How to Install User-Written CALC Database Procedures

For Use with the Single-Thread DMS Interface of I-QU PLUS-1 and for CALC Simulation (CALSIM)

This procedure is provided for those customers who have local user-written CALC database procedures (DBPs), and wish to collect them with I-QU PLUS-1 and also perform the CALC simulation using them. I-QU PLUS-1 already provides simulation for the Unisys supplied DMSCALC and RANDENTIAL routines.

There are two issues at stake:

1) The inclusion of user-written DBPs (any type - "Old CALC" or standard database procedures) in the ***single-thread*** DMR included in I-QU PLUS-1.

   Note: When using standard database procedures, nothing special needs to be done for the multi-thread interface as I-QU PLUS-1 simply calls the DMR like any other run-unit; however, when using "Old CALC" database procedures, a CALSEG relocatable must be created and collected with I-QU PLUS-1 (see the paper titled, **"How to Install an 'Old CALC' Routine in I-QU PLUS-1 and Q-LINK"**, from KMSystems).

2) The inclusion of the user-written "Old CALC" DBPs into the code for the "DEFINE C" directive and the "CALSIM" command.

The reason that there are two different issues is that CALSIM is not a DMS command. Therefore, the user-written CALC DBPs will be collected twice in I-QU PLUS-1: once with the single-thread code and again with CALSIM.

The modifications to CALSIM discussed here are valid only for "Old CALC" database procedures.

Refer to the Unisys UDS Data Management System Administration, Operations and Support Guide for a description of an "Old CALC" database procedure as opposed to a standard database procedure as introduced with DMS1100 level 8.

I-QU PLUS-1 does not currently support simulation using the newer form of CALC database procedure.

Separate DBPGEN for Single-Thread Interface:

To implement the user-written DBPs with the single-thread DMR code, generate I-QU PLUS-1 and follow the procedure shown in section 3.4.3, "Database Procedure SGSs", of the I-QU PLUS-1 Installation Guide. As stated in the installation guide, the DBPGEN runstream ***must be*** run against a "fresh" copy of the schema (i.e., one that has ***not*** had a DBPGEN/DBPUP run against it previously);

therefore, it is a good idea to place the @START of this runstream in the runstream that compiles your schema and subschemas. You probably already do something similar (the execution of DBPUP) for user-written DBPs in the multi-thread environment. Remember, this DBPGEN is for the single-thread interface collected with I-QU PLUS-1; therefore, you ***must*** place a fresh copy of the schema/subschema absolutes into a file other than the one you use for the multi-thread interface.

User-CALC DBPs (up to 3) and CALSIM:

The inclusion of user-written "Old CALC" DBPs with CALSIM takes a little more work. The following is a short write-up on how to link three (3) user routines to the reserved words, "U1", "U2" and "U3", accepted on the DEFINE C directive. It requires applying changes to two elements: CALSIM and GENSKEL.

Apply the following corrections to CALSIM using COMUS (see Appendix F of the installation guide):

```
*CALSIM
-162
          TNE,U     A4,2                . Was U1 on the DEF C?
          J         CS00U1              . Yes
          TNE,U     A4,3                . Was U2 on the DEF C?
          J         CS00U2              . Yes
          TNE,U     A4,4                . Was U3 on the DEF C?
          J         CS00U3              . Yes
-169
CS00U1    J         CALC-entry-1
          J         CS0700
CS00U2    J         CALC-entry-2
          J         CS0700
CS00U3    J         CALC-entry-3
          J         CS0700
```

The correct entry points for the user "Old CALC" routines must be used in place of CALC-entry-1, CALC-entry-2 and CALC-entry-3.

Note: The line numbers shown are for release level 11R2 (current release) and may be different for your release. See the attached example for the implementation of U4 to see the surrounding code.

How to Install User-Written CALC Database Procedures

The following changes must be made ***manually*** to GENSKEL after the inclusion of the DMSCALC and RANDENTIAL routines (with an editor, look for "DEF CALSIM" and place the changes immediately before it):

```
IN pf.U1-relocatable
IN pf.U2-relocatable
IN pf U3-relocatable
```

User-CALC DBPs (more than 3) and CALSIM:

If you need to link more than three routines to CALSIM, you will need to apply changes to five elements: CALSIM, GENSKEL, IQUCDMS, KEYWORDS and PARSE.

Attached to these instructions is a partial listing of the five elements showing where the code should be changed. Line numbers are not shown as the numbers will be different from release to release. Use an editor to examine the elements in the F1 file to determine the appropriate line numbers. The changes shown are for a forth DBP called, "U4".

The entries in two of these elements (KEYWORDS and PARSE) have a one-to-one relationship to each other. For example, if you add an "U4" entry to KEYWORDS, you must add an "U4" entry to PARSE. Violating this one-to-one relationship will cause the generation to fail. Also, notice that there are two reserved names shown on each entry. The first name ("U1", "U2", etc.) is the "Old CALC" routine name to be used on the DEFINE C directive. You may change this to a more meaningful name so long as it is unique within the KEYWORDS and PARSE elements. The second name ("T-U1", "T-U2", etc.) is an internal name used to generate data items referenced in IQUCDMS. Whatever name you choose for the second name must match the procedural code you add to IQUCDMS.

Steps to Implement Local Code:

1) Do a "vanilla" TAPE-TO-DISK gen of I-QU PLUS-1.

2) Manually make the changes to GENSKEL in the F0 file.

3) Apply changes to the other elements through COMUS.

4) Provide the SGSs for the CO$TCONPROCESSOR, CO$MCONPROCESSOR and CO$DCONPROCESSOR (only release level 5R1 or higher of the processors may be used). On most systems the SGSs would appear as follows:

```
CO$TCONPROCESSOR CNI ''TCON'' OA ''NONE'' LNI ''SYS$LIB$*GSA''
CO$DCONPROCESSOR CNI ''DCON''
CO$MCONPROCESSOR CNI ''MCON''
```

5) During the COMUS BUILD dialog, specify an "ALL" type gen (the gen may be DISK-TO-TAPE or DISK-TO-DISK, what ever you prefer).

6) If all goes well, do a COMUS INSTALL.

Since KMSystems, Inc., does not support implementing more than three routines, this procedure for installing more than 3 routines has not been tried in-house; therefore, **we recommend that you install a second copy of I-QU PLUS-1 using a second mode** (see the boxed paragraph in section 3.5.1 of the installation guide). Your currently installed copy will continue to function as before but without user-written DBP capability.

```
CALSIM
 .
 .
        . *** SET REGISTERS FOR ENTRY TO DMSCALC....
        CS0030    L,U       X2,CALCTYPE          . BASE X2 TO SIM S$WORK ADDR
                  AN,U      X2,0407              .  IEC011 IS 0407 OFF S$WORK
                  L,U       X11,KEYP00           .
                  L,U       X9,CALP00            .
        . *** JUMP TO CALC ROUTINE                .
                  TNE,U     A4,0                 .
                  J         CS0040               .
                  TNE,U     A4,1                 . RANDENTIAL ???
                  J         CS0050               . YES, JUMP TO GS13 ENTRY
                  J         CS0700               . **** FOR ADDING USER CALC ***
        .
        CS0040    J         GS11                 . DO DMSCALC
                  J         CS0700               . GO PROCESS RESULTS
        .
        CS0050    J         GS13                 . DO RANDENTIAL
                  J         CS0700               . GO PROCESS RESULTS.
        .
        . *** RETURN FROM DMSCALC ROUTINE
        .
        CS0700*   L         A1,SVPGCH            .
                  S         A15,0,A1             .
                  L         A0,(11,A5,11,X1)     .
                  LRS       A0,SREGS             .
        .
                  J         C$RET0               . JUMP BACK TO USER
        .
                  END       .
 .
 .

GENSKEL
 .
 .
        *
        *. BUILD THE APPROPRIATE CALC SIMULATION R-OPTION MAP BASED ON THE CALCFILE SGS
        *.
        *PRCSR MAP RI CALSIMR NONE F4 F4 NONE NONE
             IN F4.CALSIM
        *IF ROW SEARCH FROM CALCFILE,1,2 FOR ''DMSCALC''
             IN [CALCFILE,1,1,1].GS11
        *ELSE . FOUND DMSCALC
             IN F4.GS11DUMMY
        *END . ~FOUND DMSCALC
        *IF ROW SEARCH FROM CALCFILE,1,2 FOR ''RANDENTIAL''
             IN [CALCFILE,1,1,1].GS13
        *ELSE . FOUND RANDENTIAL
             IN F4.GS13DUMMY
        *END  ~FOUND RANDENTIAL
             DEF CALSIM
             DEF C$BCALSIM
             END
 .
 .
```

```
TNE,U    A4,2
J        CS00U4
```

```
CS00U4   J    U4
         J    CS0700
```

```
IN   pf.U4
```

```
IQUCDMS
 .
 .
                /*************************************************************
                ** DEFINE CALC SIMULATION PARAMETERS FOR CALSIM COMMAND
                *************************************************************
                 CALC-SIM-DEF SECTION.
                 CALC-SIM-D.
                *** IMPORT REFERENCED INDEX VARIABLES FOR THIS PROGRAM
                     SET WLSX TO SH-WLSX.
                     SET DWX TO SH-DWX.

                     IF WLN-WORDN (1) = 0
                        OR WLN-WORDN (2) = 0
                         MOVE 55 TO SW-ERROR
                         PERFORM 9999-ERRMSG
                         GO TO CALC-SIM-EXIT.

               *** BASIC EDIT COMPLETE, NOW START BUILDING PARAMETERS...
                     MOVE WLN-WORDINT (1) TO CS-NO-PAGES.
                     MOVE WLN-WORDINT (2) TO CS-NO-CHAINS.
                     SET CS-NO-KEYS TO WLSX.
                     SUBTRACT 1 FROM CS-NO-KEYS.

                     IF WLA-TYPE (4) = T-D
                         MOVE 0 TO CS-CALC-TYPE-CODE.
                     IF WLA-TYPE (4) = T-R
                         MOVE 1 TO CS-CALC-TYPE-CODE.
                     IF WLA-TYPE (4) = T-U1
                         MOVE 2 TO CS-CALC-TYPE-CODE.
                     IF WLA-TYPE (4) = T-U2
                         MOVE 3 TO CS-CALC-TYPE-CODE.
                     IF WLA-TYPE (4) = T-U3
                         MOVE 4 TO CS-CALC-TYPE-CODE.

                     MOVE 0 TO CS-DATA-MODE.
                     IF WLA-TYPE (5) = T-FDATA
                         MOVE 1 TO CS-DATA-MODE
                     ELSE
                     IF WLA-TYPE (5) NOT = T-SPACE
                        AND WLA-TYPE (5) NOT = T-ASCII
                         PERFORM 9900-INTERNAL-ERROR.
 .
 .
```

IF WLA-TYPE (4) = T-U4
    MOVE 5 TO CS-CALC-TYPE-CODE.

```
KEYWORDS
 .
 .
          "          THIS KEYWORD TABLE CONTAINS ALL OF THE SYNTACTIC KEY WORDS
          "          FOR INPUT PARSING. THE FIRST SECTION OF THE TABLE DEFINES
          "          ALL KEY WORDS WHICH CAN EVER BE USED AS A COMMAND NAME
          "          OR IN A PARAMETER FIELD. THE FIRST DATA FIELD IS THE
          "          MINIMUM ABBREVIATION LENGTH IN CHARACTERS FOR COMMAND NAMES,
          "          IF IT IS NON-ZERO. IF IT IS ZERO, THEN THIS KEYWORD MAY
          "          NOT BE USED AS A COMMAND NAME. THE SECOND PARAMETER
          "          FIELD CONTAINS THE PARSER TOKEN TYPE. SYNONYMS ARE MARKED
          "          BY HAVING A -1 CODED WHICH SETS THIER PARSER TYPE TO THAT
          "          OF THE IMMEDIATELY PRECEEDING KEYWORD. THE THIRD PARAMETER
          "          (IF PRESENT) CONTAINS THE COMMAND TYPE FOR THIS COMMAND.
          "          ALL NON-COMMAND TYPE KEYWORDS HAVE THIS FIELD ZERO AS DO THE
          "          SPECIAL COMMANDS NOOP AND ENDIF AND THE UN-COMMAND @EOL.
          "
          "          THE FOURTH PARAMETER IS USED TO PRE-SET THE OBJECT SUB-COMMAND
          "          TO DIFFERENTIATE COMMANDS WHICH USE THE SAME OBJECT COMMAND
          "          CODE BUT DIFFER SLIGHTLY AT EXECUTION TIME.
          "
          "          OBJECT TYPES 300..318 ARE PSEUDO TYPES FOR CONTROL COMMANDS
          "             AND DO NOT ACTUALLY APPEAR IN THE GENERATED COMMAND TABLES.
          "
$GRATE    1
$VALUE                              9     9     9     9
$STATE                              0     +1    0     0
" 345678901234567890123456789012 3.56789.12345678901234567890123 4
 .
 .
TRIMEDIT                            5     +0    14          " $$ T-TRIMEDIT
TED                                 2     -1    14
U                  " SHORT FOR 'UPDATE', ALSO FILE MODE     " $$ T-U
U1                                                          " $$ T-U1
U2                                                          " $$ T-U2
U3                                                          " $$ T-U3
UB1                                                         " $$ T-UB1
UB6                                                         " $$ T-UB6
UB9                                                         " $$ T-UB9
UN                                                          " $$ T-UN
 .
 .
U4                                  " $$ T-U4
```