IBM DB2 for i: Code example

After Update Trigger

```
*=========================================================*
 * This program is intended to illustrate an after update *
 * trigger that simulates an update cascade.              *
 *                                                        *
 *                                                        *
 * CRTSQLRPGI OBJ(CORPDATA/TRG03) SRCFILE(MJASRC/RPG)     *
 *            COMMIT(*NONE) OUTPUT(*PRINT)                *
 *            OPTION(*XREF *NOGEN) DBGVIEW(*SOURCE)       *
 *            USRPRF(*OWNER) DYNUSRPRF(*OWNER)            *
 *                                                        *
 * CRTBNDRPG PGM(CORPDATA/TRG03) SRCFILE(QTEMP/QSQLTEMP1) *
 *           DFTACTGRP(*NO) ACTGRP(*CALLER)               *
 *           DBGVIEW(*SOURCE) ALWNULL(*YES)               *
 *           USRPRF(*OWNER)                               *
 *                                                        *
 * ADDPFTRG FILE(CORPDATA/DEPARTMENT)                     *
 *          TRGTIME(*AFTER) TRGEVENT(*UPDATE)             *
 *          PGM(CORPDATA/TRG03) ALWREPCHG(*YES)           *
 *                                                        *
 *                                                        *
 *=========================================================*
 *
 *=========================================================*
 * Definition of the structure passed as the first        *
 * parameter from database to the trigger program.        *
 * The include is used so that any additional fields      *
 * in the interface template in the future will be        *
 * brought into the program if it is recompiled.          *
 *                                                        *
 * The includes in the QSYSINC library must be on         *
 * your system to compile this program. Option 13 of      *
 * the OS/400 install will install them.                  *
 *                                                        *
 *=========================================================*
D/COPY QSYSINC/QRPGLESRC,TRGBUF
 *
 *=========================================================*
 * This is an overlay used to set addressability          *
 * to the various sections of the interface buffer        *
 * such as the before and after record images.            *
 *=========================================================*
D INTARR          S            1A   BASED(INTPTR) DIM(32767)
D INTPTR          S             *
 *
 *=========================================================*
 * Definition of the trigger buffer length passed as      *
 * the second parameter from database to the trigger      *
 * program.                                               *
 *=========================================================*
D PARM2           DS
D  LENG                  1      4B 0
 *
 *=========================================================*
 * These pointers are used to point to the before         *
 * and after images. The before and after images          *
 * are passed in the first parameter structure.           *
 *=========================================================*
D BIMAGE          S               *
D AIMAGE          S               *
 *
```

```
 *============================================================*
 * These based structures provide the subfields of           *
 * the record images. Externally defined data                *
 * structures are used so a recompile of the                 *
 * program will always pick up the latest field              *
 * defintions.                                               *
 *============================================================*
D BEMP          E DS                  EXTNAME(DEPARTMENT)
D                                     BASED(BIMAGE)
D                                     PREFIX(B)
D AEMP          E DS                  EXTNAME(DEPARTMENT)
D                                     BASED(AIMAGE)
D                                     PREFIX(A)
 *
 *============================================================*
 * Error output from QMHSNDPM                                *
 *============================================================*
D/COPY QSYSINC/QRPGLESRC,QUSEC
D  RTNDTA                17     56
 *
 *============================================================*
 * Parameters for QMHSNDPM                                   *
 *============================================================*
D FLDS            DS
D  MSGLEN                 1      4B 0
D  PGMSTK                 5      8B 0
D  RTVLEN                 9     12B 0
D  MSGQLEN               13     16B 0
D  PGMWTT                17     20B 0
D  LASTOPEN              21     21A
 *
 *============================================================*
 * If an error occurs, the trigger program should send an    *
 * error to database to indicate that the operation has      *
 * failed. To send an escape message, use the QMHSNDPM       *
 * API. We must send the message to the call stack entry     *
 * that comes immediately before the trigger program.        *
 * The first call stack entry in an ILE RPG trigger          *
 * program is the PEP. Below, we have defined the            *
 * name of the PEP for this ILE trigger program.             *
 *============================================================*
D MSGQNAM         C                   CONST('_QRNP_PEP_TRG02')
 *
 *============================================================*
 * Place the name of your message file in the first 10       *
 * characters, padding with blanks to the 10th character.    *
 * Place the library name in the second 10 characters        *
 * padding with blanks.                                      *
 *============================================================*
D MSGFNAME        C                CONST('MSGF      MJATST ')
D LEN             C                CONST(15)
D MODNAME         C                CONST('*NONE     *NONE  ')
 *
C     *ENTRY        PLIST
C     QDBTB         PARM                    QDBTB
C     PARM2         PARM                    PARM2
 *
 *============================================================*
 * This is the parameter list of QMHSNDPM.                   *
 *============================================================*
C     PLIST1        PLIST
C                   PARM                    MSGID            7
C                   PARM                    MSGF            20
C                   PARM                    MSGDTA          25
C                   PARM                    MSGLEN
C                   PARM                    MSGTYP          10
C                   PARM                    MSGQUE          19
C                   PARM                    PGMSTK
C                   PARM                    MSGKEY           4
C                   PARM                    QUSEC
C                   PARM                    MSGQLEN
C                   PARM                    CSEQUAL         20
C                   PARM                    PGMWTT
```

```
 *=============================================================*
 * Set the basing pointers for the interface                   *
 * structure and the before and after images                   *
 *=============================================================*
C                EVAL      INTPTR = %ADDR(QDBTB)
C                EVAL      BIMAGE = %ADDR(INTARR(QDBORO+1))
C                EVAL      AIMAGE = %ADDR(INTARR(QDBNRO+1))
 *
 *=============================================================*
 * Only open or perform an update if the manager               *
 * number has changed.                                         *
 *=============================================================*
C     AMGRNO        IFNE      BMGRNO
 *
 *=============================================================*
 * Update the employee record. Note that SQL will keep         *
 * any of the used ODPs open.                                  *
 * Also SQL is able to use a specific isolation level          *
 * either by using the SET TRANSACTION statement or            *
 * by statement level isolation levels.                        *
 *=============================================================*
C     QDBCLL        IFEQ      '0'
C/EXEC SQL UPDATE CORPDATA/EMPLOYEE SET WORKDEPT = :ADEPTNO
C+         WHERE EMPNO = :AMGRNO WITH NC
C/END-EXEC
C                   ELSE
C     QDBCLL        IFEQ      '1'
C/EXEC SQL UPDATE CORPDATA/EMPLOYEE SET WORKDEPT = :ADEPTNO
C+         WHERE EMPNO = :AMGRNO WITH UR
C/END-EXEC
C                   ELSE
C     QDBCLL        IFEQ      '2'
C/EXEC SQL UPDATE CORPDATA/EMPLOYEE SET WORKDEPT = :ADEPTNO
C+         WHERE EMPNO = :AMGRNO WITH CS
C/END-EXEC
C                   ELSE
C     QDBCLL        IFEQ      '3'
C/EXEC SQL UPDATE CORPDATA/EMPLOYEE SET WORKDEPT = :ADEPTNO
C+         WHERE EMPNO = :AMGRNO WITH RS
C/END-EXEC
C                   ELSE
C/EXEC SQL UPDATE CORPDATA/EMPLOYEE SET WORKDEPT = :ADEPTNO
C+         WHERE EMPNO = :AMGRNO WITH RR
C/END-EXEC
C                   ENDIF
C                   ENDIF
C                   ENDIF
C                   ENDIF
 *
 *=============================================================*
 * If the SQL operation is unsuccessful for                    *
 * some reason return a message to the application to          *
 * make the update fail.                                       *
 *=============================================================*
C     SQLCOD        IFLT      0
 *
 *=============================================================*
 * Place the appropriate message ID here.                      *
 *=============================================================*
C                   MOVEL     'TRG0001'     MSGID
C                   MOVEL     MSGFNAME      MSGF
C                   MOVE      '  '          MSGDTA
C                   Z-ADD     25            MSGLEN
C                   MOVEL(P)  '*ESCAPE'     MSGTYP
C                   MOVEL(P)  MSGQNAM       MSGQUE
C                   MOVEL(P)  MODNAME       CSEQUAL
C                   MOVE      '  '          MSGDTA
C                   Z-ADD     1             PGMSTK
C                   Z-ADD     LEN           MSGQLEN
C                   MOVE      ' '           MSGKEY
C                   Z-ADD     66            QUSBPRV
C                   Z-ADD     0             QUSBAVL
C                   MOVE      ' '           QUSEI
```

```
C                   MOVE      ' '           QUSERVED
C                   MOVE      ' '           RTNDTA
C                   CALL      'QMHSNDPM'    PLIST1
C                   ENDIF
C                   ENDIF
 *
C                   RETURN
```
