(ILE) Testing for Omitted Parameters

Tech tip courtesy of Dynax Solutions, Inc. and Jeff Young

Here are sample code snippets to illustrate how to check for and handle parameters that you expect in your code but may be omitted by the caller. The examples are written in RPGLE, but the APIs can be used in any ILE language.

Example of using the MONITOR op-code to "catch" an error. It is used here regarding a missing parameter, but MONITOR has other uses as well:

```
 * This example will use the MONITOR Op Code to ignore
 * any error when loading a parm field that was not passed.

 * Normally, if Parm 3 and Parm 4 are not passed to this
 * program, a program exception would cause the program
 * to end abnormally.

 * The MONITOR Op Code allows you to test for specific
 * errors, or by not having any tests, handle all errors.

 * The MONITOR Op Code is in effect for ALL instructions
 * between the MONITOR and the ENDMON Op Code.

 * Load return parms - test if passed

C                   Monitor
C                   Move      Output_Parm_3 Parm3
C                   On-Error
C                   EndMon

C                   Monitor
C                   Move      Output_Parm_4 Parm4
C                   On-Error
C                   EndMon

C                   Eval      *InLr = *On
C                   Return

 * Program initialization routine
C     *InzSr        BegSR
 * Entry parms
C     *Entry        Plist
C                   Parm                    Parm1
C                   Parm                    Parm2
C                   Parm                    Parm3
C                   Parm                    Parm4
C                   Parm                    Parm5
C                   EndSr
```

Here is another code fragment. In this one, the CEETSTA API is used to actually test if a parameter is omitted before trying to use it instead of catching the error:

```
 * The procedure CEETSTA will be used to test if a
 * parm was passed to this program
 ***********************************************
 * Define parms required to call procedure CEETSTA
 * Purpose        Test if parm was omitted
 * Output:
 *  Parm_Passed   Flag indicating whether parm omitted.
```

```
 *          1 = passed, 0 = omitted
 *  Feed_Back     Feedback Code (may be omitted).
 *          CEE0000 = Successful,
 *          CEE0503 = Parm Number not valid,
 *          CEE3005 = Parm number n

 *  Input:
 *  Parm_Number   Position of parm in calling list


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

D CEETSTA         PR                  ExtProc('CEETSTA')

 * Flag to indicate if parm was passed
D  Parm_Passed                  10I 0

 * Position of parm in parm list
D  Parm_Number                  10I 0 Const

 * Feedback Code (Not used)
D  Feed_Back                    12A   Options(*Omit)

 ***********************************************
 * Load return parms - test if passed
C                   Eval      Parm_Number = 2
C                   Exsr      Test_Parm
C                   If        Parm_Passed = 1
 * Your code here
C                   EndIf

C                   Eval      Parm_Number = 5
C                   Exsr      Test_Parm
C                   If        Parm_Passed = 1
 * Your code here
C                   EndIf

C                   Eval      *InLr = *On
C                   Return

 * This subroutine will test if an input parm was passed
C     Test_Parm     Begsr
C                   Callp     CEETSTA ( Parm_Passed :
C                                       Parm_Number : *OMIT )
C                   EndSr

 * Program initialization routine
C     *InzSr        BegSR

 * Entry parms
C     *Entry        Plist
C                   Parm                    Parm1
C                   Parm                    Parm2
C                   Parm                    Parm3
C                   Parm                    Parm4
C                   Parm                    Parm5

C                   EndSr
```

That was easy, wasn't it?