Hi Charles,

By default, all Processor Steps run under the ALTID at least you specify ALTID=N kewword at the STEP definition. Although the step is executed under the ALT ID, the creation or deletion of files during the activation of the step is carried out under the security context (user ID) of the user who performs the action in Endevor.

During the execution of a program under Endevor control, you could switch the security context between user ID and Alternate ID using the allocation of special ddnames provided by Endevor: LGNT$$$I and LGNT$$$O. If you wanted to change the step processor to the Alternate ID security context, you would need to execute an open instruction on the LGNT$$$I ddname using the primitives provided by the programming language you used (Cobol: Open, Assembler: Open Macro , c/c++: fopen("dd:LGNT$$$I", "w"), etc.). If you want to return to the user's execution context environment, you must first close the LGNT$$$I ddname using the instruction provided by the programming language and then open the LGNT$$$O ddname following the same procedure described above for LGNT$$$I.

Broadly speaking, the effect caused by opening/closing these special ddnames is the change of the user ID in the field pointed to by ACEEUNAM in the ACEE control block. The address of the ACEE is obtained from the TCBSENV field in the TCB control block of the task.

The ddnames LGNT$$$I and LGNT$$$O could be pre-allocated in the Endevor processor step or allocated dynamically in the program using BPXWDYN  / TSO ALLOC command  /SVC 99 (depending on the language).

Things change a bit when you try to run the LGN$$$I/LGNT$$$0 logic described above under Db2, depending on whether DB2 security is under SAF control. If it is not, DB2 uses the ASXBUSER value of the Address Space Extension Block (ASXB) to determine the security context . If SAF is being used, the value of the user ID field in the ACEE control block pointed to by ASXBSENV field of ASXB is used. The LGN$$$I/LGNT$$$0 mechanism updates both fields

Below you will find a REXX template which uses the ddname logic, executing DB2 Command Processor (DSN SYSTEM) and intercepting the output generated by it. BuildBindCard Procedure is not included but the main purpose is to create the Bind Card Statement according to your standards/needs:

/\*REXX\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

parse upper arg DB2ssid

/\* "alloc FI(LGNT$$$I) DUMMY REUSE"

"alloc FI(LGNT$$$O) DUMMY REUSE" \*/

call BuildBindCard

queue strip(BindCard)

queue "END"

/\*

    Swap to Alternate User ID

\*/

"execio 0 diskw LGNT$$$I (open"

x=outtrap('BindOutput.',,noconcat)

"DSN SYSTEM("DB2ssid")"

db2\_rc = rc

x=outtrap("OFF")

Say ''

Say 'BINDPKG02I Output generated by the Bind Process:'

Say ''

Do i=1 to BindOutput.0

  Say '           'BindOutput.i

End

say ' '

/\*

    Swap to caller User ID

\*/

"execio 0 diskw LGNT$$$I (finis"

"execio 0 diskw LGNT$$$O (open"

"execio 0 diskw LGNT$$$O (finis"

Select

  When db2\_rc = 0 Then Do

    Say '\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'||,

        '\*\*\*\*\*\*\*\*\*\*\*\*'

    say ''

    say ' B I N D   O K.'

    say ''

    Say '\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'||,

        '\*\*\*\*\*\*\*\*\*\*\*\*'

    say ''

    Say 'BINDPKG01I DB2 Command Processor has executed',

        'the BIND Command succesfully.'

    say ''

  End /\*rc = 0\*/

  When db2\_rc > 4 Then Do

    Say '\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'||,

        '\*\*\*\*\*\*\*\*\*\*\*\*'

    say ''

    say ' B I N D   E R R O R'

    say ''

    Say '\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'||,

        '\*\*\*\*\*\*\*\*\*\*\*\*'

    say ''

    Say 'BINDPKG01E DB2 Command Processor has issued',

        'an error during BIND Command execution.'

    say ''

  End

  When db2\_rc = 4 Then Do

    Say '\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'||,

        '\*\*\*\*\*\*\*\*\*\*\*\*'

    say ''

    say ' B I N D   W A R N I N G'

    say ''

    Say '\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'||,

        '\*\*\*\*\*\*\*\*\*\*\*\*'

    say ''

  otherwise ;

End /\* Select \*/

Return db2\_rc