

IBM DB2 for i: Code example

After Update Trigger (without null values)

```

*=====*
* This program is intended to illustrate an after update *
* trigger that simulates an update cascade. *
* *
* This program will only work on files that do not *
* contain null values. If a field contains null values, *
* SQL must be used to update the file. *
* *
* CRTBNDRPG PGM(CORPDATA/TRG02) SRCFILE(MJASRC/RPG) *
*          DFTACTGRP(*NO) ACTGRP(*CALLER) *
*          DBGVIEW(*SOURCE) ALWNULL(*YES) *
*          USRPRF(*OWNER) *
* *
* ADPPFTRG FILE(CORPDATA/DEPARTMENT) *
*          TRGTIME(*BEFORE) TRGEVENT(*UPDATE) *
*          PGM(CORPDATA/TRG02) ALWREPCBG(*YES) *
* *
*=====*
*
*=====*
* File definition used to update the record. *
*=====*
      FEMPLOYEE  UF   E           K DISK      RENAME(EMPLOYEE:EMPFMT)
      F                               USROPN
      F                               COMMIT(LCKLVL)
*
*=====*
* 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          DS          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        *
* definitions.                                         *
*=====*
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 RTNDDTA          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(QDBOR0+1))
C          EVAL          AIMAGE = %ADDR(INTARR(QDBNR0+1))
*
*=====*
* Only open or perform an update if the manager *
* number has changed. *
*=====*
C          AMGRNO          IFNE          BMGRNO
*
*=====*
* Open the file. We need to open the file with or *
* without commit and we dont want to have to open *
* and close it too often. Note that ILE RPG can only*
* open with or without commit and cannot specify *
* a specific isolation level. SQL is necessary to *
* control the isolation precisely. *
*=====*
*
C          QDBCLL          IFGT          '0'
C          MOVE          '1'          LCKLVL
C          ELSE
C          MOVE          '0'          LCKLVL
C          ENDIF
C          LCKLVL          IFNE          LASTOPEN
C          LASTOPEN          IFNE          ' '
C          CLOSE          EMPLOYEE
C          ENDIF
C          OPEN          EMPLOYEE          60
C          N60          MOVE          LCKLVL          LASTOPEN
C          ENDIF
*
*=====*
* Get the same record for update (this assumes EMPNO *
* is unique). Change the WORKDEPT field for the *
* new manager. *
*=====*
C          AMGRNO          CHAIN          EMPFMT          6060
C          N60          MOVE          ADEPTNO          WORKDEPT
C          N60          UPDATE          EMPFMT          60
*
*=====*
* If the CHAIN or UPDATE operation is unsuccessful for *
* some reason send a message to database to make the *
* update fail. *
*=====*
C          *IN60          IFEQ *ON
*
*=====*
* Place the appropriate message ID here. *
*=====*
C          MOVE          'TRG0001'          MSGID
C          MOVE          MSGFNAME          MSGF
C          MOVE          ' '          MSGDTA
C          Z-ADD          25          MSGLEN
C          MOVE          (P)          '*ESCAPE'          MSGTYP
C          MOVE          (P)          MSGQNAM          MSGQUE
C          MOVE          (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       ' '        RTNDDTA
C          CALL       'QMHSNDPM'  PLIST1
C          ENDIF
C          ENDIF
C          *
C          RETURN
```
