GPBD-UACC                     PIC X(8).
            05  FILLER                        PIC X(1).
            05  GPBD-NOTERMUACC               PIC X(4).
            05  FILLER                        PIC X(1).
            05  GPBD-INSTALL-DATA             PIC X(254).
            05  FILLER                        PIC X(1).
            05  GPBD-MODEL                    PIC X(44).
```

32 of 42

MAX/REXX Uses

Sample MAX/REXX Program

```
/* REXX */
/* DOC: calculate new base amount for all COLORADO employees        */

  IF ADDRESS()="TSO" THEN "ALLOC FI(SYSUT1) DA('MXS.TEST.RACFBU') SHR"
                                                       /*            */
/* OPEN the file using a copybook and the field names can be used    */
/*       to access the data in each field of the record              */
                                                       /*            */
  IF 'RXVSAM'("OPEN FILE(SYSUT1) COPYBOOK(MXS.TEST.MAXBDS(RACF100))",
              " SEQ")<>0 THEN DO
     SAY 'RC='VSAMCODE 'MSG='VSAMMSG
     CALL CLEANUP
  END
                                                       /*            */
/* use MAX/REXX to list all group information on the RACF database   */
                                                       /*            */
  DO WHILE 'RXVSAM'('READNEXT FILE(SYSUT1) WHERE(1,EQ,C'0100')")=0
    DATA_LINE='GROUP NAME='GPBD_NAME                   /* group name) */
    SAY DATA_LINE
    DATA_LINE='SUBGROUP_ID='GPBD_SUPGRP_ID             /* sub id      */
    DATA_LINE=DATA_LINE||' CREATE_DATE='GPBD_CREATE_DATE /* date      */
    DATA_LINE=DATA_LINE||' OWNER_ID='GPBD_OWNER_ID         /* owner id  */
    SAY DATA_LINE
  END
 CALL CLEANUP
END
CLEANUP:
   CALL 'RXVSA
   IF ADDRESS(
   EXIT 0
```

```
SUBGROUP-ID=GPBD_SUPGRP_ID  CREATE_DATE=1995-06-14  OWNER_ID=E390   GROUP NAME=DCBGRP
SUBGROUP-ID=GPBD_SUPGRP_ID  CREATE_DATE=1995-10-30  OWNER_ID=E390   GROUP NAME=GLOBAL
SUBGROUP-ID=GPBD_SUPGRP_ID  CREATE_DATE=1995-06-14  OWNER_ID=E390
```

33 of 42

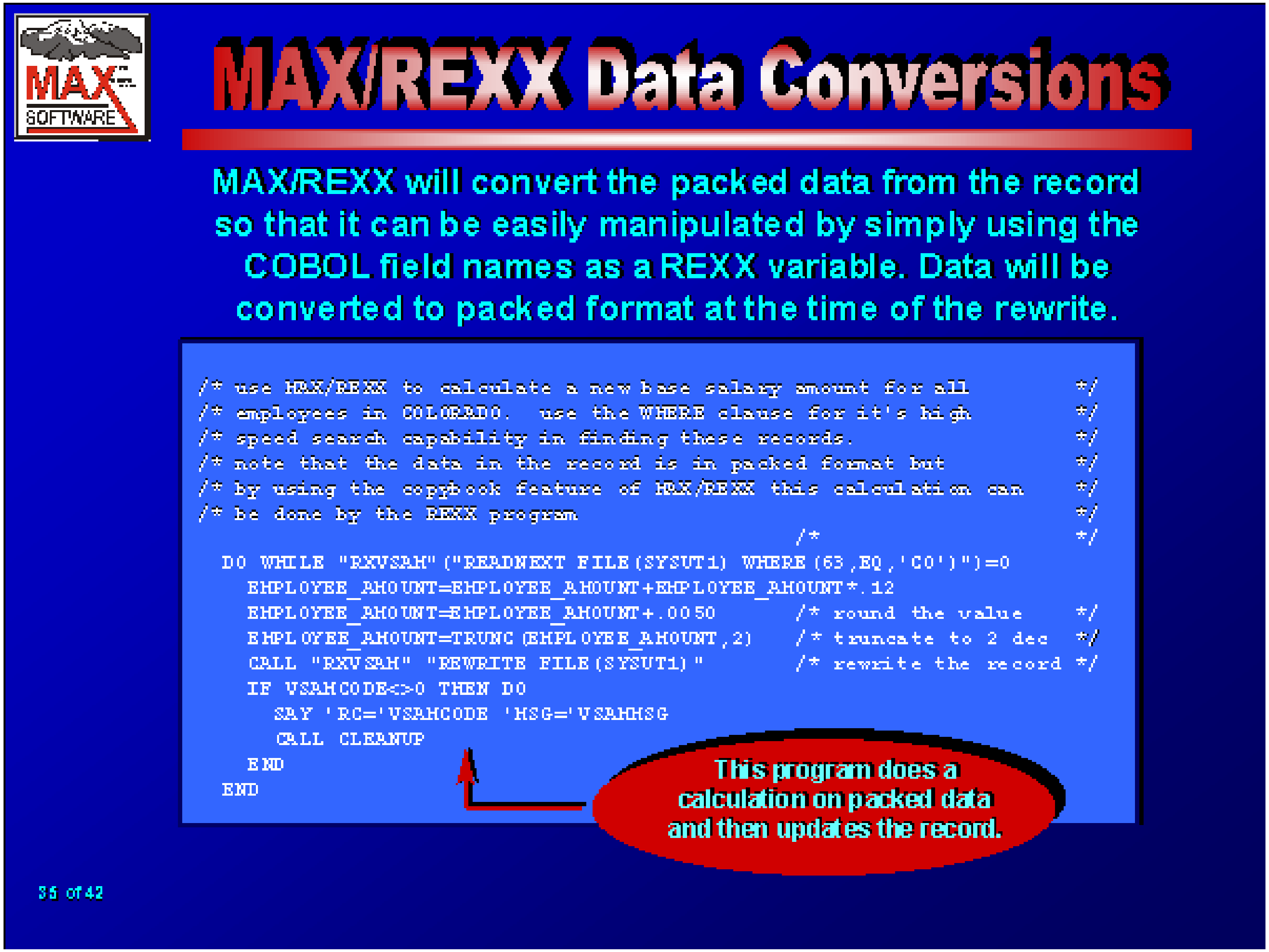
MAX/REXX Using COPYBOOK

MAX/REXX makes it easier to work with packed numeric data which is difficult to handle in classic REXX.

At OPEN time, data format is associated to a Copybook.

```
/* REXX */
/* DOC: calculate new base amount  for all COLORADO employees        */

  IF ADDRESS()="TSO" THEN "ALLOC FI(SYSUT1) DA('MXS.TEST.KSDS') SHR"
                                                    /*               */
/* OPEN the file using a copybook and the field names can be used    */
/*      to access the data in each field of the record               */
                                                    /*               */
  IF "RXVSAM"("OPEN FILE(SYSUT1) COPYBOOK(MXS.P390.COPYLIB(CBHDR))",
               " SEQUPD")<>0 THEN DO
     SAY 'RC='VSAMCODE 'MSG='VSAMMSG
     CALL CLEANUP
  END
```

This is a sample of RXVSAM which shows opening a file using a COPYBOOK. This allows for access of the data by using the copybook field names.



MAX/REXX Data Conversions

MAX/REXX will convert the packed data from the record so that it can be easily manipulated by simply using the COBOL field names as a REXX variable. Data will be converted to packed format at the time of the rewrite.

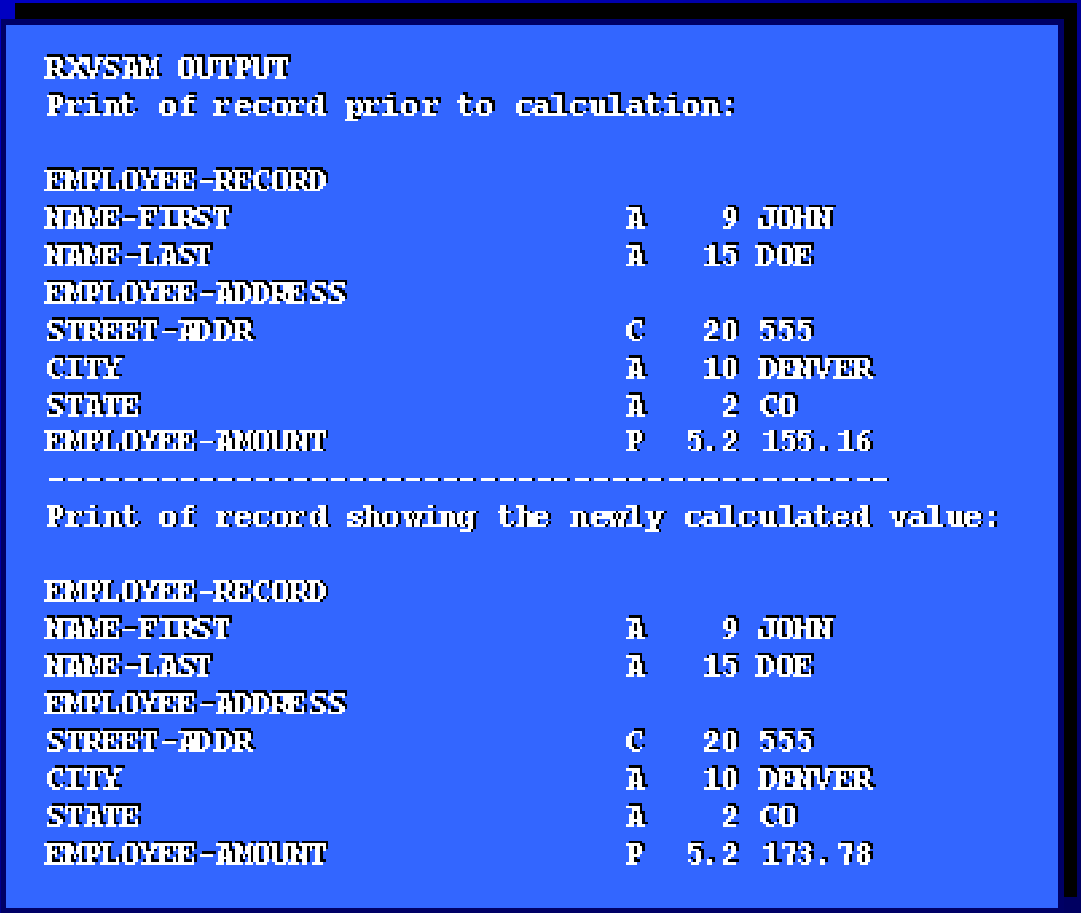
```
/* use MAX/REXX to calculate a new base salary amount for all          */
/* employees in COLORADO.  use the WHERE clause for it's high          */
/* speed search capability in finding these records.                   */
/* note that the data in the record is in packed format but            */
/* by using the copybook feature of MAX/REXX this calculation can      */
/* be done by the REXX program                                         */
                                                  /*                   */
  DO WHILE "RXVSAM"("READNEXT FILE(SYSUT1) WHERE(63,EQ,'CO')")=0
    EMPLOYEE_AMOUNT=EMPLOYEE_AMOUNT+EMPLOYEE_AMOUNT*.12
    EMPLOYEE_AMOUNT=EMPLOYEE_AMOUNT+.0050         /* round the value   */
    EMPLOYEE_AMOUNT=TRUNC(EMPLOYEE_AMOUNT,2)      /* truncate to 2 dec */
    CALL "RXVSAM" "REWRITE FILE(SYSUT1)"          /* rewrite the record */
    IF VSAMCODE<>0 THEN DO
      SAY 'RC='VSAMCODE 'MSG='VSAMMSG
      CALL CLEANUP
    END
  END
```

This program does a calculation on packed data and then updates the record.

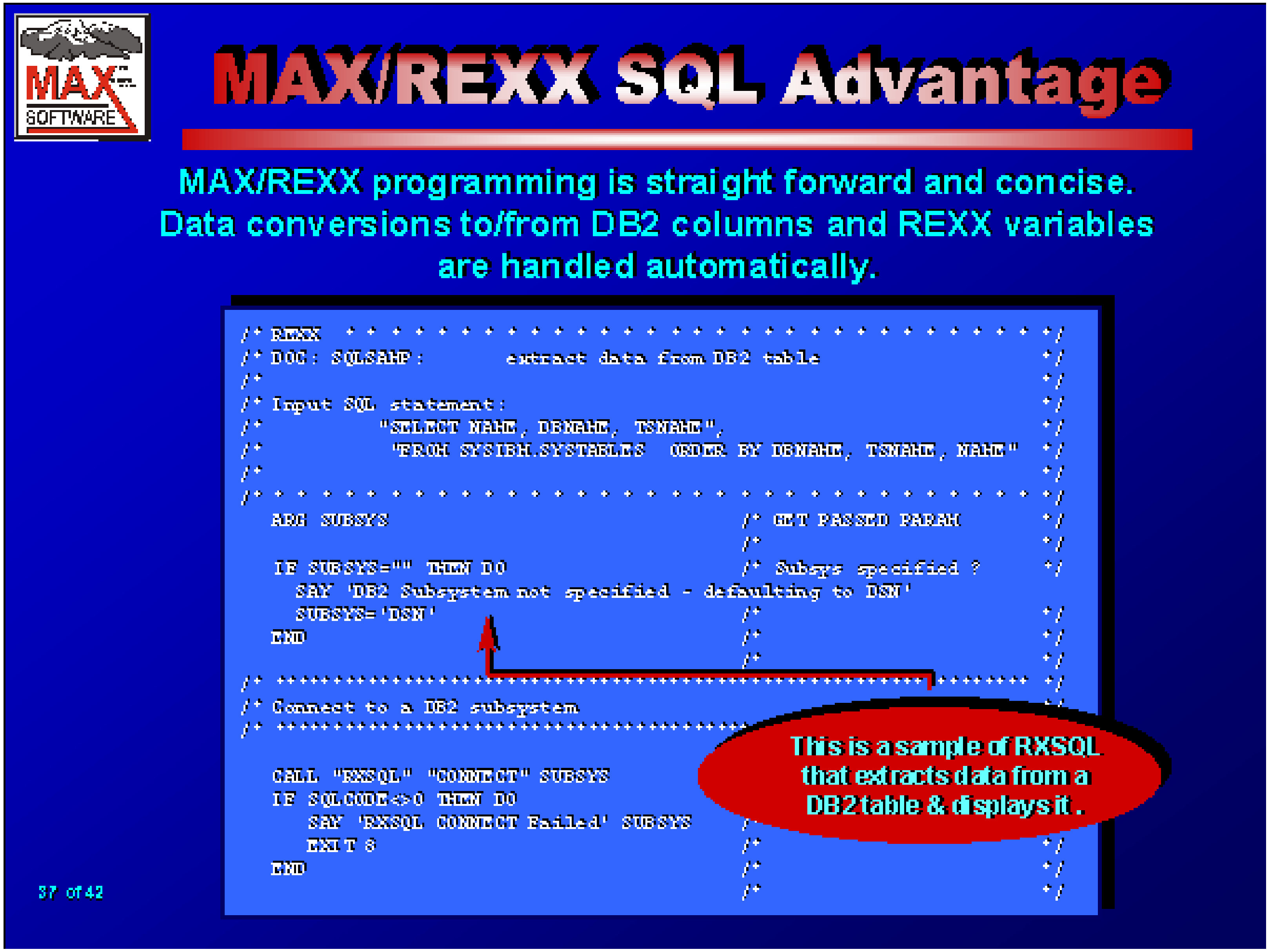
35 of 42

MAX/REXX Output

This is a sample record before & after the EMPLOYEE-AMOUNT has been recalculated.



```
RXVSAM OUTPUT
Print of record prior to calculation:

EMPLOYEE-RECORD
NAME-FIRST                     A     9 JOHN
NAME-LAST                      A    15 DOE
EMPLOYEE-ADDRESS
STREET-ADDR                    C    20 555
CITY                           A    10 DENVER
STATE                          A     2 CO
EMPLOYEE-AMOUNT                P   5.2 155.16
---------------------------------------------
Print of record showing the newly calculated value:

EMPLOYEE-RECORD
NAME-FIRST                     A     9 JOHN
NAME-LAST                      A    15 DOE
EMPLOYEE-ADDRESS
STREET-ADDR                    C    20 555
CITY                           A    10 DENVER
STATE                          A     2 CO
EMPLOYEE-AMOUNT                P   5.2 173.78
```



MAX/REXX SQL Advantage

MAX/REXX programming is straight forward and concise. Data conversions to/from DB2 columns and REXX variables are handled automatically.

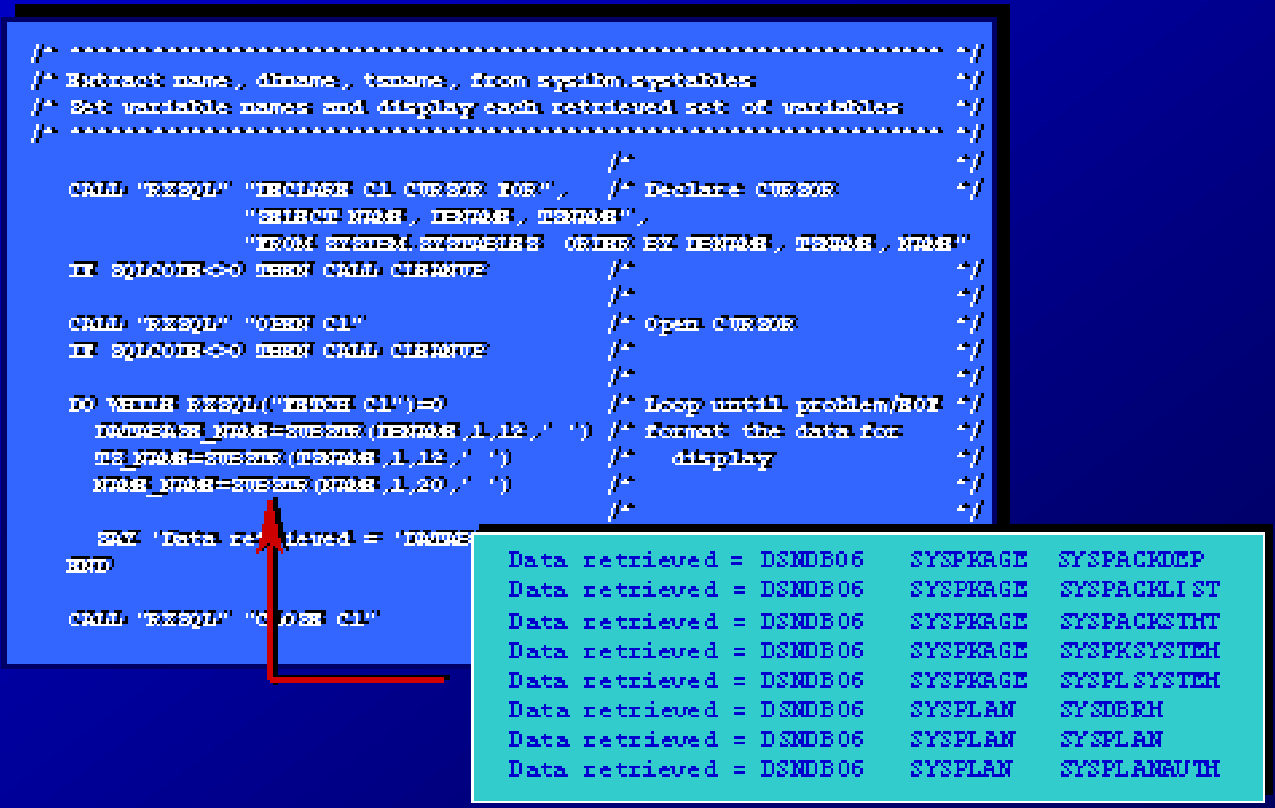
```
/* REXX  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
/* DOC : SQLSAMP :        extract data from DB2 table                    */
/*                                                                       */
/* Input SQL statement :                                                 */
/*           "SELECT NAME , DBNAME,  TSNAME",                            */
/*            "FROM SYSIBM.SYSTABLES  ORDER BY DBNAME,  TSNAME , NAME"   */
/*                                                                       */
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
   ARG  SUBSYS                                   /* GET PASSED PARAM     */
                                                 /*                      */
   IF SUBSYS=""  THEN DO                         /* Subsys specified ?   */
     SAY  'DB2 Subsystem not  specified - defaulting to DSN'
     SUBSYS= 'DSN'                               /*                      */
   END                                           /*                      */
                                                 /*                      */
/* *************************************************************        */
/* Connect to a DB2 subsystem
/* ***********************************

   CALL "RXSQL" "CONNECT" SUBSYS
   IF SQLCODE<>0  THEN  DO
      SAY 'RXSQL  CONNECT Failed' SUBSYS         /
      EXIT 8                                     /*                      */
   END                                           /*                      */
                                                 /*                      */
```

This is a sample of RXSQL that extracts data from a DB2 table & displays it .

37 of 42

MAX/REXX SQL Advantage

A simple loop with one RX/SQL statement can be used to display all the table information.



```
/* ******************************************************************* */
/* Extract name, dbname, tsname, from sysibm.systables                 */
/* Set variable names and display each retrieved set of variables      */
/* ******************************************************************* */
                                                  /*                   */
   CALL "RXSQL" "DECLARE C1 CURSOR FOR",          /* Declare CURSOR    */
                "SELECT NAME, DBNAME, TSNAME",
                "FROM SYSIBM.SYSTABLES  ORDER BY DBNAME, TSNAME, NAME"
   IF SQLCODE<>0 THEN CALL CLEANUP                /*                   */
                                                  /*                   */
   CALL "RXSQL" "OPEN C1"                         /* Open CURSOR       */
   IF SQLCODE<>0 THEN CALL CLEANUP                /*                   */
                                                  /*                   */
   DO WHILE RXSQL("FETCH C1")=0                   /* Loop until problem/EOF */
     DATABASE_NAME=SUBSTR(DBNAME,1,12,' ')        /* format the data for */
     TS_NAME=SUBSTR(TSNAME,1,12,' ')              /*   display         */
     NAME_NAME=SUBSTR(NAME,1,20,' ')              /*                   */
                                                  /*                   */
     SAY 'Data retrieved = 'DATAB
   END

   CALL "RXSQL" "CLOSE C1"
```

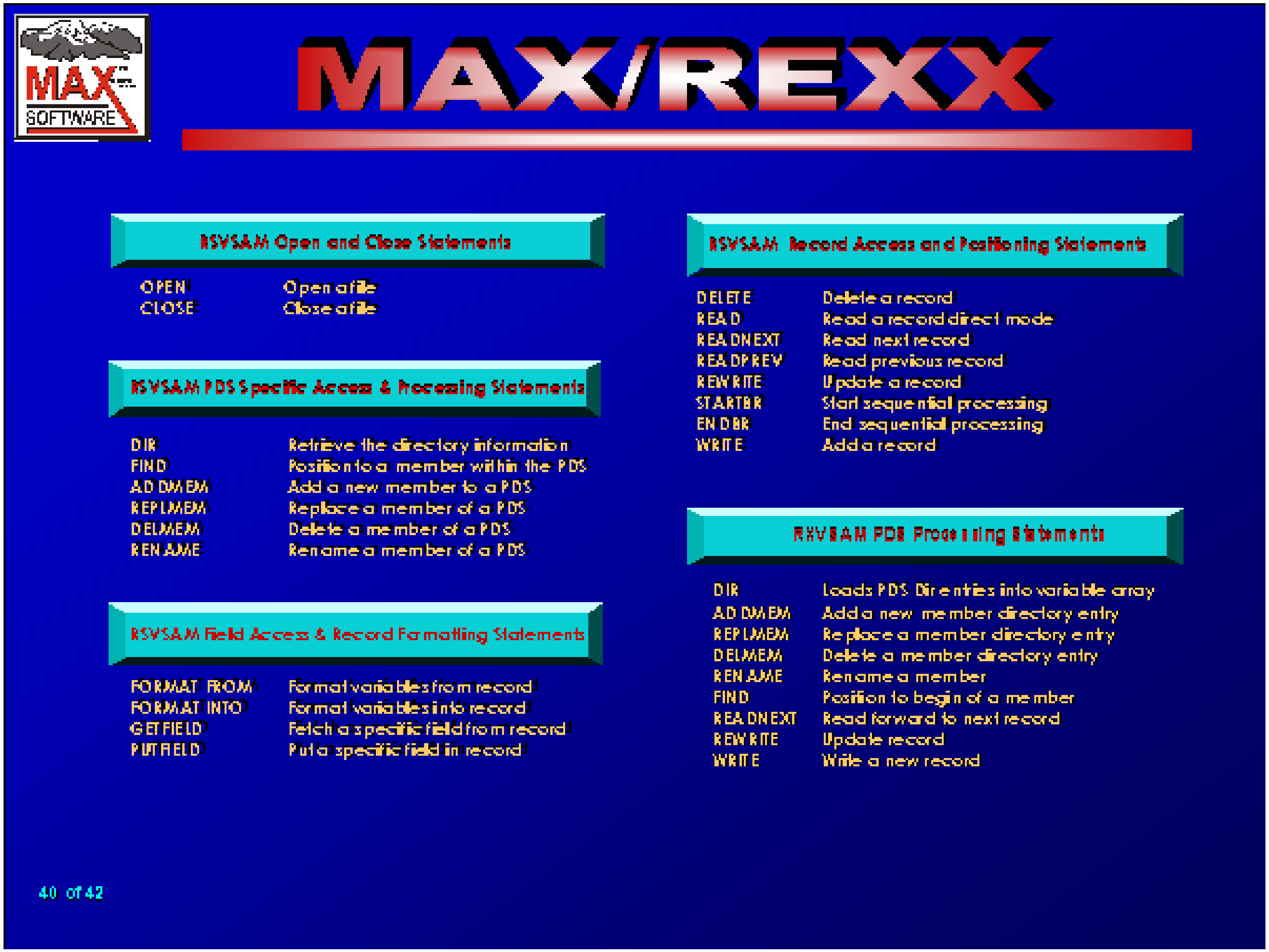
```
Data retrieved = DSNDB06    SYSPKAGE    SYSPACKDEP
Data retrieved = DSNDB06    SYSPKAGE    SYSPACKLIST
Data retrieved = DSNDB06    SYSPKAGE    SYSPACKSTMT
Data retrieved = DSNDB06    SYSPKAGE    SYSPKSYSTEM
Data retrieved = DSNDB06    SYSPKAGE    SYSPLSYSTEM
Data retrieved = DSNDB06    SYSPLAN     SYSDBRM
Data retrieved = DSNDB06    SYSPLAN     SYSPLAN
Data retrieved = DSNDB06    SYSPLAN     SYSPLANAUTH
```

38 of 42

MAX/REXX I-Compiler

- MAX/REXX compiles REXX source programs into executable object modules.
- The compiled programs may be executed directly from JCL, called from a program, or invoked as a TSO command procedure.
- The compiler provides the same security and change control as other languages such as COBOL or PL1.
- An optional, optimizing compiler is available for even greater performance.

39 of 42



MAX/REXX

RSVSAM Open and Close Statements

OPEN	Open a file
CLOSE	Close a file

RSVSAM PDS Specific Access & Processing Statements

DIR	Retrieve the directory information
FIND	Position to a member within the PDS
ADDMEM	Add a new member to a PDS
REPLMEM	Replace a member of a PDS
DELMEM	Delete a member of a PDS
RENAME	Rename a member of a PDS

RSVSAM Field Access & Record Formatting Statements

FORMAT FROM	Format variables from record
FORMAT INTO	Format variables into record
GETFIELD	Fetch a specific field from record
PUTFIELD	Put a specific field in record

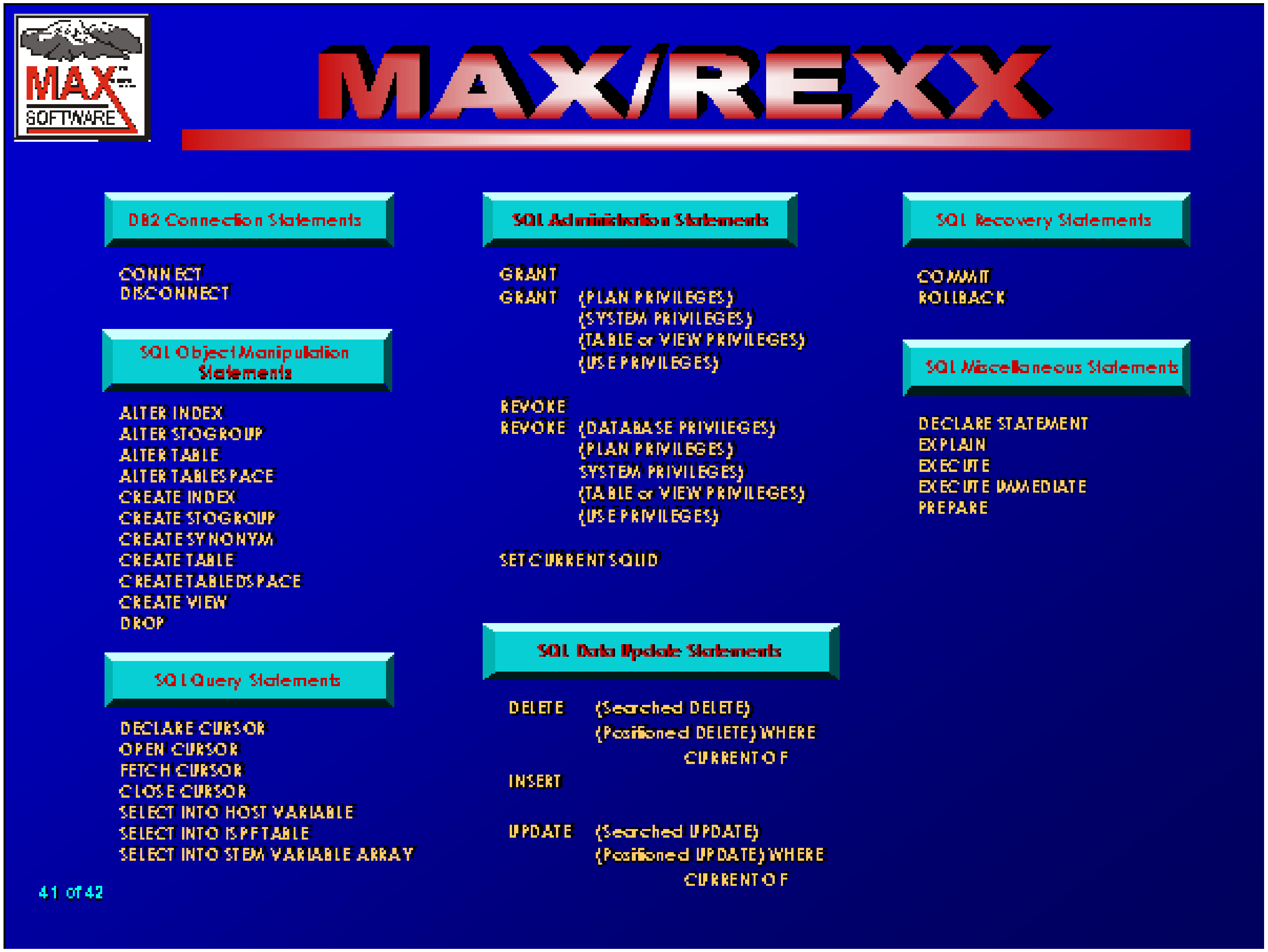
RSVSAM Record Access and Positioning Statements

DELETE	Delete a record
READ	Read a record direct mode
READNEXT	Read next record
READPREV	Read previous record
REWRITE	Update a record
STARTBR	Start sequential processing
ENDBR	End sequential processing
WRITE	Add a record

RXVSAM PDS Processing Statements

DIR	Loads PDS Dir entries into variable array
ADDMEM	Add a new member directory entry
REPLMEM	Replace a member directory entry
DELMEM	Delete a member directory entry
RENAME	Rename a member
FIND	Position to begin of a member
READNEXT	Read forward to next record
REWRITE	Update record
WRITE	Write a new record

40 of 42



MAX/REXX

DB2 Connection Statements

CONNECT
DISCONNECT

SQL Object Manipulation Statements

ALTER INDEX
ALTER STOGROUP
ALTER TABLE
ALTER TABLESPACE
CREATE INDEX
CREATE STOGROUP
CREATE SYNONYM
CREATE TABLE
CREATE TABLEDSPACE
CREATE VIEW
DROP

SQL Query Statements

DECLARE CURSOR
OPEN CURSOR
FETCH CURSOR
CLOSE CURSOR
SELECT INTO HOST VARIABLE
SELECT INTO ISPF TABLE
SELECT INTO STEM VARIABLE ARRAY

SQL Administration Statements

GRANT
GRANT (PLAN PRIVILEGES)
(SYSTEM PRIVILEGES)
(TABLE or VIEW PRIVILEGES)
(USE PRIVILEGES)

REVOKE
REVOKE (DATABASE PRIVILEGES)
(PLAN PRIVILEGES)
SYSTEM PRIVILEGES)
(TABLE or VIEW PRIVILEGES)
(USE PRIVILEGES)

SET CURRENT SQLID

SQL Data Update Statements

DELETE (Searched DELETE)
(Positioned DELETE) WHERE
CURRENT OF

INSERT

UPDATE (Searched UPDATE)
(Positioned UPDATE) WHERE
CURRENT OF

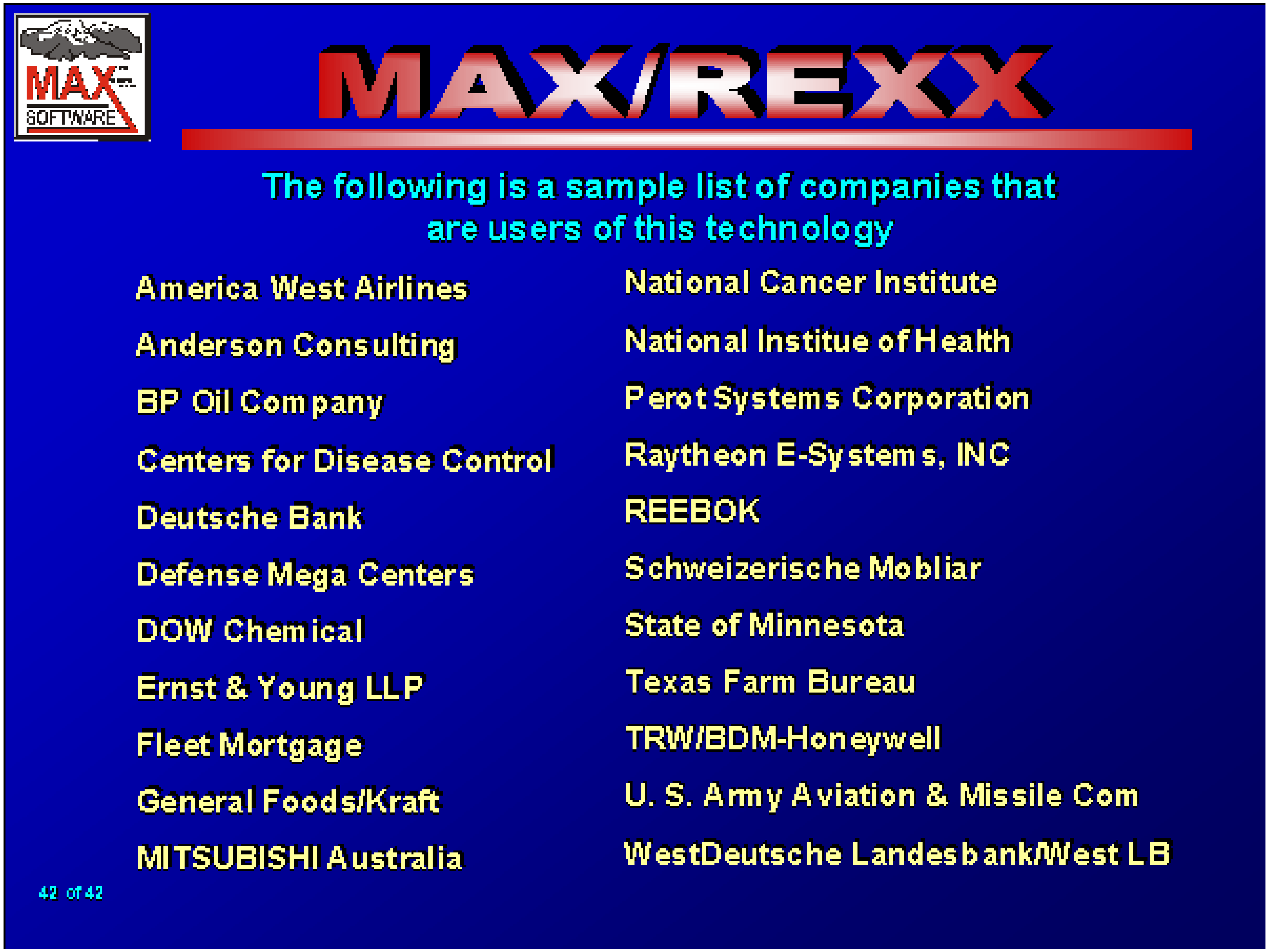
SQL Recovery Statements

COMMIT
ROLLBACK

SQL Miscellaneous Statements

DECLARE STATEMENT
EXPLAIN
EXECUTE
EXECUTE IMMEDIATE
PREPARE

41 of 42



MAX/REXX

The following is a sample list of companies that are users of this technology

America West Airlines	National Cancer Institute
Anderson Consulting	National Institue of Health
BP Oil Company	Perot Systems Corporation
Centers for Disease Control	Raytheon E-Systems, INC
Deutsche Bank	REEBOK
Defense Mega Centers	Schweizerische Mobliar
DOW Chemical	State of Minnesota
Ernst & Young LLP	Texas Farm Bureau
Fleet Mortgage	TRW/BDM-Honeywell
General Foods/Kraft	U. S. Army Aviation & Missile Com
MITSUBISHI Australia	WestDeutsche Landesbank/West LB

42 of 42