16 RACF

May 1999

update

In this issue

RACF Update

Published by

Xephon
27-35 London Road
Newbury
Berkshire RG14 1JL
England
Telephone: 01635 38030
From USA: 01144 1635 38030
E-mail: xephon@compuserve.com

North American office

Xephon/QNA
1301 West Highway 407, Suite 201-405
Lewisville, TX 75077-2150
USA
Telephone: 940 455 7050

Contributions

Articles published in *RACF Update* are paid for at the rate of £170 ($250) per 1000 words and £90 ($140) per 100 lines of code for original material. To find out more about contributing an article, without any obligation, please contact us at any of the addresses above and we will send you a copy of our *Notes for Contributors*.

RACF Update on-line

Code from *RACF Update* can be downloaded from our Web site at http://www.xephon.com; you will need the user-id shown on your address label.

Editor

Robert Burgess

Disclaimer

Readers are cautioned that, although the information in this journal is presented in good faith, neither Xephon nor the organizations or individuals that supplied information in this journal give any warranty or make any representations as to the accuracy of the material it contains. Neither Xephon nor the contributing organizations or individuals accept any liability of any kind howsoever arising out of the use of such material. Readers should satisfy themselves as to the correctness and relevance to their circumstances of all advice, information, code, JCL, and other contents of this journal before making any use of it.

Subscriptions and back-issues

A year's subscription to *RACF Update*, comprising four quarterly issues, costs £190.00 in the UK; $290.00 in the USA and Canada; £196.00 in Europe; £202.00 in Australasia and Japan; and £200.50 elsewhere. In all cases the price includes postage. Individual issues, starting with the August 1995 issue, are available separately to subscribers for £50.50 ($77.50) each including postage.

Printed in England.

Deleting RACF profiles

Have you ever encountered a RACF profile that could not be deleted? Under CICS protected by RACF, users can type a '‡' and the RACF profile cannot be deleted by an RDELETE command. The solution is to use the following RACFDGEN Assembler program and RACFEXEC.

RACFDGEN

```
//*
//ASM  EXEC ASMACL,MAC1='SYS1.MODGEN',PARM.LKED='AC=1'
//ASM.SYSIN DD *
RACFDGEN CSECT ,
RACFDGEN AMODE 31                  COMMAND'S ADDRESSING MODE
RACFDGEN RMODE ANY                 COMMAND'S RESIDENCY MODE
RACFDGEN CSECT
*
* RACFDGEN : PGM ASM
*            DELETION OF A RACF RESOURCE
*
*
RØ       EQU   Ø
R1       EQU   1
R2       EQU   2
R3       EQU   3
R4       EQU   4
R5       EQU   5
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
RIØ      EQU   1Ø
R11      EQU   11
R12      EQU   12
R13      EQU   13
R14      EQU   14
R15      EQU   15
*
         USING RACFDGEN,R12,R11
         SAVE  (14,12),,RACFDGEN.&SYSDATE   SAVE REGISTERS
         LR    R12,R15           |
         LA    R11,4Ø95(R12)     |  INITIALIZATION OF BASE REGISTERS
         LA    R11,1(R11)        |
         LR    R2,R13               ADDRESS OF PRECEDING SAVEAREA
         LA    R13,SAVEAREA         ADDRESS OF PROGRAM SAVEAREA
         ST    R13,8(R2)            CREATION OF POINTERS FOR
         ST    R2,4(R13)             SAVEAREA CHAIN
```

```
*
*
*        START OF PROGRAM
*
*********************************************************************
*         RESOURCE DEFINED UNDER RACF?
*********************************************************************
TSTGRESS DS    ØH
         WTO   '  '
         WTO   'RACFDGEN '
*
*   CLASS OF THE RESOURCE TO BE DELETED
         MVC   WCLASS(8),=C'STARTED '
*
*   LENGTH OF NAME OF THE RESOURCE TO BE DELETED
         MVC   WGRESSL(1),=FL1'6'
*
*   NAME OF RESOURCE TO BE DELETED
         MVC   WGRESS(6),=C'CIC*.*'
*
         LA    R8,FINWA1
         LA    R15,WRKAREA1
         SR    R8,R15
         ST    R8,WRKAREA1
         XC    WA1(FINWA1-WA1),WA1
LOCRESS  ICHEINTY LOCATE,TYPE='GEN',                                  X
               GENERIC=NO,                                            X
               ENTRY=WGRESSL,          FUNCTION                       X
               CLASS=WCLASS,           CLASS-ID                       X
               WKAREA=WRKAREA1,        WRKAREA                        X
               RELEASE=(2.1),MF=I
*              RELEASE=(1.8),DATAMAP=OLD,MF=I
*
TESTRF1  DS    ØH                      CHECK  RETURN CODE
         ST    RØ,RETCODEØ
         ST    R15,RETCODEF
         CH    R15,=XL2'ØC'            RC > ØC (12) -> ERREUR
         BH    ERRORICH
         B     BRCHTBL1(R15)
BRCHTBL1 DS    ØH                    CHECK  RETURN CODE
         B     DELGRESS   X'ØØ'      OK, FUNCTION FOUND
         B     ERRORICH   X'Ø4'      INVALID CODE
         B     ERRORICH   X'Ø8'      EXISTING ENTRY
         B     GRESSNF1   X'ØC'      ENTRY NOT FOUND
*
*
*********************************************************************
*         DELETION OF THE RESOURCE
*********************************************************************
DELGRESS DS    ØH
         WTO   'RACFDGEN : RESOURCE FOUND'
DELRESS  ICHEINTY DELETE,TYPE='GEN',                                  X
```

```
               GENERIC=NO,                                            X
               ENTRY=WGRESSL,         RESOURCE                        X
               CLASS=WCLASS,          CLASS-ID                        X
               RELEASE=(2.1),MF=I
*              RELEASE=(1.8),DATAMAP=NEW,MF=I
*
TESTRF2  DS    ØH                     CHECK  RETURN CODE
         ST    RØ,RETCODEØ
         ST    R15,RETCODEF
         CH    R15,=XL2'ØC'            RC > ØC (12) —> ERROR
         BH    ERRORICH
         B     BRCHTBL2(R15)
BRCHTBL2 DS    ØH                     CHECK  RETURN CODE
         B     DELETEOK   X'ØØ'       OK, DELETION OK
         B     ERRORICH   X'Ø4'       INVALID CODE
         B     ERRORICH   X'Ø8'       EXISTING ENTRY
         B     GRESSNF2   X'ØC'       ENTRY NOT FOUND
*
Ø1820506
DELETEOK DS    ØH
         WTO   'RACFDGEN : DELETION COMPLETED'
*
*
*  NORMAL END OF PROGRAM
FIN      DS    ØH
         L     R13,SAVEAREA+4         RESTORATION OF REGISTER 13
         RETURN (14,12),T,RC=Ø        RETURN TO MVS
*
*  NORMAL END OF PROGRAM
GRESSNF1 DS    ØH
         WTO   'RACFDGEN : RESOURCE NOT FOUND 1'
         ABEND 1ØØ1,DUMP
         L     R13,SAVEAREA+4         RESTORATION OF REGISTER 13
         RETURN (14,12),T,RC=1        RETURN TO MVS
*
*  FIN NORMALE DU PROGRAMME.
GRESSNF2 DS    ØH
         WTO   'RACFDGEN : RESOURCE NOT FOUND 2'
         ABEND 1ØØ2,DUMP
         L     R13,SAVEAREA+4         RESTORATION OF REGISTER 13
         RETURN (14,12),T,RC=2        RETURN TO MVS
*
*
*  END OF PROGRAM WITH ERROR
ERRORICH DS    ØH
         WTO   'RACFDGEN: SERIOUS PROBLEM'
         ABEND 1111,DUMP
         L     R13,SAVEAREA+4         RESTORATION OF REGISTER 13
         RETURN (14,12),T,RC=12       RETURN TO MVS
*
*
* PREREQUISITES
```

```
         LTORG
WLIBCLAS DS    ØCL8
         DC    C'GCI'
WNUMCLAS DC    C' '
         DC    C'TRAN'
WZERO    DC    H'Ø'
*
SAVEAREA DS    18F
*
         DC    C'RRRR'
RETCODEF DS    F
RETCODEØ DS    F
         DC    C'GGGG'
WGRESSL  DS    FL1                 LENGTH OF NAME OF RESOURCE
WGRESS   DS    CL2Ø                NAME OF THE RESOURCE
         DC    C'CCCC'
WCLASS   DS    CL8                 CLASS
*
*
         DC    C'WWWW'
WRKAREA1 DS    ØF                  WORKAREA FOR MACRO ICHEINTY. LEVEL
         DS    F                   TOTAL LENGTH OF WORKAREA
WA1      DS    XL2Ø                RESERVED AREA
LNGDATA1 DS    F                   LENGTH OF RETURNED DATA DS LA WA
DATA1    DS    ØX
ZTESTL1  DS    XL2                 AREA TO TEST
LEVEL1   DS    1ØX                 LEVEL
FINWA1   EQU   *
*
*
         END
//*
//*
//LKED.SYSLMOD DD DSN=RDVUSER.LINKLIB,DISP=SHR  <-BIB AUTHORIZED
//LKED.SYSIN   DD *
    SETCODE AC(1)
    NAME RACFDGEN(R)
//*
```

RACFEXEC

```
//*
//*
//*
//STEP1   EXEC PGM=RACFDGEN
//STEPLIB  DD  DSN=RDVUSER.LINKLIB,DISP=SHR       BB AUTHORIZED
//SYSPRINT DD  SYSOUT=T
//SYSUDUMP DD  SYSOUT=T
//*
```

A RACFPROF ISPF interface

This program is designed to encourage user accountability and reduce the administration of dataset profiles/groups by the RACF administrator. It gives users a way of controlling access to their databases, allowing them to create/delete/modify and display the contents of dataset profiles/groups.

RACFPROF

```
/******************************** REXX ********************************/
/* Program name:                                                      */
/*   RACFPROF        (RACFPROF ISPF interface)                        */
/*- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - */
/* Purpose:                                                           */
/*       - To encourage user accountability and lessen the RACF       */
/*         administration of dataset profiles/groups on the part      */
/*         of the RACF administrator.                                 */
/*                                                                    */
/*       - To provide the user with an easy way to control access to  */
/*         his/her datasets.                                          */
/*                                                                    */
/*       - Creation, deletion, modification and displaying contents   */
/*         of dataset profiles/groups.                                */
/*                                                                    */
/*                                                                    */
/*- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - */
/* Invocation : TSO RACFPROF  *you can add to your panel as an option */
/*- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - */
/* Input:                                                             */
/*   Parameters:                                                      */
/*              None is needed.                                       */
/*                                                                    */
/*- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - */
/* Process:                                                           */
/*          Upon invocation, the program will display the invoker     */
/*          profiles by virtue of masking.                            */
/*                                                                    */
/*          However, the user can display any other profile by        */
/*          changing the mask field on the initial panel to any       */
/*          other user-id or HLQ                                      */
/*                                                                    */
/*- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - */
/* Output:           : full panel interface.                          */
/*- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - */
/* RACF requirements : In order to fully utilize this utility, one or */
/*                     more of the following must be true:            */
```

```
/*                     1- You are the profile owner                    */
/*                     2- You have group special connection            */
/*                     3- Auditor or Special attribute                 */
/*                                                                     */
/*- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - */
/*********************************************************************/
trace

ADDRESS ISPEXEC              /* set our environment                    */
Call R_Open                  /* display the panel via libdef services */
Mask = "*"                   /* always default to the caller-id        */
RcProf = Ø                   /* set this variable                      */
Do while RcProf ¬= 8         /* as long as our return code is good     */
   Call Prof_Displ           /* go get the main RACF info              */
end                          /* finished for now                       */
Call R_Close                 /* end the entire program                 */
Return                       /* get back from all calls                */

/*********************************************************************/
/* Display RACF lines, put in table, display in panel                */
/*********************************************************************/
Prof_Displ:
"CONTROL DISPLAY SAVE"
lcmd = ""
RacfMsg = ""

"TBCREATE,RacfProf,NAMES(ProfLine),
          NOWRITE,REPLACE"
X = OUTTRAP("RACF.",'*',"NOCONCAT")
ADDRESS TSO "SR MASK(" || Mask || ")"
DO I = 1 TO RACF.Ø
   ProfLine = Racf.i
   "TbAdd RacfProf"
END

"TbTop RacfProf"
OldMask = Mask
RcProf  = Ø
DO WHILE RcProf < 4
   "TbDispl RacfProf PANEL(RACFPROF) position(ProfPos)"
   RcProf = RC
   if Mask ¬= OldMask then do
      RcProf = 24
   end
   SlProf = Ztdsels
   if lcmd = "INFO" & ztdsels > Ø then call Info_Displ
   if lcmd = "ADDG" & ztdsels > Ø then call Racf_Add_Grp
   if lcmd = "AUTH" & ztdsels > Ø then call Auth_Displ
   if lcmd = "DSNS" & ztdsels > Ø then call Dsns_Displ
   if lcmd = "PE"   & ztdsels > Ø then call Racf_Permit
   if lcmd = "REM"  & ztdsels > Ø then call Racf_Remove
   if lcmd = "ADD"  & ztdsels > Ø then do
```

```
      call Racf_Add_Prof
      RcProf = 24
      end
   if lcmd = "DEL"  & ztdsels > Ø then do
      call Racf_Del_Prof
      RcProf = 24
      end
   Ztdsels = SlProf
   DO WHILE Ztdsels > 1 & RcProf < 8
      "TBDISPL RacfProf position(ProfPos)"
      RcProf = RC
      SlProf = Ztdsels
      if lcmd = "ADDG" then call Racf_Add_Grp
      if lcmd = "AUTH" then call Auth_Displ
      if lcmd = "DSNS" then call Dsns_Displ
      if lcmd = "PE"   then call Racf_Permit
      if lcmd = "REM"  then call Racf_Remove
      if lcmd = "ADD"  then do
         call Racf_Add_Prof
         RcProf = 24
         end
      if lcmd = "DEL"  then do
         call Racf_Del_Prof
         RcProf = 24
         end
      Ztdsels = SlProf
      END
   END
lcmd = ""
"TBEND,RacfProf"
"CONTROL DISPLAY RESTORE"
RETURN RcProf

/*********************************************************************/
/* Display Info lines, get me the creation date of said profile      */
/*********************************************************************/
/* New code starts here.                                             */
Info_Displ:
"CONTROL DISPLAY SAVE"
lcmd = ""
RacfMsg = ""
"TBCREATE,RacfInfo,NAMES(InfoLine),
          NOWRITE,REPLACE"                    /* new table creation */
parse value profline with AuthMask " " .      /* get me that info   */
X = OUTTRAP("RACF.",'*',"NOCONCAT")
ADDRESS TSO "LD DA('" || AuthMask || "') HISTORY"  /* issue command  */
AuthFlag = Ø
DO I = 1 TO RACF.Ø
   InfoLine = strip(Racf.i)
   if substr(InfoLine,1,1) = "-" then iterate     /* don't need this */
   if substr(InfoLine,1,1) = " " then do
```

```
      parse value strip(Racf.24) with Cday Cmon .
      Cday = Cday "Day Of Year"    /* reformat to new form */
      Cmon = Cmon "Year"
   end
   if AuthFlag = 1 then "TbAdd RacfInfo"
END

"TbTop RacfInfo"
RcInfo  = Ø
DO WHILE RcInfo < 4
   "TbDispl RacfInfo PANEL(RACFINFO) position(InfoPos)"
   RcInfo = RC
   SlInfo = Ztdsels
      "TbDelete RacfInfo"
   Ztdsels = SlInfo
   DO WHILE Ztdsels > 1 & RcInfo < 8
      "TBDISPL RacfInfo position(InfoPos)"
      RcInfo = RC
         "TbDelete RacfInfo"
      Ztdsels = SlInfo
      END
 END
lcmd = ""
"TBEND,RacfInfo"
"CONTROL DISPLAY RESTORE"
RETURN Ø
/* New code ends here                                                  */

/*********************************************************************/
/* Display RACF lines                                                */
/*********************************************************************/
Auth_Displ:
"CONTROL DISPLAY SAVE"
lcmd = ""
RacfMsg = ""
"TBCREATE,RacfAuth,NAMES(AuthLine), NOWRITE,REPLACE"
parse value profline with AuthMask " " .
X = OUTTRAP("RACF.",'*',"NOCONCAT")
ADDRESS TSO "LD DA('" || AuthMask || "') AUTH"
AuthFlag = Ø
DO I = 1 TO RACF.Ø
   AuthLine = strip(Racf.i)
   if substr(AuthLine,1,1) = "-" then iterate
   if substr(AuthLine,1,1) = " " then iterate
   if substr(AuthLine,1,2Ø) = "ID    ACCESS   CLASS" then iterate
   if substr(AuthLine,1,11) = "YOUR ACCESS" then do
      parse value strip(racf.5) with . own uac .  /* addition*/
      parse value strip(racf.9) with adt . /*new addition*/
      x = i + 2
      parse value strip(Racf.x) with YourAuth " " .
   end
```

```
   if AuthFlag = 1 then "TbAdd RacfAuth"
   if substr(AuthLine,1,13) = "ID     ACCESS" then AuthFlag = 1
END

"TbTop RacfAuth"
RcAuth  = Ø
DO WHILE RcAuth < 4
   "TbDispl RacfAuth PANEL(RACFAUTH) position(AuthPos)"
   RcAuth = RC
   SlAuth = Ztdsels
   if lcmd = "LU"   & ztdsels > Ø then call Lst_User
   if lcmd = "LG"   & ztdsels > Ø then call Lst_Group
   if lcmd = "ADDG" & ztdsels > Ø then call Racf_Add_Grp
   if lcmd = "REM"  & ztdsels > Ø then do
      parse value authline with UserId " " .
      call Racf_Remove
      "TbDelete RacfAuth"
   end
   Ztdsels = SlAuth
   DO WHILE Ztdsels > 1 & RcAuth < 8
      "TBDISPL RacfAuth position(AuthPos)"
      RcAuth = RC
      if lcmd = "LU"  then call Lst_User
      if lcmd = "LG"  then call Lst_Group
      if lcmd = "ADDG" then call Lst_Group
      if lcmd = "REM" then do
         parse value authline with UserId " " .
         call Racf_Remove
         "TbDelete RacfAuth"
      end
      Ztdsels = SlAuth
      END
   END
lcmd = ""
"TBEND,RacfAuth"
"CONTROL DISPLAY RESTORE"
RETURN Ø

/*********************************************************************/
/* Display RACF lines                                                */
/*********************************************************************/
Dsns_Displ:
"CONTROL DISPLAY SAVE"
lcmd = ""
RacfMsg = ""
"TBCREATE,RacfDsns,NAMES(DsnsLine),
          NOWRITE,REPLACE"
parse value profline with DsnsMask " " .
X = OUTTRAP("RACF.",'*',"NOCONCAT")
ADDRESS TSO "LD DA('" || DsnsMask || "') DSNS"
DsnsFlag = Ø
```

```
DO I = 1 TO RACF.Ø
   DsnsLine = Racf.i
   if substr(DsnsLine,1,1) = "-" then iterate
   if DsnsFlag = 1 then "TbAdd RacfDsns"
   if substr(DsnsLine,1,2Ø) = "CATALOGUED DATA SETS" then DsnsFlag = 1
END

"TbTop RacfDsns"
RcDsns  = Ø
DO WHILE RcDsns < 4
   "TbDispl RacfDsns PANEL(RACFDSNS) position(DsnsPos)"
   RcDsns = RC
   SlDsns = Ztdsels
   if lcmd = "PE"   & ztdsels > Ø then call Racf_Permit
   if lcmd = "REM"  & ztdsels > Ø then call Racf_Remove
   Ztdsels = SlDsns
   DO WHILE Ztdsels > 1 & RcDsns < 8
      "TBDISPL RacfDsns position(DsnsPos)"
      RcDsns = RC
      if lcmd = "PE"  then call Racf_Permit
      if lcmd = "REM" then call Racf_Remove
      Ztdsels = SlDsns
      END
   END
lcmd = ""
"TBEND,RacfDsns"
"CONTROL DISPLAY RESTORE"
RETURN Ø
/*———————————————————————————————————————
 * Display the pop-up panel to update the RACF access
 *———————————————————————————————————————*/
Racf_Permit:
"CONTROL DISPLAY SAVE"
lcmd    = ""
RacfMsg = ""
Racf.   = ""
"ADDPOP   POPLOC(Z)    ROW(2)   COLUMN(2)"
IF Rc ¬= Ø THEN say "AddPop error Rc("Rc")," zerrlm
ELSE do
   parse value ProfLine with PermDsn " " .
   Racf.Ø = 1
   do while Racf.Ø > Ø
      X = OUTTRAP("RACF.",'*',"NOCONCAT")
      Racf.Ø = Ø
      "Display panel(RacfPop1)"
      if Rc < 8 then do
         RacfAcc = ""
         if None = "X" then RacfAcc = "NONE"
         if Execute = "X" then RacfAcc = "EXECUTE"
         if Update = "X" then RacfAcc = "UPDATE"
         if Read = "X" then RacfAcc = "READ"
```

```
         if Control = "X" then RacfAcc = "CONTROL"
         if Alter = "X" then RacfAcc = "ALTER"
         ADDRESS TSO "PE '" || PermDsn || "' GEN ID(" || UserId || "),
                     ACC(" || RacfAcc ||")"
         if Racf.Ø > Ø then RacfMsg = Racf.1 || ", " || Racf.2
      end
   end
   "REMPOP   ALL"
      IF Rc ¬= Ø THEN say "RemPop error Rc("Rc")," zerrlm
end
lcmd = ""
"CONTROL DISPLAY RESTORE"
return

/*———————————————————————————————————————————
 * Display the pop-up panel to delete the RACF access
 *———————————————————————————————————————————*/
Racf_Remove:
"CONTROL DISPLAY SAVE"
lcmd    = ""
RacfMsg = ""
Racf.   = ""
"ADDPOP   POPLOC(Z)    ROW(2)   COLUMN(2)"
IF Rc ¬= Ø THEN say "AddPop error Rc("Rc")," zerrlm
ELSE do
   parse value ProfLine with PermDsn " " .
   Racf.Ø = 1
   do while Racf.Ø > Ø
      X = OUTTRAP("RACF.",'*',"NOCONCAT")
      Racf.Ø = Ø
      "Display panel(RacfPop5)"
      if Rc < 8 then do
         RacfAcc = ""
         ADDRESS TSO "PE '" || PermDsn || "' GEN ID(" || UserId || "),
                     DELETE"
         if Racf.Ø > Ø then RacfMsg = Racf.1 || ", " || Racf.2
      end
   end
   "REMPOP   ALL"
      IF Rc ¬= Ø THEN say "RemPop error Rc("Rc")," zerrlm
end
lcmd = ""
"CONTROL DISPLAY RESTORE"
return

/*———————————————————————————————————————————
 * Display the pop-up panel to update the RACF access
 *———————————————————————————————————————————*/
Racf_Add_Prof:
"CONTROL DISPLAY SAVE"
lcmd    = ""
RacfMsg = ""
```

```
Racf.   = ""
"ADDPOP   POPLOC(Z)    ROW(2)   COLUMN(2)"
IF Rc ¬= Ø THEN say "AddPop error Rc("Rc")," zerrlm
ELSE do
   parse value profline with PermDsn " " .
   Racf.Ø = 1
   do while Racf.Ø > Ø
      X = OUTTRAP("RACF.",'*',"NOCONCAT")
      Racf.Ø = Ø
      "Display panel(RacfPop2)"
      if Rc < 8 then do
        ADDRESS TSO "ADDSD '" || PermDsn || "' GEN"
        if ModelDsn ¨= " " then
        ADDRESS TSO "PE '" ||PermDsn||"' GEN FROM('" ||ModelDsn || "'),
                    FGEN FCLASS(DATASET)"
         else
         if Racf.Ø > Ø then RacfMsg = Racf.1 || ", " || Racf.2
      end
   end
   "REMPOP   ALL"
      IF Rc ¬= Ø THEN say "RemPop error Rc("Rc")," zerrlm
end
lcmd = ""
"CONTROL DISPLAY RESTORE"
return

/*———————————————————————————————
 * Add Groups to RACF
 *———————————————————————————————*/
 Racf_Add_Grp:
 "CONTROL DISPLAY SAVE"
 lcmd    = ""
 RacfMsg = ""
 Racf.   = ""
 "ADDPOP   POPLOC(Z)    ROW(2)   COLUMN(2)"
 IF Rc ¬= Ø THEN say "AddPop error Rc("Rc")," zerrlm
 ELSE do
       "Display panel(RacfPop7)"
       if Rc < 8 then do
          if Supgroup ¨= " " then  /*Superior group requested...*/
          ADDRESS TSO "ADDGROUP " || Group || " SUPGROUP(" || Supgroup |
          else
          ADDRESS TSO "ADDGROUP " || Group || " "
       /* else
          if Racf.Ø > Ø then RacfMsg = Racf.1 || ", " || Racf.2 */
                     end
    "REMPOP   ALL"
       IF Rc ¬= Ø THEN say "RemPop error Rc("Rc")," zerrlm
 end
 lcmd = ""
"CONTROL DISPLAY RESTORE"
RETURN
```

```
/*-------------------------------------------------------------
 * Display the pop-up panel to update the RACF access
 *-------------------------------------------------------------*/
Racf_Del_Prof:
"CONTROL DISPLAY SAVE"
lcmd    = ""
RacfMsg = ""
Racf.   = ""
"ADDPOP   POPLOC(Z)    ROW(2)   COLUMN(2)"
IF Rc ¬= Ø THEN say "AddPop error Rc("Rc")," zerrlm
ELSE do
   parse value profline with PermDsn " " .
   Racf.Ø = 1
   do while Racf.Ø > Ø
      X = OUTTRAP("RACF.",'*',"NOCONCAT")
      Racf.Ø = Ø
      "Display panel(RacfPop6)"
      if Rc < 8 then do
         ADDRESS TSO "DELDSD '" || PermDsn || "' GEN"
         if Racf.Ø > Ø then RacfMsg = Racf.1 || ", " || Racf.2
      end
   end
   "REMPOP   ALL"
      IF Rc ¬= Ø THEN say "RemPop error Rc("Rc")," zerrlm
end
lcmd = ""
"CONTROL DISPLAY RESTORE"
return

/*-------------------------------------------------------------
 * Display the pop-up panel to update the RACF access
 *-------------------------------------------------------------*/
Con_User:
"CONTROL DISPLAY SAVE"
lcmd    = ""
RacfMsg = ""
Racf.   = ""
"ADDPOP   POPLOC(Z)    ROW(2)   COLUMN(2)"
IF Rc ¬= Ø THEN say "AddPop error Rc("Rc")," zerrlm
ELSE do
   Racf.Ø = 1
   do while Racf.Ø > Ø
      X = OUTTRAP("RACF.",'*',"NOCONCAT")
      Racf.Ø = Ø
      "Display panel(RacfPop3)"
      if Rc < 8 then do
        ADDRESS TSO "CONNECT "|| ConUser|| " GROUP(" || ConGroup || ")"
        if Racf.Ø > Ø then RacfMsg = Racf.1 || ", " || Racf.2
      end
   end
   "REMPOP   ALL"
```

```
      IF Rc ¬= Ø THEN say "RemPop error Rc("Rc")," zerrlm
end
lcmd = ""
"CONTROL DISPLAY RESTORE"
return

/*————————————————————————————————————
 * Display the pop-up panel to update the RACF access
 *————————————————————————————————————*/
Rem_User:
"CONTROL DISPLAY SAVE"
lcmd    = ""
RacfMsg = ""
Racf.   = ""
"ADDPOP   POPLOC(Z)    ROW(2)   COLUMN(2)"
IF Rc ¬= Ø THEN say "AddPop error Rc("Rc")," zerrlm
ELSE do
   Racf.Ø = 1
   do while Racf.Ø > Ø
     X = OUTTRAP("RACF.",'*',"NOCONCAT")
     Racf.Ø = Ø
     "Display panel(RacfPop4)"
     if Rc < 8 then do
       ADDRESS TSO "REMOVE " || RemUser || " GROUP(" || RemGroup || ")"
        if Racf.Ø > Ø then RacfMsg = Racf.1 || ", " || Racf.2
      end
   end
   "REMPOP   ALL"
      IF Rc ¬= Ø THEN say "RemPop error Rc("Rc")," zerrlm
end
lcmd = ""
"CONTROL DISPLAY RESTORE"
return

/*————————————————————————————————————
 * Display the pop-up panel to update the RACF access
 *————————————————————————————————————*/
Lst_User:
"CONTROL DISPLAY SAVE"
lcmd    = ""
"TBCREATE,RacfLstu,NAMES(LstuLine),
          NOWRITE,REPLACE"
parse value authline with LstuMask " " .
if substr(LstuMask,1,1) = "#" then LstuMask = substr(LstuMask,2)
LstuUser = " "
LstuName = " "
LstuGrp  = " "
X = OUTTRAP("RACF.",'*',"NOCONCAT")
ADDRESS TSO "LU " || LstuMask
DO I = 1 TO RACF.Ø
   LstuLine = strip(Racf.i)
```

```
   if substr(Lstuline,1,5) = "USER=" then do
      parse value LstuLine with "USER=" LstuUser " " . "NAME=" LstuName
   end
   if substr(Lstuline,1,7) = "DEFAULT" then do
      parse value lstuLine with "GROUP=" LstuGrp " " .
   end
   if substr(Lstuline,1,6) = "GROUP=" then "TbAdd RacfLstu"
END

"TbTop RacfLstu"
RcLstu  = Ø
DO WHILE RcLstu < 4
   "TbDispl RacfLstu PANEL(RACFLstu) position(LstuPos)"
   RcLstu = RC
   SlLstu = Ztdsels
   Ztdsels = SlLstu
   DO WHILE Ztdsels > 1 & RcLstu < 8
      "TBDISPL RacfLstu position(LstuPos)"
      RcLstu = RC
      SlLstu = Ztdsels
      Ztdsels = SlLstu
      END
   END
"TBEND,RacfLstu"
lcmd = ""
"CONTROL DISPLAY RESTORE"
return

/*————————————————————————————————————
 * Display the pop-up panel to update the RACF access
 *————————————————————————————————————*/
Lst_Group:
"CONTROL DISPLAY SAVE"
lcmd    = ""
"TBCREATE,RacfLstg,NAMES(LstgLine), NOWRITE,REPLACE"
parse value authline with LstgMask " " .
LstgSgrp = " "
LstgGrp  = " "
X = OUTTRAP("RACF.",'*',"NOCONCAT")
ADDRESS TSO "LG " || LstgMask
LstgFlag = Ø
DO I = 1 TO RACF.Ø
   LstgLine = strip(Racf.i)
   if substr(Lstgline,1,15) = "INFORMATION FOR" then do
      parse value LstgLine with "GROUP " LstgGrp " " .
   end
   if substr(Lstgline,1,Ø6) = "REVOKE" then iterate
   if substr(Lstgline,1,Ø7) = "CONNECT" then iterate
   if substr(Lstgline,1,Ø8) = "SUPERIOR" then do
      parse value LstgLine with . "GROUP=" LstgSgrp " " .
```

```
   end
   if LstgFlag = 1 then "TbAdd RacfLstg"
   if substr(LstgLine,1,Ø7) = "USER(S)" then LstgFlag = 1
END

"TbSort RacfLstg,fields(Lstgline,c,a)"
"TbTop RacfLstg"
RcLstg  = Ø
DO WHILE RcLstg < 4
   "TbDispl RacfLstg PANEL(RACFLstg) position(LstgPos)"
   RcLstg = RC
   SlLstg = Ztdsels
   if lcmd = "CON" & ztdsels > Ø then do
      ConUser = " "
      ConGroup = LstgGrp
      call Con_User
   end
   if lcmd = "REM" & ztdsels > Ø then do
      parse value LstgLine with RemUser " " .
      RemGroup = LstgGrp
      call Rem_User
   end
   Ztdsels = SlLstg
   DO WHILE Ztdsels > 1 & RcLstg < 8
      "TBDISPL RacfLstg position(LstgPos)"
      RcLstg = RC
      SlLstg = Ztdsels
      if lcmd = "CON" then do
         ConUser = " "
         ConGroup = LstgGrp
         call Con_User
      end
      if lcmd = "REM" then do
         parse value LstgLine with RemUser " " .
         RemGroup = LstgGrp
         call Rem_User
      end
      Ztdsels = SlLstg
      END
   END
"TBEND,RacfLstg"
lcmd = ""
"CONTROL DISPLAY RESTORE"
return

/*———————————————————————————————————————
   Put the name of DSN that contains the panels
  ———————————————————————————————————————*/
R_Open:
"LIBDEF ISPpLIB DATASET ID('your panel dsn')"
RETURN Ø
```

```
/*********************************************************************/
/* CLOSE ALL FILES AND TABLES                                        */
/*********************************************************************/
R_Close:
ADDRESS TSO "SETROPTS GENERIC(DATASET) REFRESH"
"LIBDEF ISPpLIB"
RETURN Ø
```

RACFDSNS PANEL

```
)ATTR
   + TYPE(TEXT) INTENS(LOW) SKIP(ON) CAPS(ON)
   % TYPE(TEXT) INTENS(HIGH) SKIP(ON) CAPS(ON)
   _ TYPE(input) INTENS(HIGH) CAPS(ON) just(left) pad(' ')
   # TYPE(input) INTENS(HIGH) CAPS(ON) just(left) pad('_')
   @ TYPE(output) INTENS(low)
)BODY WIDTH(&ZSCREENW) EXPAND(//)
%/-/%RACF Dataset List %/-/
%COMMAND ===>_ZCMD                                          %SCROLL
===>_AMT +
+
+Dataset Profile:%&DsnsMask
+
%PE  +Permit Access
%REM +Remove Access
+
+     Catalogued dataset(s) affected by profile
+- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -  -
)MODEL
#z   @dsnsline
)INIT
.CURSOR = ZCMD
.ZVARS = '(LCMD)'
&ZCMD = ' '
)REINIT
.CURSOR = ZCMD
)PROC
 VER (&LCMD,LIST,PE,REM)
 VPUT (AMT) PROFILE
)END
```

Editor's note: this article will be concluded in the next issue.

Salah Balboul
System Programmer (USA)

Replacement for the RACF Report Writer – part 5

This month we continue the code for the reporting tool in SAS which can act as a replacement for the RACF Report Writer. This enables you to have SQL-like capabilities without having to keep your SMF data in databases.

The article is an extensive piece of work which will be published over several issues of RACF Update. *To enable the article to be published in a manageable form, considerable editing of the original code has been necessary. The original, unedited code is available on our Web site (www.xephon.com) and can be downloaded in the usual way. This service is free to subscribers.*

```
      UNLFOUID = 'Owner UID'
      UNLFOGID = 'Owner GID'
      UNLLSTDL = 'Last link deleted?'
      UNLFILPL = 'File pool'
      UNLFILSP = 'File space'
      UNLINODE = 'Inode'
      UNLSCID  = 'File SCID'
      UNLDCELK = 'DCE link'
      UNLAUTYP = 'Request type'
      ;
        OUTPUT RACF.UNLINK;
      END;
    %END;
%MEND UNLINK;
./        ADD   LIST=ALL,NAME=UMNTFSYS
%MACRO UMNTFSYS(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
      %PUT Including variables from UMNTFSYS extension;
      RACF.UMNTFSYS (KEEP=%SMFHDR
                          %SMF8ØHDR(REQ=DEFINE)
                          UFSCLASS
                          UFSUSERN
                          UFSUTKNE
                          UFSUPRE
                          UFSUVFYX
                          UFSUNJEU
                          UFSUUAUD
                          UFSUSPEC
                          UFSUDFLT
                          UFSUUNDF
```

```
                        UFSUERR
                        UFSUTRST
                        UFSUSEST
                        UFSUSURO
                        UFSURMT
                        UFSUPRVL
                        UFSUSECL
                        UFSUEXND
                        UFSUSUSR
                        UFSUSNOD
                        UFSUSGRP
                        UFSUSPOE
                        UFSUSPCL
                        UFSUTUSR
                        UFSUTGRP
                        UFSUTDFT
                        UFSUTSEC
                        UFSAPPC
                        UFSAUDIT
                        UFSORUID
                        UFSOEUID
                        UFSOSUID
                        UFSORGID
                        UFSOEGID
                        UFSOSGID
                        UFSPATHN
                        UFSFILID
                        UFSFOUID
                        UFSFOGID
                        UFSHFSDS
                        UFSDCELK
                        UFSAUTYP
                    )
  %END;
%IF &REQ = EXTRACT %THEN
  %DO;
    %PUT Including datadefinition for UMNTFSYS extension;
    WHEN('UMNTFSYS') DO;
      INPUT %SMF80HDR(REQ=EXTRACT)
            UFSCLASS $       282-289
            UFSUSERN $       291-310
            UFSUTKNE $       312-315
            UFSUPRE  $       317-320
            UFSUVFYX $       322-325
            UFSUNJEU $       327-330
            UFSUUAUD $       332-335
            UFSUSPEC $       337-340
            UFSUDFLT $       342-345
            UFSUUNDF $       347-350
            UFSUERR  $       352-355
            UFSUTRST $       357-360
```

```
                UFSUSEST $       362-369
                UFSUSURO $       371-374
                UFSURMT  $       376-379
                UFSUPRVL $       381-384
                UFSUSECL $       386-393
                UFSUEXND $       395-402
                UFSUSUSR $       404-411
                UFSUSNOD $       413-420
                UFSUSGRP $       422-429
                UFSUSPOE $       431-438
                UFSUSPCL $       440-447
                UFSUTUSR $       449-456
                UFSUTGRP $       458-465
                UFSUTDFT $       467-470
                UFSUTSEC $       472-475
                UFSAPPC  $       477-492
                UFSAUDIT $       494-504
                UFSORUID         506-515
                UFSOEUID         517-526
                UFSOSUID         528-537
                UFSORGID         539-548
                UFSOEGID         550-559
                UFSOSGID         561-570
                UFSPATHN $       572-771
                UFSFILID $      1596-1627
                UFSFOUID        1629-1638
                UFSFOGID        1640-1649
                UFSHFSDS $      1651-1694
                UFSDCELK $      1696-1711
                UFSAUTYP $      1713-1725
                 ;
LABEL UFSCLASS = 'Class name'
      UFSUSERN = 'User name'
      UFSUTKNE = 'Utoken encr.?'
      UFSUPRE  = 'Pre-1.9?'
      UFSUVFYX = 'VERIFYX propagation?'
      UFSUNJEU = 'Undefined NJE user?'
      UFSUUAUD = 'UAUDIT?'
      UFSUSPEC = 'RACF special?'
      UFSUDFLT = 'Default token?'
      UFSUUNDF = 'Undefined user?'
      UFSUERR  = 'Token in error?'
      UFSUTRST = 'User trusted?'
      UFSUSEST = 'Session type'
      UFSUSURO = 'Surrogate user?'
      UFSURMT  = 'Remote job?'
      UFSUPRVL = 'Privileged user?'
      UFSUSECL = 'User SECLABEL'
      UFSUEXND = 'Execution node'
      UFSUSUSR = 'Submitting user'
      UFSUSNOD = 'Submitting node'
```

```
      UFSUSGRP = 'Submitting group'
      UFSUSPOE = 'Port of entry'
      UFSUSPCL = 'Class of POE'
      UFSUTUSR = 'Userid'
      UFSUTGRP = 'Groupid'
      UFSUTDFT = 'Default group?'
      UFSUTSEC = 'Default SECLABEL?'
      UFSAPPC  = 'APPC key link'
      UFSAUDIT = 'Audit code'
      UFSORUID = 'Old real UID'
      UFSOEUID = 'Old effective UID'
      UFSOSUID = 'Old saved UID'
      UFSORGID = 'Old real GID'
      UFSOEGID = 'Old effective GID'
      UFSOSGID = 'Old saved GID'
      UFSPATHN = 'Path name'
      UFSFILID = 'File id'
      UFSFOUID = 'Owner UID'
      UFSFOGID = 'Owner GID'
      UFSHFSDS = 'HFS datasetname'
      UFSDCELK = 'DCE link'
      UFSAUTYP = 'Request type'
      ;
        OUTPUT RACF.UMNTFSYS;
      END;
    %END;
%MEND UMNTFSYS;
./        ADD   LIST=ALL,NAME=CHKFOWN
%MACRO CHKFOWN(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
      %PUT Including variables from CHKFOWN extension;
      RACF.CHKFOWN (KEEP=%SMFHDR
                         %SMF8ØHDR(REQ=DEFINE)
                         CFOCLASS
                         CFOUSERN
                         CFOUTKNE
                         CFOUPRE
                         CFOUVFYX
                         CFOUNJEU
                         CFOUUAUD
                         CFOUSPEC
                         CFOUDFLT
                         CFOUUNDF
                         CFOUERR
                         CFOUTRST
                         CFOUSEST
                         CFOUSURO
                         CFOURMT
                         CFOUPRVL
```

```
                        CFOUSECL
                        CFOUEXND
                        CFOUSUSR
                        CFOUSNOD
                        CFOUSGRP
                        CFOUSPOE
                        CFOUSPCL
                        CFOUTUSR
                        CFOUTGRP
                        CFOUTDFT
                        CFOUTSEC
                        CFOAPPC
                        CFOAUDIT
                        CFOORUID
                        CFOOEUID
                        CFOOSUID
                        CFOORGID
                        CFOOEGID
                        CFOOSGID
                        CFOPATHN
                        CFOFILID
                        CFOFOUID
                        CFOFOGID
                        CFOFILPL
                        CFOFILSP
                        CFOINODE
                        CFOSCID
                        CFODCELK
                        CFOAUTYP
                    )
  %END;
%IF &REQ = EXTRACT %THEN
  %DO;
    %PUT Including datadefinition for CHKFOWN extension;
    WHEN('CHKFOWN') DO;
      INPUT %SMF80HDR(REQ=EXTRACT)
            CFOCLASS $       282-289
            CFOUSERN $       291-310
            CFOUTKNE $       312-315
            CFOUPRE  $       317-320
            CFOUVFYX $       322-325
            CFOUNJEU $       327-330
            CFOUUAUD $       332-335
            CFOUSPEC $       337-340
            CFOUDFLT $       342-345
            CFOUUNDF $       347-350
            CFOUERR  $       352-355
            CFOUTRST $       357-360
            CFOUSEST $       362-369
            CFOUSURO $       371-374
            CFOURMT  $       376-379
```

```
                CFOUPRVL $       381-384
                CFOUSECL $       386-393
                CFOUEXND $       395-402
                CFOUSUSR $       404-411
                CFOUSNOD $       413-420
                CFOUSGRP $       422-429
                CFOUSPOE $       431-438
                CFOUSPCL $       440-447
                CFOUTUSR $       449-456
                CFOUTGRP $       458-465
                CFOUTDFT $       467-470
                CFOUTSEC $       472-475
                CFOAPPC  $       477-492
                CFOAUDIT $       494-504
                CFOORUID         506-515
                CFOOEUID         517-526
                CFOOSUID         528-537
                CFOORGID         539-548
                CFOOEGID         550-559
                CFOOSGID         561-570
                CFOPATHN $       572-771
                CFOFILID $      1596-1627
                CFOFOUID        1629-1638
                CFOFOGID        1640-1649
                CFOFILPL $      1651-1658
                CFOFILSP $      1660-1667
                CFOINODE        1669-1678
                CFOSCID         1680-1689
                CFODCELK $      1691-1706
                CFOAUTYP $      1708-1720
                 ;
LABEL CFOCLASS = 'Class name'
      CFOUSERN = 'User name'
      CFOUTKNE = 'Utoken encr.?'
      CFOUPRE  = 'Pre-1.9?'
      CFOUVFYX = 'VERIFYX propagation?'
      CFOUNJEU = 'Undefined NJE user?'
      CFOUUAUD = 'UAUDIT?'
      CFOUSPEC = 'RACF special?'
      CFOUDFLT = 'Default token?'
      CFOUUNDF = 'Undefined user?'
      CFOUERR  = 'Token in error?'
      CFOUTRST = 'User trusted?'
      CFOUSEST = 'Session type'
      CFOUSURO = 'Surrogate user?'
      CFOURMT  = 'Remote job?'
      CFOUPRVL = 'Privileged user?'
      CFOUSECL = 'User SECLABEL'
      CFOUEXND = 'Execution node'
      CFOUSUSR = 'Submitting user'
      CFOUSNOD = 'Submitting node'
```

```
      CFOUSGRP = 'Submitting group'
      CFOUSPOE = 'Port of entry'
      CFOUSPCL = 'Class of POE'
      CFOUTUSR = 'Userid'
      CFOUTGRP = 'Groupid'
      CFOUTDFT = 'Default group?'
      CFOUTSEC = 'Default SECLABEL?'
      CFOAPPC  = 'APPC key link'
      CFOAUDIT = 'Audit code'
      CFOORUID = 'Old real UID'
      CFOOEUID = 'Old effective UID'
      CFOOSUID = 'Old saved UID'
      CFOORGID = 'Old real GID'
      CFOOEGID = 'Old effective GID'
      CFOOSGID = 'Old saved GID'
      CFOPATHN = 'Path name'
      CFOFILID = 'File id'
      CFOFOUID = 'Owner UID'
      CFOFOGID = 'Owner GID'
      CFOFILPL = 'File pool'
      CFOFILSP = 'File space'
      CFOINODE = 'Inode'
      CFOSCID  = 'File SCID'
      CFODCELK = 'DCE link'
      CFOAUTYP = 'Request type'
      ;
        OUTPUT RACF.CHKFOWN;
      END;
    %END;
%MEND CHKFOWN;
./        ADD   LIST=ALL,NAME=CHKPRIV
%MACRO CHKPRIV(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
      %PUT Including variables from CHKPRIV extension;
      RACF.CHKPRIV (KEEP=%SMFHDR
                         %SMF8ØHDR(REQ=DEFINE)
                         CPRCLASS
                         CPRUSERN
                         CPRUTKNE
                         CPRUPRE
                         CPRUVFYX
                         CPRUNJEU
                         CPRUUAUD
                         CPRUSPEC
                         CPRUDFLT
                         CPRUUNDF
                         CPRUERR
                         CPRUTRST
                         CPRUSEST
```

```
                        CPRUSURO
                        CPRURMT
                        CPRUPRVL
                        CPRUSECL
                        CPRUEXND
                        CPRUSUSR
                        CPRUSNOD
                        CPRUSGRP
                        CPRUSPOE
                        CPRUSPCL
                        CPRUTUSR
                        CPRUTGRP
                        CPRUTDFT
                        CPRUTSEC
                        CPRAPPC
                        CPRAUDIT
                        CPRORUID
                        CPROEUID
                        CPROSUID
                        CPRORGID
                        CPROEGID
                        CPROSGID
                        CPRDCELK
                        CPRAUTYP
                    )
  %END;
%IF &REQ = EXTRACT %THEN
  %DO;
    %PUT Including datadefinition for CHKPRIV extension;
    WHEN('CHKPRIV') DO;
      INPUT %SMF80HDR(REQ=EXTRACT)
            CPRCLASS $       282-289
            CPRUSERN $       291-310
            CPRUTKNE $       312-315
            CPRUPRE  $       317-320
            CPRUVFYX $       322-325
            CPRUNJEU $       327-330
            CPRUUAUD $       332-335
            CPRUSPEC $       337-340
            CPRUDFLT $       342-345
            CPRUUNDF $       347-350
            CPRUERR  $       352-355
            CPRUTRST $       357-360
            CPRUSEST $       362-369
            CPRUSURO $       371-374
            CPRURMT  $       376-379
            CPRUPRVL $       381-384
            CPRUSECL $       386-393
            CPRUEXND $       395-402
            CPRUSUSR $       404-411
```

```
                CPRUSNOD $       413-420
                CPRUSGRP $       422-429
                CPRUSPOE $       431-438
                CPRUSPCL $       440-447
                CPRUTUSR $       449-456
                CPRUTGRP $       458-465
                CPRUTDFT $       467-470
                CPRUTSEC $       472-475
                CPRAPPC  $       477-492
                CPRAUDIT $       494-504
                CPRORUID         506-515
                CPROEUID         517-526
                CPROSUID         528-537
                CPRORGID         539-548
                CPROEGID         550-559
                CPROSGID         561-570
                CPRDCELK $       572-587
                CPRAUTYP $       589-601
                 ;
LABEL CPRCLASS = 'Class name'
      CPRUSERN = 'User name'
      CPRUTKNE = 'Utoken encr.?'
      CPRUPRE  = 'Pre-1.9?'
      CPRUVFYX = 'VERIFYX propagation?'
      CPRUNJEU = 'Undefined NJE user?'
      CPRUUAUD = 'UAUDIT?'
      CPRUSPEC = 'RACF special?'
      CPRUDFLT = 'Default token?'
      CPRUUNDF = 'Undefined user?'
      CPRUERR  = 'Token in error?'
      CPRUTRST = 'User trusted?'
      CPRUSEST = 'Session type'
      CPRUSURO = 'Surrogate user?'
      CPRURMT  = 'Remote job?'
      CPRUPRVL = 'Privileged user?'
      CPRUSECL = 'User SECLABEL'
      CPRUEXND = 'Execution node'
      CPRUSUSR = 'Submitting user'
      CPRUSNOD = 'Submitting node'
      CPRUSGRP = 'Submitting group'
      CPRUSPOE = 'Port of entry'
      CPRUSPCL = 'Class of POE'
      CPRUTUSR = 'Userid'
      CPRUTGRP = 'Groupid'
      CPRUTDFT = 'Default group?'
      CPRUTSEC = 'Default SECLABEL?'
      CPRAPPC  = 'APPC key link'
      CPRAUDIT = 'Audit code'
      CPRORUID = 'Old real UID'
      CPROEUID = 'Old effective UID'
      CPROSUID = 'Old saved UID'
```

```
      CPRORGID = 'Old real GID'
      CPROEGID = 'Old effective GID'
      CPROSGID = 'Old saved GID'
      CPRDCELK = 'DCE link'
      CPRAUTYP = 'Request type'
      ;
        OUTPUT RACF.CHKPRIV;
      END;
    %END;
%MEND CHKPRIV;
./        ADD   LIST=ALL,NAME=OPENSTTY
%MACRO OPENSTTY(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
      %PUT Including variables from OPENSTTY extension;
      RACF.OPENSTTY (KEEP=%SMFHDR
                          %SMF8ØHDR(REQ=DEFINE)
                          OSTCLASS
                          OSTUSERN
                          OSTUTKNE
                          OSTUPRE
                          OSTUVFYX
                          OSTUNJEU
                          OSTUUAUD
                          OSTUSPEC
                          OSTUDFLT
                          OSTUUNDF
                          OSTUERR
                          OSTUTRST
                          OSTUSEST
                          OSTUSURO
                          OSTURMT
                          OSTUPRVL
                          OSTUSECL
                          OSTUEXND
                          OSTUSUSR
                          OSTUSNOD
                          OSTUSGRP
                          OSTUSPOE
                          OSTUSPCL
                          OSTUTUSR
                          OSTUTGRP
                          OSTUTDFT
                          OSTUTSEC
                          OSTAPPC
                          OSTAUDIT
                          OSTORUID
                          OSTOEUID
                          OSTOSUID
                          OSTORGID
```

```
                        OSTOEGID
                        OSTOSGID
                        OSTTRUID
                        OSTTEUID
                        OSTTSUID
                        OSTTPID
                    )
  %END;
%IF &REQ = EXTRACT %THEN
  %DO;
    %PUT Including datadefinition for OPENSTTY extension;
    WHEN('OPENSTTY') DO;
      INPUT %SMF80HDR(REQ=EXTRACT)
            OSTCLASS $       282-289
            OSTUSERN $       291-310
            OSTUTKNE $       312-315
            OSTUPRE  $       317-320
            OSTUVFYX $       322-325
            OSTUNJEU $       327-330
            OSTUUAUD $       332-335
            OSTUSPEC $       337-340
            OSTUDFLT $       342-345
            OSTUUNDF $       347-350
            OSTUERR  $       352-355
            OSTUTRST $       357-360
            OSTUSEST $       362-369
            OSTUSURO $       371-374
            OSTURMT  $       376-379
            OSTUPRVL $       381-384
            OSTUSECL $       386-393
            OSTUEXND $       395-402
            OSTUSUSR $       404-411
            OSTUSNOD $       413-420
            OSTUSGRP $       422-429
            OSTUSPOE $       431-438
            OSTUSPCL $       440-447
            OSTUTUSR $       449-456
            OSTUTGRP $       458-465
            OSTUTDFT $       467-470
            OSTUTSEC $       472-475
            OSTAPPC  $       477-492
            OSTAUDIT $       494-504
            OSTORUID         506-515
            OSTOEUID         517-526
            OSTOSUID         528-537
            OSTORGID         539-548
            OSTOEGID         550-559
            OSTOSGID         561-570
            OSTTRUID         572-581
            OSTTEUID         583-592
```

```
              OSTTSUID         594-6Ø3
              OSTTPID          6Ø5-614
               ;
LABEL OSTCLASS = 'Class name'
      OSTUSERN = 'User name'
      OSTUTKNE = 'Utoken encr.?'
      OSTUPRE  = 'Pre-1.9?'
      OSTUVFYX = 'VERIFYX propagation?'
      OSTUNJEU = 'Undefined NJE user?'
      OSTUUAUD = 'UAUDIT?'
      OSTUSPEC = 'RACF special?'
      OSTUDFLT = 'Default token?'
      OSTUUNDF = 'Undefined user?'
      OSTUERR  = 'Token in error?'
      OSTUTRST = 'User trusted?'
      OSTUSEST = 'Session type'
      OSTUSURO = 'Surrogate user?'
      OSTURMT  = 'Remote job?'
      OSTUPRVL = 'Privileged user?'
      OSTUSECL = 'User SECLABEL'
      OSTUEXND = 'Execution node'
      OSTUSUSR = 'Submitting user'
      OSTUSNOD = 'Submitting node'
      OSTUSGRP = 'Submitting group'
      OSTUSPOE = 'Port of entry'
      OSTUSPCL = 'Class of POE'
      OSTUTUSR = 'Userid'
      OSTUTGRP = 'Groupid'
      OSTUTDFT = 'Default group?'
      OSTUTSEC = 'Default SECLABEL?'
      OSTAPPC  = 'APPC key link'
      OSTAUDIT = 'Audit code'
      OSTORUID = 'Old real UID'
      OSTOEUID = 'Old effective UID'
      OSTOSUID = 'Old saved UID'
      OSTORGID = 'Old real GID'
      OSTOEGID = 'Old effective GID'
      OSTOSGID = 'Old saved GID'
      OSTTRUID = 'Tgt. real UID'
      OSTTEUID = 'Tgt. effective UID'
      OSTTSUID = 'Tgt. saved UID'
      OSTTPID  = 'Tgt. process ID'
      ;
        OUTPUT RACF.OPENSTTY;
      END;
    %END;
%MEND OPENSTTY;
./        ADD   LIST=ALL,NAME=RACLINK
%MACRO RACLINK(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
```

```
  %DO;
    %PUT Including variables from RACLINK extension;
    RACF.RACLINK (KEEP=%SMFHDR
                       %SMF8ØHDR(REQ=DEFINE)
                       RACUSERN
                       RACUTKNE
                       RACUPRE
                       RACUVFYX
                       RACUNJEU
                       RACUUAUD
                       RACUSPEC
                       RACUDFLT
                       RACUUNDF
                       RACUERR
                       RACUTRST
                       RACUSEST
                       RACUSURO
                       RACURMT
                       RACUPRVL
                       RACUSECL
                       RACUEXND
                       RACUSUSR
                       RACUSNOD
                       RACUSGRP
                       RACUSPOE
                       RACUSPCL
                       RACUTUSR
                       RACUTGRP
                       RACUTDFT
                       RACUTSEC
                       RACPHASE
                       RACISNOD
                       RACISID
                       RACSRCID
                       RACTGNOD
                       RACTGID
                       RACTGAID
                       RACSRSMF
                       RACSRTME
                       RACSRDTE
                       RACPWDST
                       RACASCST
                       RACSPEFD
                  )
  %END;
%IF &REQ = EXTRACT %THEN
  %DO;
    %PUT Including datadefinition for RACLINK extension;
    WHEN('RACLINK') DO;
      INPUT %SMF8ØHDR(REQ=EXTRACT)
            RACUSERN $       282-3Ø1
```

```
                RACUTKNE $       303-306
                RACUPRE  $       308-311
                RACUVFYX $       313-316
                RACUNJEU $       318-321
                RACUUAUD $       323-326
                RACUSPEC $       328-331
                RACUDFLT $       333-336
                RACUUNDF $       338-341
                RACUERR  $       343-346
                RACUTRST $       348-351
                RACUSEST $       353-360
                RACUSURO $       362-365
                RACURMT  $       367-370
                RACUPRVL $       372-375
                RACUSECL $       377-384
                RACUEXND $       386-493
                RACUSUSR $       395-402
                RACUSNOD $       404-411
                RACUSGRP $       413-420
                RACUSPOE $       422-429
                RACUSPCL $       431-438
                RACUTUSR $       440-447
                RACUTGRP $       449-456
                RACUTDFT $       458-461
                RACUTSEC $       463-466
                RACPHASE $       468-487
                RACISNOD $       489-496
                RACISID  $       498-505
                RACSRCID $       507-514
                RACTGNOD $       516-523
                RACTGID  $       525-532
                RACTGAID $       534-541
                RACSRSMF $       543-546
                RACSRTME $       548-555
                RACSRDTE $       557-566
                RACPWDST $       568-575
                RACASCST $       577-584
                RACSPEFD $       586-785
                ;
LABEL RACUSERN = 'User name'
      RACUTKNE = 'Utoken encr.?'
      RACUPRE  = 'Pre-1.9?'
      RACUVFYX = 'VERIFYX propagation?'
      RACUNJEU = 'Undefined NJE user?'
      RACUUAUD = 'UAUDIT?'
      RACUSPEC = 'RACF special?'
      RACUDFLT = 'Default token?'
      RACUUNDF = 'Undefined user?'
      RACUERR  = 'Token in error?'
      RACUTRST = 'User trusted?'
```

```
      RACUSEST = 'Session type'
      RACUSURO = 'Surrogate user?'
      RACURMT  = 'Remote job?'
      RACUPRVL = 'Privileged user?'
      RACUSECL = 'User SECLABEL'
      RACUEXND = 'Execution node'
      RACUSUSR = 'Submitting user'
      RACUSNOD = 'Submitting node'
      RACUSGRP = 'Submitting group'
      RACUSPOE = 'Port of entry'
      RACUSPCL = 'Class of POE'
      RACUTUSR = 'Userid'
      RACUTGRP = 'Groupid'
      RACUTDFT = 'Default group?'
      RACUTSEC = 'Default SECLABEL?'
      RACPHASE = 'Phase of command'
      RACISNOD = 'Originating node'
      RACISID  = 'Originating id'
      RACSRCID = 'Association id'
      RACTGNOD = 'Destination node'
      RACTGID  = 'Destination id'
      RACTGAID = 'Establishing id'
      RACSRSMF = 'Destination SMF id'
      RACSRTME = 'Time originated'
      RACSRDTE = 'Date originated'
      RACPWDST = 'Password status'
      RACASCST = 'Association status'
      RACSPEFD = 'Keywords specified'
      ;
        OUTPUT RACF.RACLINK;
      END;
    %END;
%MEND RACLINK;
./        ADD   LIST=ALL,NAME=IPCCHK
%MACRO IPCCHK(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
      %PUT Including variables from IPCCHK extension;
      RACF.IPCCHK (KEEP=%SMFHDR
                        %SMF8ØHDR(REQ=DEFINE)
                        ICHCLASS
                        ICHUSERN
                        ICHUTKNE
                        ICHUPRE
                        ICHUVFYX
                        ICHUNJEU
                        ICHUUAUD
                        ICHUSPEC
                        ICHUDFLT
                        ICHUUNDF
```

```
                        ICHUERR
                        ICHUTRST
                        ICHUSEST
                        ICHUSURO
                        ICHURMT
                        ICHUPRVL
                        ICHUSECL
                        ICHUEXND
                        ICHUSUSR
                        ICHUSNOD
                        ICHUSGRP
                        ICHUSPOE
                        ICHUSPCL
                        ICHUTUSR
                        ICHUTGRP
                        ICHUTDFT
                        ICHUTSEC
                        ICHAPPC
                        ICHAUDIT
                        ICHORUID
                        ICHOEUID
                        ICHOSUID
                        ICHORGID
                        ICHOEGID
                        ICHOSGID
                        ICHKOUID
                        ICHKOGID
                        ICHREQRE
                        ICHREQWR
                        ICHREQEX
                        ICHRSV1
                        ICHACTYP
                        ICHALWRE
                        ICHALWWR
                        ICHRSV2
                        ICHKEY
                        ICHID
                        ICHCRUID
                        ICHCRGID
                    )
  %END;
%IF &REQ = EXTRACT %THEN
  %DO;
    %PUT Including datadefinition for IPCCHK extension;
    WHEN('IPCCHK') DO;
      INPUT %SMF8ØHDR(REQ=EXTRACT)
            ICHCLASS $       282-289
            ICHUSERN $       291-31Ø
            ICHUTKNE $       312-315
            ICHUPRE  $       317-32Ø
```

```
                ICHUVFYX $       322-325
                ICHUNJEU $       327-330
                ICHUUAUD $       332-335
                ICHUSPEC $       337-340
                ICHUDFLT $       342-345
                ICHUUNDF $       347-350
                ICHUERR  $       352-355
                ICHUTRST $       357-360
                ICHUSEST $       362-369
                ICHUSURO $       371-374
                ICHURMT  $       376-379
                ICHUPRVL $       381-384
                ICHUSECL $       386-393
                ICHUEXND $       395-402
                ICHUSUSR $       404-411
                ICHUSNOD $       413-420
                ICHUSGRP $       422-429
                ICHUSPOE $       431-438
                ICHUSPCL $       440-447
                ICHUTUSR $       449-456
                ICHUTGRP $       458-465
                ICHUTDFT $       467-470
                ICHUTSEC $       472-475
                ICHAPPC  $       477-492
                ICHAUDIT $       494-504
                ICHORUID         506-515
                ICHOEUID         517-526
                ICHOSUID         528-537
                ICHORGID         539-548
                ICHOEGID         550-559
                ICHOSGID         561-570
                ICHKOUID         572-581
                ICHKOGID         583-592
                ICHREQRE $       594-597
                ICHREQWR $       599-602
                ICHREQEX $       604-607
                ICHRSV1  $       609-612
                ICHACTYP $       614-621
                ICHALWRE $       623-626
                ICHALWWR $       628-631
                ICHRSV2  $       633-636
                ICHKEY   $       638-645
                ICHID            647-656
                ICHCRUID         658-667
                ICHCRGID         669-678
                 ;
LABEL ICHCLASS = 'Class name'
      ICHUSERN = 'User name'
      ICHUTKNE = 'Utoken encr.?'
      ICHUPRE  = 'Pre-1.9?'
```

```
      ICHUVFYX = 'VERIFYX propagation?'
      ICHUNJEU = 'Undefined NJE user?'
      ICHUUAUD = 'UAUDIT?'
      ICHUSPEC = 'RACF special?'
      ICHUDFLT = 'Default token?'
      ICHUUNDF = 'Undefined user?'
      ICHUERR  = 'Token in error?'
      ICHUTRST = 'User trusted?'
      ICHUSEST = 'Session type'
      ICHUSURO = 'Surrogate user?'
      ICHURMT  = 'Remote job?'
      ICHUPRVL = 'Privileged user?'
      ICHUSECL = 'User SECLABEL'
      ICHUEXND = 'Execution node'
      ICHUSUSR = 'Submitting user'
      ICHUSNOD = 'Submitting node'
      ICHUSGRP = 'Submitting group'
      ICHUSPOE = 'Port of entry'
      ICHUSPCL = 'Class of POE'
      ICHUTUSR = 'Userid'
      ICHUTGRP = 'Groupid'
      ICHUTDFT = 'Default group?'
      ICHUTSEC = 'Default SECLABEL?'
      ICHAPPC  = 'APPC key link'
      ICHAUDIT = 'Audit code'
      ICHORUID = 'Old real UID'
      ICHOEUID = 'Old effective UID'
      ICHOSUID = 'Old saved UID'
      ICHORGID = 'Old real GID'
      ICHOEGID = 'Old effective GID'
      ICHOSGID = 'Old saved GID'
      ICHKOUID = 'Key owner UID'
      ICHKOGID = 'Key owner GID'
      ICHREQRE = 'Read req?'
      ICHREQWR = 'Write req?'
      ICHREQEX = 'Exec req?'
      ICHRSV1  = 'Reserved'
      ICHACTYP = 'Access type'
      ICHALWRE = 'Read allowed?'
      ICHALWWR = 'Write allowed?'
      ICHRSV2  = 'Reserved'
      ICHKEY   = 'IPC key'
      ICHID    = 'IPC unique id'
      ICHCRUID = 'Creator UID'
      ICHCRGID = 'Creator GID'
      ;
        OUTPUT RACF.IPCCHK;
      END;
    %END;
%MEND IPCCHK;
```

```
./        ADD   LIST=ALL,NAME=IPCGET
%MACRO IPCGET(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
      %PUT Including variables from IPCGET extension;
      RACF.IPCGET (KEEP=%SMFHDR
                        %SMF8ØHDR(REQ=DEFINE)
                        IGTCLASS
                        IGTUSERN
                        IGTUTKNE
                        IGTUPRE
                        IGTUVFYX
                        IGTUNJEU
                        IGTUUAUD
                        IGTUSPEC
                        IGTUDFLT
                        IGTUUNDF
                        IGTUERR
                        IGTUTRST
                        IGTUSEST
                        IGTUSURO
                        IGTURMT
                        IGTUPRVL
                        IGTUSECL
                        IGTUEXND
                        IGTUSUSR
                        IGTUSNOD
                        IGTUSGRP
                        IGTUSPOE
                        IGTUSPCL
                        IGTUTUSR
                        IGTUTGRP
                        IGTUTDFT
                        IGTUTSEC
                        IGTAPPC
                        IGTAUDIT
                        IGTORUID
                        IGTOEUID
                        IGTOSUID
                        IGTORGID
                        IGTOEGID
                        IGTOSGID
                        IGTKOUID
                        IGTKOGID
                        IGTRSV1
                        IGTRSV2
                        IGTRSV3
                        IGTRQORD
                        IGTRQOWR
```

```
                        IGTRQOEX
                        IGTRQGRD
                        IGTRQGWR
                        IGTRQGEX
                        IGTRQWRD
                        IGTRQWWR
                        IGTRQWEX
                        IGTKEY
                        IGTID
                        IGTCRUID
                        IGTCRGID
                    )
  %END;
%IF &REQ = EXTRACT %THEN
  %DO;
    %PUT Including datadefinition for IPCGET extension;
    WHEN('IPCGET') DO;
      INPUT %SMF8ØHDR(REQ=EXTRACT)
            IGTCLASS $       282-289
            IGTUSERN $       291-31Ø
            IGTUTKNE $       312-315
            IGTUPRE  $       317-32Ø
            IGTUVFYX $       322-325
            IGTUNJEU $       327-33Ø
            IGTUUAUD $       332-335
            IGTUSPEC $       337-34Ø
            IGTUDFLT $       342-345
            IGTUUNDF $       347-35Ø
            IGTUERR  $       352-355
            IGTUTRST $       357-36Ø
            IGTUSEST $       362-369
            IGTUSURO $       371-374
            IGTURMT  $       376-379
            IGTUPRVL $       381-384
            IGTUSECL $       386-393
            IGTUEXND $       395-4Ø2
            IGTUSUSR $       4Ø4-411
            IGTUSNOD $       413-42Ø
            IGTUSGRP $       422-429
            IGTUSPOE $       431-438
            IGTUSPCL $       44Ø-447
            IGTUTUSR $       449-456
            IGTUTGRP $       458-465
            IGTUTDFT $       467-47Ø
            IGTUTSEC $       472-475
            IGTAPPC  $       477-492
            IGTAUDIT $       494-5Ø4
            IGTORUID         5Ø6-515
            IGTOEUID         517-526
            IGTOSUID         528-537
```

```
                IGTORGID         539-548
                IGTOEGID         55Ø-559
                IGTOSGID         561-57Ø
                IGTKOUID         572-581
                IGTKOGID         583-592
                IGTRSV1  $       594-597
                IGTRSV2  $       599-6Ø2
                IGTRSV3  $       6Ø4-6Ø7
                IGTRQORD $       6Ø9-612
                IGTRQOWR $       614-617
                IGTRQOEX $       619-622
                IGTRQGRD $       624-627
                IGTRQGWR $       629-632
                IGTRQGEX $       634-637
                IGTRQWRD $       639-642
                IGTRQWWR $       644-647
                IGTRQWEX $       649-652
                IGTKEY   $       654-661
                IGTID            663-672
                IGTCRUID         674-683
                IGTCRGID         685-694
                 ;
LABEL IGTCLASS = 'Class name'
      IGTUSERN = 'User name'
      IGTUTKNE = 'Utoken encr.?'
      IGTUPRE  = 'Pre-1.9?'
      IGTUVFYX = 'VERIFYX propagation?'
      IGTUNJEU = 'Undefined NJE user?'
      IGTUUAUD = 'UAUDIT?'
      IGTUSPEC = 'RACF special?'
      IGTUDFLT = 'Default token?'
      IGTUUNDF = 'Undefined user?'
      IGTUERR  = 'Token in error?'
      IGTUTRST = 'User trusted?'
      IGTUSEST = 'Session type'
      IGTUSURO = 'Surrogate user?'
      IGTURMT  = 'Remote job?'
      IGTUPRVL = 'Privileged user?'
      IGTUSECL = 'User SECLABEL'
      IGTUEXND = 'Execution node'
      IGTUSUSR = 'Submitting user'
      IGTUSNOD = 'Submitting node'
      IGTUSGRP = 'Submitting group'
      IGTUSPOE = 'Port of entry'
      IGTUSPCL = 'Class of POE'
      IGTUTUSR = 'Userid'
      IGTUTGRP = 'Groupid'
      IGTUTDFT = 'Default group?'
      IGTUTSEC = 'Default SECLABEL?'
      IGTAPPC  = 'APPC key link'
```

```
      IGTAUDIT = 'Audit code'
      IGTORUID = 'Old real UID'
      IGTOEUID = 'Old effective UID'
      IGTOSUID = 'Old saved UID'
      IGTORGID = 'Old real GID'
      IGTOEGID = 'Old effective GID'
      IGTOSGID = 'Old saved GID'
      IGTKOUID = 'Key owner UID'
      IGTKOGID = 'Key owner GID'
      IGTRSV1  = 'Reserved'
      IGTRSV2  = 'Reserved'
      IGTRSV3  = 'Reserved'
      IGTRQORD = 'Owner read req ?'
      IGTRQOWR = 'Owner write req ?'
      IGTRQOEX = 'Owner exec req ?'
      IGTRQGRD = 'Group read req ?'
      IGTRQGWR = 'Group write req ?'
      IGTRQGEX = 'Group exec req ?'
      IGTRQWRD = 'Other read req ?'
      IGTRQWWR = 'Other write req ?'
      IGTRQWEX = 'Other exec req ?'
      IGTKEY   = 'IPC key'
      IGTID    = 'IPC unique id'
      IGTCRUID = 'Creator UID'
      IGTCRGID = 'Creator GID'
      ;
        OUTPUT RACF.IPCGET;
      END;
    %END;
%MEND IPCGET;
./        ADD   LIST=ALL,NAME=IPCCTL
%MACRO IPCCTL(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
      %PUT Including variables from IPCCTL extension;
      RACF.IPCCTL (KEEP=%SMFHDR
                        %SMF8ØHDR(REQ=DEFINE)
                        ICTCLASS
                        ICTUSERN
                        ICTUTKNE
                        ICTUPRE
                        ICTUVFYX
                        ICTUNJEU
                        ICTUUAUD
                        ICTUSPEC
                        ICTUDFLT
                        ICTUUNDF
                        ICTUERR
                        ICTUTRST
```

```
ICTUSEST
ICTUSURO
ICTURMT
ICTUPRVL
ICTUSECL
ICTUEXND
ICTUSUSR
ICTUSNOD
ICTUSGRP
ICTUSPOE
ICTUSPCL
ICTUTUSR
ICTUTGRP
ICTUTDFT
ICTUTSEC
ICTAPPC
ICTAUDIT
ICTORUID
ICTOEUID
ICTOSUID
ICTORGID
ICTOEGID
ICTOSGID
ICTKOUID
ICTKOGID
ICTUID
ICTGID
ICTRSV1
ICTRSV2
ICTRSV3
ICTOLORD
ICTOLOWR
ICTOLOEX
ICTOLGRD
ICTOLGWR
ICTOLGEX
ICTOLWRD
ICTOLWWR
ICTOLWEX
ICTRSV4
ICTRSV5
ICTRSV6
ICTNWORD
ICTNWOWR
ICTNWOEX
ICTNWGRD
ICTNWGWR
ICTNWGEX
ICTNWWRD
ICTNWWWR
```

```
ICTNWWEX
ICTSRVCD
ICTRSV7
ICTRSV8
ICTRSV9
ICTRQORD
ICTRQOWR
ICTRQOEX
ICTRQGRD
ICTRQGWR
ICTRQGEX
ICTRQWRD
ICTRQWWR
ICTRQWEX
ICTKEY
ICTID
ICTCRUID
ICTCRGID
```

Editor's note: this article will be concluded in the next issue.

Cross-checking RACF/MVS definitions – part 2

This month we conclude the article on cross-checking the interdependency of certain MVS and RACF definitions.

FROM A RACF USER POINT OF VIEW

This routine will look at the definitions from the point of view of a RACF user:

```
//STEP1    EXEC PGM=IEBGENER  GENERATE CLIST
//SYSIN    DD   DUMMY
//SYSPRINT DD   SYSOUT=*
//SYSUT2   DD   DSN=&&TEMP(CLIST),DISP=(,PASS),SPACE=(TRK,(1,1,1)),
// UNIT=VIO
//SYSUT1   DD   *
  /*
  /*  CHECK THAT USER FROM RACF HAS APPROPRIATE
  /*  RACF/ALIAS/DATASET/NOT IN UADS/ DEFINITIONS
  /*
  /*  Parameters:
  /*   RC1 - RCn: mail receivers
  /*
PROC Ø ID('USER RACF CHECK') DEBUG(NEBUG) RC1(MAILUS1) RC2() +
RC3() RC4() RC5() RC6() RC7() RC8()
CONTROL   MSG NOFLUSH NOLIST NOCONLIST NOSYMLIST
ATTN DO
 SET &FLUSH = FLUSH         /* NEXT STATEMENT MUST BE NULL LINE     */

END
ERROR DO
 SET &RET = &LASTCC
 RETURN
END
IF &FLUSH = FLUSH THEN DO
 EXIT CODE(Ø)
END
IF &STR(&DEBUG) = DEBUG THEN DO
 CONTROL MSG NOFLUSH LIST CONLIST SYMLIST
END
OPENFILE OUT OUTPUT
SET OUT =  USER STATUS FROM A RACF POINT OF VIEW
PUTFILE OUT
IF &SYSPREF.. = . THEN DO
 PROFILE PREFIX(INSTPREF)           /* Default prefix */
END
```

```
ELSE DO
 PROFILE PREFIX(&SYSUID)
END
SET &TSTAMP = +
&STR(T)&SUBSTR(1:2,&SYSTIME)&SUBSTR(4:5,&SYSTIME)&SUBSTR(7:8,&SYSTIME)
DEL '&SYSPREF..&TSTAMP..TEMP.LIST'
SET &CNT = Ø
SET &RET = Ø
ALLOC FI(LIST) DA('&SYSPREF..&TSTAMP..TEMP.LIST') +
NEW SPACE(1 1) CYLINDERS +
STORCLAS(TEMP) MGMTCLAS(TEMP) RECFM(F B) LRECL(8Ø) BLKSIZE(2792Ø) REUSE
DO WHILE &RET NE Ø AND &CNT < 3ØØ THEN DO
 SET &TSTAMP = +
 &STR(T)&SUBSTR(1:2,&SYSTIME)&SUBSTR(4:5,&SYSTIME)&SUBSTR(7:8,&SYSTIME)
 DEL '&SYSPREF..&TSTAMP..TEMP.LIST'
 SET &RET = Ø
 ALLOC FI(LIST) DA('&SYSPREF..&TSTAMP..TEMP.LIST') +
 NEW SPACE(1 1) CYLINDERS +
 STORCLAS(TEMP) MGMTCLAS(TEMP) RECFM(F B) LRECL(8Ø) BLKSIZE(2792Ø) REUSE
 IF &RET NE Ø THEN DO
  FREE DA('&SYSPREF..&TSTAMP..TEMP.LIST')
 END
 SET &CNT = &CNT + 1
 SLEEP 15
END
SET &PREFIX = &SYSPREF
OPENFILE LIST OUTPUT
SET LIST =  USER STATUS FROM A RACF POINT OF VIEW
PUTFILE LIST
SET &RP = &STR())
PROFILE NOPREFIX
SET &SYSOUTTRAP = 999999
SEARCH ALL CLASS(USER) NOMASK LIST
SET &SYSOUTTRAP = Ø
SET &MAXLINE = &SYSOUTLINE
SET &USER = USER
SET &P = Ø
SET &RET = Ø
DO WHILE &P < &MAXLINE
 SET &P = &P + 1
 SET &SYSDVAL = &STR(&&SYSOUTLINE&P)
 SET &SYSDVAL = &STR(&SYSDVAL)
 READDVAL &A1 &A2 &A3 &A4 &A5 &A6 &A7 &A8
 SET &&USER&P = &STR(&A1)
END
SET &P = Ø
SET &RET = Ø
DO WHILE &P < &MAXLINE
 SET &P = &P + 1
 SET &A1 = &STR(&&USER&P)
```

```
IF &STR(&A1) NE &STR() THEN DO
 WRITE
 WRITE &STR(========>) INFORMATION FOR USER &STR("&A1")
 SET &HLQ = NO
 SET &DS = NO
 SET &RACFU = NO
 SET &RACFT = NO
 SET &GROUP = NO
 SET &DSP = NO
 SET &UADS = NO
 SET &SYSOUTTRAP = 99999
 SET &RET = Ø
 LISTC ALL ENT('&A1'&RP
 SET &SYSOUTTRAP = Ø
 IF &RET = Ø THEN DO
  SET &HLQ = YES
  SET &SYSOUTTRAP = 99999
  SET &RET = Ø
  LISTC     LEVEL(&A1&RP
  SET &SYSOUTTRAP = Ø
  IF &RET = Ø THEN DO
   SET &DS = YES
   WRITE SOME DATASETS/ENTRIES DO EXIST UNDER ALIAS &STR("&A1")
  END
  ELSE DO
   WRITE NO DATASETS/ENTRIES EXIST UNDER ALIAS &STR("&A1")
  END
 END
 ELSE DO
  WRITE NO ALIAS EXISTS FOR &STR("&A1") ANY LONGER
 END
 SET &RET = Ø
 LG &A1
 IF &RET = Ø THEN DO
  SET &GROUP = YES
 END
 ELSE DO
  WRITE NO RACF GROUP EXISTS FOR &STR("&A1")
  CONTROL NOMSG
  SET &SYSOUTTRAP = 999999
  SET &RET = Ø
  LU &A1
  SET &SYSOUTTRAP = Ø
  SET &LURET = &RET
  CONTROL MSG
  IF &LURET < 4 THEN DO
   SET &RACFU = YES
   WRITE "&A1" USER EXISTS IN RACF
   SET &SYSOUTTRAP = 999999
   LU &A1 TSO NORACF
```

```
  SET &SYSOUTTRAP = Ø
  SET &MAXTSO = &SYSOUTLINE
  SET &RACFTSO = &STR()
  SET &N = Ø
  SET &RET = Ø
  DO WHILE &N < &MAXTSO
   SET &N = &N + 1
   SET &SYSDVAL = &STR(&&SYSOUTLINE&N)
   SET &SYSDVAL = &STR(&SYSDVAL)
   READDVAL &B1 &B2 &B3 &B4 &B5 &B6 &B7 &B8
   IF &STR(&B1) = NO AND &STR(&B2) = TSO AND &STR(&B3) = INFORMATION +
   THEN DO
    WRITE "&A1" IS NOT DEFINED AS TSO USER IN RACF
    SET &RACFTSO = NO
   END
  END
  IF &RACFTSO NE NO THEN DO
   SET &RACFT = YES
   WRITE "&A1" IS DEFINED AS TSO USER IN RACF
  END
 END
 ELSE DO
  WRITE "&A1" DOES NOT EXIST AS USER IN RACF
 END
END
SET &RET = Ø
SET &MEMBER = &A1
SET &MEMBER = &MEMBER.&STR(.)
SEARCH ALL CLASS(DATASET) MASK(&MEMBER) LIST
IF &RET > Ø THEN DO
 WRITE NO RACF DATASET PROFILES EXIST FOR &STR("&A1")
END
ELSE DO
 SET &DSP = YES
END
SET &MEMBER = &A1
SET &MEMBER = &MEMBER.Ø
IF &LENGTH(&STR(&A1)) < 8 THEN DO
 SET &UADS = &STR(&SYSDSN('SYS1.UADS(&MEMBER)'))
 WRITE TSO DEFINITION IN SYS1.UADS = &UADS FOR &STR("&A1")
 IF &STR(&UADS) = OK THEN DO
  SET &UADS = YES
 END
END
ELSE DO
 WRITE TSO DEFINITION IN SYS1.UADS = MEMBER NOT FOUND FOR &STR("&A1")
END
IF &RACFU NE YES THEN DO
 WRITE ===> LOGICAL ERROR: USER &STR("&A1") MISSING
 SET &OUT = &STR(LOGICAL ERROR: USER &STR("&A1") MISSING)
```

```
 PUTFILE OUT
 SET &LIST = &STR(&OUT)
 PUTFILE LIST
 WTP &STR(&OUT)
 SET &ERROR = YES
END
IF &RACFT NE YES THEN DO
 IF &UADS = YES THEN DO
  WRITE ===> ERROR: &STR("&A1") TSO DEFINED IN UADS, SHOULD BE IN RACF
  SET &OUT = +
  &STR(ERROR: &STR("&A1") TSO DEFINED IN UADS, SHOULD BE IN RACF)
  PUTFILE OUT
  SET &LIST = &STR(&OUT)
  PUTFILE LIST
  WTP &STR(&OUT)
  SET &ERROR = YES
 END
 IF &HLQ = YES THEN DO
  /* Substitute the following names, USPFn, with users without TSO. */
  /* Segment allowed to have HLQ defined, eg certain STCs           */
  IF &STR(&A1) NE USPF1   AND &STR(&A1) NE USPF2   AND +
  &STR(&A1) NE USPF3   AND &STR(&A1) NE USPF4   +
  THEN DO
   WRITE ===> +
   ERROR: ALIAS EXISTS FOR NON-TSO USER &STR("&A1")
   SET &OUT = +
   &STR(ERROR: ALIAS EXISTS FOR NON-TSO USER &STR("&A1"))
   PUTFILE OUT
   SET &LIST = &STR(&OUT)
   PUTFILE LIST
   WTP &STR(&OUT)
   SET &ERROR = YES
  END
 END
 IF &DS = YES THEN DO
  /* Substitute the following names, USPFn, with users without TSO. */
  /* Segment allowed to have datasets defined, eg certain STCs      */
  IF &STR(&A1) NE USPF1   AND &STR(&A1) NE USPF2   AND +
  &STR(&A1) NE USPF3   AND &STR(&A1) NE USPF4   +
  THEN DO
   WRITE ===> +
   ERROR: DATASET EXISTS FOR NON-TSO USER &STR("&A1")
   SET &OUT = +
   &STR(ERROR: DATASET EXISTS FOR NON-TSO USER &STR("&A1"))
   PUTFILE OUT
   SET &LIST = &STR(&OUT)
   PUTFILE LIST
   WTP &STR(&OUT)
   SET &ERROR = YES
  END
```

```
 END
 IF &DSP = YES THEN DO
  /* Substitute the following names, USPFn, with users without TSO. */
  /* Segment allowed to have DS profiles defined, eg certain STCs   */
  IF &STR(&A1) NE USPF1   AND &STR(&A1) NE USPF2   AND +
  &STR(&A1) NE USPF3   AND &STR(&A1) NE USPF4   +
  THEN DO
   WRITE ===> +
   ERROR: DATASET PROFILE EXISTS FOR NON-TSO USER &STR("&A1")
   SET &OUT = +
   &STR(ERROR: DATASET PROFILE EXISTS FOR NON-TSO USER +
   &STR("&A1"))
   PUTFILE OUT
   SET &LIST = &STR(&OUT)
   PUTFILE LIST
   WTP &STR(&OUT)
   SET &ERROR = YES
  END
 END
 IF &DSP NE YES THEN DO
  IF &DS = YES THEN DO
   WRITE ===> ERROR: DATASET EXISTS FOR NON-TSO USER &STR("&A1") +
   WITHOUT RACF PROFILE
   SET &OUT = +
   &STR(ERROR: DATASET EXISTS FOR NON-TSO USER &STR("&A1") +
   WITHOUT RACF PROFILE)
   PUTFILE OUT
   SET &LIST = &STR(&OUT)
   PUTFILE LIST
   WTP &STR(&OUT)
   SET &ERROR = YES
  END
 END
END
IF &RACFT = YES THEN DO
 IF &HLQ NE YES THEN DO
  WRITE ===> ERROR: ALIAS MISSING FOR USER &STR("&A1")
  SET &OUT = &STR(ERROR: ALIAS MISSING FOR USER &STR("&A1"))
  PUTFILE OUT
  SET &LIST = &STR(&OUT)
  PUTFILE LIST
  WTP &STR(&OUT)
  SET &ERROR = YES
 END
 IF &DS NE YES THEN DO
  WRITE ===> POSSIBLE ERROR: NO DATASET FOR USER &STR("&A1")
  SET &OUT = &STR(POSSIBLE ERROR: NO DATASET FOR TSO USER &STR("&A1"))
  PUTFILE OUT
 END
 IF &DSP NE YES THEN DO
```

```
    WRITE ===> ERROR: NO DATASET RACF PROFILE FOR USER &STR("&A1")
    SET &OUT = &STR(ERROR: NO DATASET RACF PROFILE FOR USER &STR("&A1"))
    PUTFILE OUT
    SET &LIST = &STR(&OUT)
    PUTFILE LIST
    WTP &STR(&OUT)
    SET &ERROR = YES
    IF &DS = YES THEN DO
     WRITE ===>        BUT DATASETS EXIST.
     SET &OUT = &STR(        BUT DATASETS EXIST.)
     PUTFILE OUT
     SET &LIST = &STR(&OUT)
     PUTFILE LIST
     WTP &STR(&OUT)
    END
   END
   IF &UADS = YES THEN DO
    /* The following users are allowed in UADs for emergency purposes */
    IF &STR(&A1) NE USERØ1 AND +
    &STR(&A1) NE USERØ2  AND +
    &STR(&A1) NE USERØ3 AND +
    &STR(&A1) NE USERØ4 THEN DO
     WRITE ===> ERROR: &STR("&A1") TSO USER BOTH IN RACF AND UADS
     SET &OUT = &STR(ERROR: &STR("&A1") TSO USER BOTH IN RACF AND UADS)
     PUTFILE OUT
     SET &LIST = &STR(&OUT)
     PUTFILE LIST
     WTP &STR(&OUT)
     SET &ERROR = YES
    END
   END
  END
 END
END
CLOSFILE OUT
CLOSFILE LIST
IF &ERROR = YES THEN DO
 SET &RC = RC
 SET &N = 1
 SET &RC = &&RC&N
 DO WHILE &RC NE &STR() AND &N < 9
  %MAILSENS ID('&ID') RC('&RC') DS(&PREFIX..&TSTAMP..TEMP.LIST) +
  DEBUG(&DEBUG) RCDOMAIN(YES) /* INFORM ERROR STATUS VIA MAIL */
  SET &N = &N + 1
  SET &RC = &&RC&N
 END
END
FREE DA('&PREFIX..&TSTAMP..TEMP.LIST')
DEL '&PREFIX..&TSTAMP..TEMP.LIST'
  /*
```

```
//STEP2    EXEC PGM=IKJEFTØ1,DYNAMNBR=128,  EXECUTE CLIST
// PARM='%CLIST DEBUG(NEBUG)'
//SYSTSPRT DD   SYSOUT=*
//SYSPRINT DD   SYSOUT=*
//OUT      DD   SYSOUT=*
//SYSPROC  DD   DSN=&TEMP,DISP=(OLD,PASS)
//         DD   DSN=SYSTEM.CLIST,DISP=SHR
//SYSTSIN  DD   DUMMY
//SYSIN    DD   DUMMY
//*
```

FROM A SYS1.UADS POINT OF VIEW

This routine looks at the definitions from the point of view of SYS1.UADs:

```
//STEP1    EXEC PGM=IEBGENER  GENERATE CLIST
//SYSIN    DD   DUMMY
//SYSPRINT DD   SYSOUT=*
//SYSUT2   DD   DSN=&&TEMP(CLIST),DISP=(,PASS),SPACE=(TRK,(1,1,1)),
// UNIT=VIO
//SYSUT1   DD   *
  /*
  /*  CHECK THAT USER FROM UADS HAS APPROPRIATE
  /*  RACF/ALIAS/DATASET DEFINITIONS.
  /*  NON-AUTHORIZED USERS DEFINED IN UADS WILL BE DELETED FROM UADS
  /*
  /*  Parameters:
  /*   RC1 - RCn: mail receivers
  /*
PROC Ø ID('UADS RACF CHECK') DEBUG(NEBUG) RC1(MAILUS1) RC2() +
RC3() RC4() RC5() RC6() RC7() RC8()
CONTROL   MSG NOFLUSH NOLIST NOCONLIST NOSYMLIST
ATTN DO
 SET &FLUSH = FLUSH         /* NEXT STATEMENT MUST BE NULL LINE     */

END
ERROR DO
 SET &RET = &LASTCC
 RETURN
END
IF &FLUSH = FLUSH THEN DO
 EXIT CODE(Ø)
END
IF &STR(&DEBUG) = DEBUG THEN DO
 CONTROL MSG NOFLUSH LIST CONLIST SYMLIST
END
OPENFILE OUT OUTPUT
SET OUT =  STATUS FROM SYS1.UADS POINT OF VIEW
```

```
PUTFILE OUT
IF &SYSPREF.. = . THEN DO
 PROFILE PREFIX(INSTPREF)           /* Default prefix */
END
ELSE DO
 PROFILE PREFIX(&SYSUID)
END
SET &TSTAMP = +
&STR(T)&SUBSTR(1:2,&SYSTIME)&SUBSTR(4:5,&SYSTIME)&SUBSTR(7:8,&SYSTIME)
DEL '&SYSPREF..&TSTAMP..TEMP.LIST'
SET &CNT = Ø
SET &RET = Ø
ALLOC FI(LIST) DA('&SYSPREF..&TSTAMP..TEMP.LIST') +
NEW SPACE(1 1) CYLINDERS +
STORCLAS(TEMP) MGMTCLAS(TEMP) RECFM(F B) LRECL(8Ø) BLKSIZE(2792Ø) REUSE
DO WHILE &RET NE Ø AND &CNT < 3ØØ THEN DO
 SET &TSTAMP = +
 &STR(T)&SUBSTR(1:2,&SYSTIME)&SUBSTR(4:5,&SYSTIME)&SUBSTR(7:8,&SYSTIME)
 DEL '&SYSPREF..&TSTAMP..TEMP.LIST'
 SET &RET = Ø
 ALLOC FI(LIST) DA('&SYSPREF..&TSTAMP..TEMP.LIST') +
 NEW SPACE(1 1) CYLINDERS +
 STORCLAS(TEMP) MGMTCLAS(TEMP) RECFM(F B) LRECL(8Ø) BLKSIZE(2792Ø) REUSE
 IF &RET NE Ø THEN DO
  FREE DA('&SYSPREF..&TSTAMP..TEMP.LIST')
 END
 SET &CNT = &CNT + 1
 SLEEP 15
END
SET &PREFIX = &SYSPREF
OPENFILE LIST OUTPUT
SET LIST =  STATUS FROM SYS1.UADS POINT OF VIEW
PUTFILE LIST
SET &RP = &STR())
PROFILE NOPREFIX
SET &USER = USER
SET &P = Ø
SET &RET = Ø
SET &SYSOUTTRAP = 999999
LISTDS 'SYS1.UADS' MEMBERS
SET &P = Ø
SET &MAXLINE = &SYSOUTLINE
DO WHILE &P < &MAXLINE
 SET &P = &P + 1
 SET &SYSDVAL = &STR(&&SYSOUTLINE&P)
 SET &SYSDVAL = &STR(&SYSDVAL)
 READDVAL &A1 &A2 &A3 &A4 &A5 &A6 &A7 &A8
 IF &MEMBON = YES THEN DO
  SET &&USER&P = &STR(&A1)
 END
```

```
 IF &STR(&A1) = &STR(—MEMBERS—) THEN DO
  SET &MEMBON = YES
 END
END
SET &P = Ø
SET &RET = Ø
DO WHILE &P < &MAXLINE
 SET &P = &P + 1
 SET &A1 = &STR(&&USER&P)
 SET &A1 = &SUBSTR(1:&LENGTH(&A1)-1,&STR(&A1))
 IF &STR(&A1) NE &STR() THEN DO
  WRITE
  WRITE &STR(========>) INFORMATION FOR USER &STR("&A1")
  SET &HLQ = NO
  SET &DS = NO
  SET &RACFU = NO
  SET &RACFT = NO
  SET &GROUP = NO
  SET &DSP = NO
  SET &UADS = NO
  SET &SYSOUTTRAP = 99999
  SET &RET = Ø
  LISTC ALL ENT('&A1'&RP
  SET &SYSOUTTRAP = Ø
  IF &RET = Ø THEN DO
   SET &HLQ = YES
   SET &SYSOUTTRAP = 99999
   SET &RET = Ø
   LISTC     LEVEL(&A1&RP
   SET &SYSOUTTRAP = Ø
   IF &RET = Ø THEN DO
    SET &DS = YES
    WRITE SOME DATASETS/ENTRIES DO EXIST UNDER ALIAS &STR("&A1")
   END
   ELSE DO
    WRITE NO DATASETS/ENTRIES EXIST UNDER ALIAS &STR("&A1")
   END
  END
  ELSE DO
   WRITE NO ALIAS EXISTS FOR &STR("&A1") ANY LONGER
  END
  SET &RET = Ø
  LG &A1
  IF &RET = Ø THEN DO
   SET &GROUP = YES
  END
  ELSE DO
   WRITE NO RACF GROUP EXISTS FOR &STR("&A1")
   CONTROL NOMSG
   SET &SYSOUTTRAP = 999999
```

```
 SET &RET = Ø
 LU &A1
 SET &SYSOUTTRAP = Ø
 SET &LURET = &RET
 CONTROL MSG
 IF &LURET < 4 THEN DO
  SET &RACFU = YES
  WRITE "&A1" USER EXISTS IN RACF
  SET &SYSOUTTRAP = 999999
  LU &A1 TSO NORACF
  SET &SYSOUTTRAP = Ø
  SET &MAXTSO = &SYSOUTLINE
  SET &RACFTSO = &STR()
  SET &N = Ø
  SET &RET = Ø
  DO WHILE &N < &MAXTSO
   SET &N = &N + 1
   SET &SYSDVAL = &STR(&&SYSOUTLINE&N)
   SET &SYSDVAL = &STR(&SYSDVAL)
   READDVAL &B1 &B2 &B3 &B4 &B5 &B6 &B7 &B8
   IF &STR(&B1) = NO AND &STR(&B2) = TSO AND &STR(&B3) = INFORMATION +
   THEN DO
    WRITE "&A1" IS NOT DEFINED AS TSO USER IN RACF
    SET &RACFTSO = NO
   END
  END
  IF &RACFTSO NE NO THEN DO
   SET &RACFT = YES
   WRITE "&A1" IS DEFINED AS TSO USER IN RACF
  END
 END
 ELSE DO
  WRITE "&A1" DOES NOT EXIST AS USER IN RACF
 END
END
SET &RET = Ø
SET &MEMBER = &A1
SET &MEMBER = &MEMBER.&STR(.)
SEARCH ALL CLASS(DATASET) MASK(&MEMBER) LIST
IF &RET > Ø THEN DO
 WRITE NO RACF DATASET PROFILES EXIST FOR &STR("&A1")
END
ELSE DO
 SET &DSP = YES
END
SET &MEMBER = &A1
SET &MEMBER = &MEMBER.Ø
IF &LENGTH(&STR(&A1)) < 8 THEN DO
 SET &UADS = &STR(&SYSDSN('SYS1.UADS(&MEMBER)'))
 WRITE TSO DEFINITION IN SYS1.UADS = &UADS FOR &STR("&A1")
```

```
 IF &STR(&UADS) = OK THEN DO
  SET &UADS = YES
 END
END
ELSE DO
 WRITE TSO DEFINITION IN SYS1.UADS = MEMBER NOT FOUND FOR &STR("&A1")
END
IF &UADS NE YES THEN DO
 WRITE ===> LOGICAL ERROR: USER &STR("&A1") MISSING
 SET &OUT = &STR(LOGICAL ERROR: USER &STR("&A1") MISSING)
 PUTFILE OUT
 SET &LIST = &STR(&OUT)
 PUTFILE LIST
 WTP &STR(&OUT)
 SET &ERROR = YES
END
IF &UADS = YES THEN DO
 IF &HLQ NE YES THEN DO
  WRITE ===> ERROR: ALIAS MISSING FOR USER &STR("&A1")
  SET &OUT = &STR(ERROR: ALIAS MISSING FOR USER &STR("&A1"))
  PUTFILE OUT
  SET &LIST = &STR(&OUT)
  PUTFILE LIST
  WTP &STR(&OUT)
  SET &ERROR = YES
 END
 IF &DS NE YES THEN DO
  WRITE ===> POSSIBLE ERROR: NO DATASET FOR USER &STR("&A1")
  SET &OUT = &STR(POSSIBLE ERROR: NO DATASET FOR USER &STR("&A1"))
  PUTFILE OUT
 END
 IF &DSP NE YES THEN DO
  WRITE ===> ERROR: NO DATASET RACF PROFILE FOR USER &STR("&A1")
  SET &OUT = &STR(ERROR: NO DATASET RACF PROFILE FOR USER &STR("&A1"))
  PUTFILE OUT
  SET &LIST = &STR(&OUT)
  PUTFILE LIST
  WTP &STR(&OUT)
  SET &ERROR = YES
  IF &DS = YES THEN DO
   WRITE ===>        BUT DATASETS EXIST.
   SET &OUT = &STR(        BUT DATASETS EXIST.)
   PUTFILE OUT
   SET &LIST = &STR(&OUT)
   PUTFILE LIST
   WTP &STR(&OUT)
  END
 END
 /* The following users are allowed in UADs for emergency purposes. */
 /* Installation standard requires TSO users to be exclusively      */
```

```
/* defined using RACF TSO segment otherwise.                        */
IF &STR(&A1) NE USERØ1 AND +
&STR(&A1) NE USERØ2  AND +
&STR(&A1) NE USERØ3 AND +
&STR(&A1) NE USERØ4 THEN DO
 IF &RACFT = YES AND &UADS = YES THEN DO
  WRITE ===> ERROR: &STR("&A1") TSO USER BOTH IN RACF AND UADS
  SET &OUT = &STR(ERROR: &STR("&A1") TSO USER BOTH IN RACF AND UADS)
  PUTFILE OUT
  SET &LIST = &STR(&OUT)
  PUTFILE LIST
  WTP &STR(&OUT)
  SET &ERROR = YES
  ALLOC FI(UADS) DA('SYS1.UADS') SHR REUSE
  DEL 'SYS1.UADS(&STR(&A1.Ø))' FILE(UADS)
  DEL 'SYS1.UADS(&STR(&A1.1))' FILE(UADS)
  DEL 'SYS1.UADS(&STR(&A1.2))' FILE(UADS)
  /* delete evt up to suffix 9 */
  FREE FI(UADS)
  WRITE &STR(===> USER &A1 DELETED FROM UADS)
  SET &OUT = &STR(===> USER &A1 DELETED FROM UADS)
  PUTFILE OUT
  SET &LIST = &STR(&OUT)
  PUTFILE LIST
  WTP &STR(&OUT)
 END
END
/* Installation standard requires TSO users to be exclusively      */
/* defined using RACF TSO segment.                                  */
IF &RACFT = NO AND &UADS = YES THEN DO
 WRITE ===> ERROR: &STR("&A1") TSO USER IS DEFINED IN UADS; SHOULD +
 BE IN RACF ONLY
 SET &OUT = &STR(ERROR: &STR("&A1") TSO USER IS DEFINED IN UADS; +
 SHOULD BE IN RACF ONLY)
 PUTFILE OUT
 SET &LIST = &STR(&OUT)
 PUTFILE LIST
 WTP &STR(&OUT)
 SET &ERROR = YES
 ALLOC FI(UADS) DA('SYS1.UADS') SHR REUSE
 DEL 'SYS1.UADS(&STR(&A1.Ø))' FILE(UADS)
 DEL 'SYS1.UADS(&STR(&A1.1))' FILE(UADS)
 DEL 'SYS1.UADS(&STR(&A1.2))' FILE(UADS)
  /* delete evt up to suffix 9 */
 FREE FI(UADS)
 WRITE &STR(===> USER &A1 DELETED FROM UADS)
 SET &OUT = &STR(===> USER &A1 DELETED FROM UADS)
 PUTFILE OUT
 SET &LIST = &STR(&OUT)
 PUTFILE LIST
```

```
    WTP &STR(&OUT)
   END
   IF &RACFU NE YES AND &GROUP NE YES THEN DO
    WRITE ===> ERROR: &STR("&A1") NEITHER RACF USER NOR GROUP
    SET &OUT = &STR(ERROR: &STR("&A1") NEITHER RACF USER NOR GROUP)
    PUTFILE OUT
    SET &LIST = &STR(&OUT)
    PUTFILE LIST
    WTP &STR(&OUT)
    SET &ERROR = YES
   END
   IF &UADS = YES AND &GROUP = YES THEN DO
    WRITE ===> ERROR: &STR("&A1") BOTH IN UADS AND RACF GROUP
    SET &OUT = &STR(ERROR: &STR("&A1") BOTH IN UADS AND RACF GROUP)
    PUTFILE OUT
    SET &LIST = &STR(&OUT)
    PUTFILE LIST
    WTP &STR(&OUT)
    SET &ERROR = YES
   END
   IF &UADS = YES AND &RACFU NE YES THEN DO
    WRITE ===> ERROR: &STR("&A1") IN UADS BUT NOT A RACF NONTSO USER
    SET &OUT = &STR(ERROR: &STR("&A1") IN UADS BUT NOT A RACF +
    NONTSO USER.)
    PUTFILE OUT
    SET &LIST = &STR(&OUT)
    PUTFILE LIST
    WTP &STR(&OUT)
    SET &ERROR = YES
   END
  END
 END
END
CLOSFILE OUT
CLOSFILE LIST
IF &ERROR = YES THEN DO
 SET &RC = RC
 SET &N = 1
 SET &RC = &&RC&N
 DO WHILE &RC NE &STR() AND &N < 9
  %MAILSENS ID('&ID') RC('&RC') DS(&PREFIX..&TSTAMP..TEMP.LIST) +
  DEBUG(&DEBUG) RCDOMAIN(YES) /* INFORM ERROR STATUS VIA MAIL */
  SET &N = &N + 1
  SET &RC = &&RC&N
 END
END
FREE DA('&PREFIX..&TSTAMP..TEMP.LIST')
DEL '&PREFIX..&TSTAMP..TEMP.LIST'
  /*
//STEP2    EXEC PGM=IKJEFTØ1,DYNAMNBR=128,  EXECUTE CLIST
```

```
// PARM='%CLIST DEBUG(NEBUG)'
//SYSTSPRT DD   SYSOUT=*
//SYSPRINT DD   SYSOUT=*
//OUT      DD   SYSOUT=*
//SYSPROC  DD   DSN=&TEMP,DISP=(OLD,PASS)
//         DD   DSN=SYSTEM.CLIST,DISP=SHR
//SYSTSIN  DD   DUMMY
//SYSIN    DD   DUMMY
//*
```


The utility program SLEEP, providing the wait function:

```
//*
//*    TSO COMMAND, EXEC PGM, CALL OR SUBROUTINE
//*      SET THE ADDRESS SPACE IN WAIT THE NUMBER OF SECS INDICATED
//*      Examples of how to call for waiting 1Ø seconds:
//*       SLEEP 1Ø
//*       // EXEC PGM=SLEEP,PARM='1Ø'
//*       CALL 'SYSn.LINKLIB(SLEEP) '1Ø'
//*
// EXEC ASMCL,MEMBER=SLEEP,
// PARM.ASM='RENT',
// PARM.LKED='XREF,LET,LIST,RENT,REFR,REUS,AC=1'
         GBLC  &ID
         GBLA  &IDLEN
SLEEP    INITR AMODE=31,RMODE=ANY,GENCODE=YES,SIZE=GETSIZE
WORKAREA DSECT
         ORG   USERWORK
DW       DS    D                        WORK FOR CONVERT
INTVL    DS    F                        WAIT INTERVAL
GETSIZE  EQU   *-WORKAREA
&ID      CSECT
         LTR   R15,R15                  TEST FOR ZERO DATA LENGTH
         BNP   EXITRC16                 IF NO DATA
         LR    R8,R14                   GET ADDR OF 1ST DATAADDR
SCAN     EQU   *
         TM    OPTIONS,ATTN             IS ATTN FLAG SET
         BO    EXIT                     RETURN IF ATTN
         CLI   Ø(R8),C'Ø'               TAKE AWAY LEADING NONDIGITS/Ø
         BH    FIRSTDIG                 GOT A DIGIT
         LA    R8,1(R8)                 POINT TO NEXT
         BCT   R15,SCAN                 RECYCLE
         B     EXITRC8                  NO VALUE, THEN EXIT WITH ERROR
FIRSTDIG EQU   *
         LR    R9,R8                    SAVE ADDR OF FIRST DIGIT
SCAN2    EQU   *
         LA    R9,1(R9)                 GET NEXT BYTE
```



```
         BCTR  R15,Ø                    COUNT DOWN RESIDUAL COUNT
         LTR   R15,R15                  SOMETHING LEFT
         BZ    ENDSCAN                  NO MORE INPUT
         CLI   Ø(R9),C'Ø'               LOOK FOR NONDIGITS
         BNL   SCAN2                    IF DIGIT RESCAN
ENDSCAN  EQU   *
         SR    R9,R8                    COMPUTE LENGTH
         CH    R9,=H'9'                 TEST FOR TOO LONG
         BNH   LENOK                    LENGTH OK
         LA    R9,9                     ASSUME LENGTH OF 9
LENOK    EQU   *
         BCTR  R9,RØ                    REDUCE FOR EXECUTE
         LA    R1Ø,7                    GET LENGTH OF DOUBLE WORD
         SLL   R1Ø,R4                   SHIFT TO HIGH ORDER
         OR    R1Ø,R9                   SET UP FOR EXECUTE
         EX    R1Ø,PACK                 PACK THE NUMBER
         CVB   R11,DW                   CONVERT TO BINARY
         LA    R6,1ØØ                   GET IN HUNDREDS
         MR    R1Ø,R6                   GET IN HUNDREDS
         ST    R11,INTVL                SAVE WAIT TIME
         STIMER WAIT,BINTVL=INTVL       WAIT
         EXITR                          RETURN
PACK     PACK  DW(Ø),Ø(Ø,R8)            EXECUTED PACK
         LTORG
         END
```

The utility program WTP, providing WTP-function:

```
// EXEC ASMCL,MEMBER=WTP
*
*  ISSUES WTO WITH ROUTECODE 11 I.E A WTP
*  PROGRAM CAN BE EXECUTED VIA JCL, TSO CALL OR AS TSO COMMAND.
*
*  // EXEC PGM=WTP WITH INPUT FROM SYSIN              OR
*  // EXEC PGM=WTP,PARM='XXX XX'                      OR
*  WTP XXX XX                                         TSO COMMAND
*
         PRINT NOGEN
         CVT   DSECT=YES,PREFIX=YES,LIST=NO
         PRINT NOGEN
         DCBD  DSORG=PS
         IHAASCB
         IHAASXB
         IHAPSA
         USING PSA,RØ
         IKJTCB
         IHAACEE
         IEZJSCB
         IKJPSCB
         IEFAJCTB
```

```
         IEFTCT
         IKJTSB
         IEESMCA
         IEFUCBOB PREFIX=YES
UCBPFLEN EQU   UCBCMSEG-UCB
         IEFJESCT .                      JESCT
         IEFJSCVT .                      JSCVT (SSCT)
         IHASDWA DSECT=YES               SDWA FOR ESTAE/SETRP MACRO
         PRINT GEN
WORKAREA DSECT                           GETMAINED WORKARE
SAVEAREA DS    CL72                      SAVE AREA
STAXD    STAX  STAXEXIT,MF=L             STAX LIST FORM
ESTAEW   DS    XL(LESTAEL)               ESTAE PARM LIST AREA
ESTAPARM DS    4F                        PARM LIST TO RETRY ROUTINE:
RETCODE  DS    A                         RETURN CODE
PARMADDR DS    A                         ADDR OF PARMLIST
OPTIONS  DS    C                         EXECUTION OPTIONS
ATTN     EQU   X'8Ø'                     ATTN FLAG SET
WORKLEN  EQU   *-WORKAREA                LENGTH TO GETMAIN
ML       EQU   12Ø-1Ø                    MSG LENGTH - JOBID
&ID      SETC  'WTP'
&IDLEN   SETA  K'&ID
&ID      INITR SIZE=WORKLEN,AMODE=24,RMODE=24
         USING WORKAREA,R13              ADDRESS WORKAREA
         LA    R15,WORKLEN-L'SAVEAREA    SET UP FOR CLEAR OF GETMAINED
         LA    R14,WORKAREA+L'SAVEAREA   POINT AFTER SAVE AREA
         XR    R2,R2                     SET UP FOR CLEAR OF GETMAINED
         XR    R3,R3                     SET UP FOR CLEAR OF GETMAINED
         MVCL  R14,R2                    CLEAR GETMAINED
         XC    RETCODE,RETCODE           CLEAR RETURN CODE
         ST    R1,PARMADDR               SAVE ADDR OF PARMLIST
*        SET UP RECOVERY ENVIRONMENT
         LA    RØ,RTRYRTN1               RETRY ROUTINE - NO SDWA
         ST    RØ,ESTAPARM               STORE IN PARAMETER LIST
         LA    RØ,RTRYRTN2               RETRY ROUTINE WITH SDWA
         ST    RØ,ESTAPARM+4             STORE IN PARAMETER LIST
         STM   R12,R13,ESTAPARM+8        STORE BASE & DATA REG IN PARM
         MVC   ESTAEW(LESTAEL),ESTAEL    MOVE IN ESTAE PARAMETER LIST
         ESTAE RECOVERY,CT,PARAM=ESTAPARM,MF=(E,ESTAEW) SETUP RCVRY
*        SET UP ATTENTION ENVIRONMENT
         MVC   STAXD(STAXLEN),STAXL      MOVE IN STAX LIST TO GETMAINED
         STAX  STAXEXIT,USADDR=OPTIONS,MF=(E,STAXD) SET ATTN EXIT
         L     R11,PARMADDR              PTR PARM ADDR - FIRST COMMAND
         L     R11,Ø(R11)                SAVE ADDR OF PARM IN R11
         LA    R1Ø,INPUT                 ADDRESS DCB
         USING IHADCB,R1Ø                ADDRESS DCB
         ICM   R2,3,Ø(R11)               LENGTH TO R2
         BZ    WTPGETIN                  YES - GET RECORD FROM SYSIN
         CH    R2,=AL2(4+&IDLEN)         TEST FOR ZERO DATA
         BL    SUBROUT                   TRY EXEC/CALL
```

```
         CLC   4(&IDLEN,R11),=C'&ID'    TSO COMMAND
         BNE   SUBROUT                  TRY EXEC/CALL
         CH    R2,=AL2(4+&IDLEN)        TEST FOR ZERO DATA
         BE    EXITRC8                  IF ZERO, IGNORE
         SH    R2,=AL2(5+&IDLEN)        SUBTRACT LENGTH OF HEADER
         LA    R11,&IDLEN+5-2(R11)      SIMULATE EXEC PARM
SUBROUT  EQU   *
         C     R2,=A(ML)                COMPARE LENGTH WITH MAX LENGTH
         BNH   WTPMOVE                  LENGTH OF PARM OKAY
         LA    R2,ML                    RESET PARM LENGTH TO MAX
WTPMOVE  EQU   *
         MVC   WTPMSG,=CL(ML)' '        CLEAR OUT MSG AREA
         BCTR  R2,Ø                     LENGTH-1 FOR MVC
         EX    R2,*+4                   MOVE PARM INTO MSGAREA
         MVC   WTPMSG(Ø),2(R11)         EXEC INSTRUCTION TO MOVE PARM
         B     WTPGOTIN                 PROCESS FIRST MSG
WTPGETIN EQU   *
         TM    DCBOFLGS,DCBOFOPN        ANY INPUT FILE
         BO    WTPGET                   BR IF SO
         OPEN  (INPUT)                  OPEN IT
         TM    DCBOFLGS,DCBOFOPN        DID IT OPEN
         BZ    EXITRC8                  NO - NO VERIFICATION
WTPGET   EQU   *
         GET   INPUT                    READ RECORD FROM SYSIN
         MVC   WTPMSG(73),Ø(R1)         MOVE INPUT TO MSG AREA
WTPGOTIN EQU   *
         CLC   WTPMSG,=CL(ML)' '        TEST FOR BLANK MESSAGE
         BE    WTPGETIN                 SKIP BLANK RECORD
         JOBNAME (R7)                   GET JOBNAME ADDR (MACRO)
JOBFND   EQU   *
         MVC   WTPWTOL+4(8),Ø(R7)       JOB/STC/TSU ID
         MVC   WTPWTOL+12(2),=C': '     DELIMITER
         MVC   WTPWTOL+14(ML),WTPMSG    MOVE WTO MSG
         TM    OPTIONS,ATTN             IS ATTN FLAG SET
         BO    EXIT                     RETURN IF ATTN
         XR    RØ,RØ                    CLEAR REGISTER Ø BEFORE WTO
         WTO   MF=(E,WTPWTOL)           DISPLAY MSG
         ST    R15,RETCODE              SAVE RETURN CODE
         TM    DCBOFLGS,DCBOFOPN        ANY INPUT FILE
         BO    WTPGETIN                 READ NEXT RECORD
WTPEOD   EQU   *
         B     EXIT                     RETURN TO CALLER
EXITRC4  EQU   *
         MVC   RETCODE,=F'4'            SET RETURNCODE 4
         B     EXIT                     GO EXIT
EXITRC8  EQU   *
         MVC   RETCODE,=F'8'            SET RETURNCODE 8
         B     EXIT                     GO EXIT
EXITRC12 EQU   *
         MVC   RETCODE,=F'12'           SET RETURNCODE 12
```

```
         B     EXIT                      GO EXIT
EXIT     EQU   *
         ESTAE Ø                         CANCEL ESTAE EXIT
QUICKOUT EQU   *
         L     R15,RETCODE               GET RETURN CODE
         EXITR RC=(R15)                  RETURN TO CALLER
*  ESTAE EXIT ROUTINE
RECOVERY EQU   *
         PUSH  USING                     SAVE PREVIOUS BASE REGS
         USING *,R15                     SET UP BASE REGISTER
         USING SDWA,R1                   SET UP ADDRESSABILITY TO SDWA
         LA    R4,12                     PUT 12 IN REGISTER FOR COMPARE
         CR    RØ,R4                     IS SDWA PRESENT?
         BNE   HAVESDWA                  YES, BR TO PROCESS WITH SDWA
         L     RØ,Ø(R2)                  LOAD RETRY ADDR FROM PARM LIST
         LA    R15,4                     SET RC TO RETRY ADDR IN RØ
         BR    R14                       RETURN WITH RETRY ADDR
HAVESDWA EQU   *                         ENTER HERE IF SDWA PRESENT
         ST    R14,12(R13)               SAVE RETURN ADDRESS
         L     R2,SDWAPARM               LOAD PARAM LIST ADDR FROM SDWA
         ST    R2,SDWASRØ1               SAVE POINTER TO ESTAE PARM LIST
         L     R2,4(R2)                  LOAD RETRY ADDRESS
         SETRP RC=4,,RETADDR=(2),RETREGS=YES,FRESDWA=YES,REGS=(14)
         DROP  R15,R1                    DROP LOCAL ADRESSABILITY
         POP   USING                     RESTORE PREVIOUS BASE REGS
RTRYRTN1 EQU   *                         RETRY ROUTINE WITH NO SDWA
RTRYRTN2 EQU   *                         ESTAE RETRY ROUTINE WITH SDWA
         LM    R12,R13,8(R1)             LOAD REGS FOR ESTAE PARM LIST
         LA    R15,16                    SET SEVERE ERROR
         ST    R15,RETCODE               INDICATE SEVERE ERROR
         B     QUICKOUT                  AND EXIT
STAXEXIT EQU   *
         USING *,R15                     ADDRESS TEMPORARILY
         SAVE  (14,12)                   SAVE REGS
         BALR  R12,Ø                     SET UP BASE
STAXBASE EQU   *
         L     R15,STAXOFFS              SET UP BASE OFFSET
         SR    R12,R15                   SET UP REAL BASE
         DROP  R15                       LEAVE TEMPORARY ADDRESSING
* CLEAN UP WHAT NEED TO
         L     R9,8(R1)                  GET USER DATA
         OI    Ø(R9),ATTN                SET ATTN FLAG
         RETURN (14,12),RC=8             RETURN
STAXOFFS DC    A(STAXBASE-&ID)           STAX BASE OFFSET
* DEFINE ESTAE AND STAX LIST FORM
ESTAEL   ESTAE MF=L                      CREATE MODEL ESTAE PARM LIST
LESTAEL  EQU   *-ESTAEL                  NAME ITS LENGTH
STAXL    STAX  STAXEXIT,MF=L             STAX LIST FORM
STAXLEN  EQU   *-STAXL                   LENGTH OF STAX
*
```

```
INPUT    DCB   DDNAME=SYSIN,DSORG=PS,MACRF=(GL),EODAD=WTPEOD
WTPMSG   DC    CL(ML)' '               STORAGE FOR ONE MSG
         CNOP  Ø,4                     ALIGN ON WORD
WTPWTOL  WTO   '12345678911234567892123456789312345678941234567895123 45*
               67896123456789712345678981234567899123456789A123456789B1*
               23456789C',                                              *
               ROUTCDE=(11),DESC=(7),MF=L
         LTORG
         END
```

The macro JOBNAME, used in the WTP program:

```
*
*   RETURNS JOBNAME POINTER IN REGISTER, DEFAULT TO R15
*    CAN BE OVERWRITTEN BY JOBNAME (RX)
*
         MACRO
&NAME    JOBNAME &REG
         LCLC  &JOBFND,&JOBNFND,&RNULL
&JOBFND  SETC  'JN1'.'&SYSNDX'
&JOBNFND SETC  'JN2'.'&SYSNDX'
&RNULL   SETC  'JN3'.'&SYSNDX'
         AIF   ('&REG' EQ '').RNULL
         AIF   ('&REG'(1,1) EQ '(').AREG
         AGO   .RNULL
AREG    ANOP
&REGR    SETC  '&REG(1)'
         AGO   .REG
RNULL   ANOP
&REGR    SETC  '15'
REG     ANOP
&NAME    DS    ØH .
         L     &REGR,X'224'            GET ASCB ADDR
         CLC   &RNULL,X'AC'(&REGR)     ANY JOBNAME ADDR
         BE    &JOBNFND                NO
         L     &REGR,X'AC'(&REGR)      GET JOBNAME IF JOB
         B     &JOBFND                 YES
&JOBNFND DS    ØH .
         L     &REGR,X'BØ'(&REGR)      ELSE STARTED TASK OR TSO
         B     &JOBFND                 YES
&RNULL   DC    AL4(Ø)                  NULL ADDRESS
&JOBFND  DS    ØH .
         MEXIT
         MEND
```

Nils Plum (Denmark)

Monitoring RRSF connectivity

There are a number of external influences that can affect RRSF processing. Typically, the end user may be aware of password synchronization problems before the RACF administrator.

To reduce the number of calls to our Support Desk relating to connectivity failures, we have introduced a schedulable audit of RRSF connectivity based on the results of TARGET.

LIST COMMANDS

The following procedure will capture the results of a TARGET LIST command issued via batch SDSF. Use edit macros to reduce the TARGET LIST output to identify disconnected states, and wrap SEND command(s) around the remaining output. Finally, the SEND command(s) are executed for selected users (RACF administrators).

For the best results, we schedule this procedure to run on each of our RRSF MAIN systems at 08:00. Then, when the SEND message recipients first log-on for the day, they are in a position to take corrective action if necessary.

The edit macro code is flexible enough to handle additions to the size of our RRSF environment.

```
//RALRRSF PROC
//*
//*  THIS PROCEDURE WILL REPORT ON RRSF CONNECTIVITY FAILURES.
//*  TO CHANGE NOTIFIABLE USERS UPDATE HLQ.RACF.CLIST(RRSF3) .
//*
//SØ1PROC     EXEC PGM=IEFBR14
//SYSPRINT DD  SYSOUT=*
//SDA  DD  DSN=HLQ.SDSF.RRSF,DISP=(OLD,DELETE)
//SDSFØØ2      EXEC PGM=ISFAFD,PARM='++8Ø,149'
//ISFOUT DD  DISP=(,CATLG,DELETE),DSN=HLQ.SDSF.RRSF,
//  DCB=(RECFM=FBA,BLKSIZE=279ØØ,LRECL=15Ø),
//  SPACE=(TRK,(1,1))
//ISFIN  DD  DISP=SHR,DSN=HLQ.CSLIB(SDSF1)
//SØ3JEFT      EXEC PGM=IKJEFTØ1,DYNAMNBR=3Ø
//SORTLIB DD  DUMMY
//ISPPROF DD  DSN=&&PROFOLE,DISP=(,PASS),SPACE=(TRK,(1,1,1)),
//  DCB=(LRECL=8Ø,BLKSIZE=616Ø,RECFM=FB),UNIT=SYSDA
```

```
//SYSLBC DD  DSN=SYS1.BRODCAST,DISP=SHR
//SYSPROC DD  DSN=HLQ.RACF.CLIST,DISP=SHR
//ISPPLIB DD  DSN=HLQ.ISPPLIB,DISP=SHR
//ISPMLIB DD  DSN=HLQ.ISPMLIB,DISP=SHR
//ISPSLIB DD  DSN=HLQ.ISPSLIB,DISP=SHR
//ISPTLIB DD  DSN=HLQ.ISPTLIB,DISP=SHR
//ISPCTL1 DD  DSN=&&CTLA,DISP=(,PASS),UNIT=SYSDA,SPACE=(TRK,(5,5)),
//  DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//ISPCTL2 DD  DSN=&&CTLB,DISP=(,PASS),UNIT=SYSDA,SPACE=(TRK,(5,5)),
//  DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN DD  DISP=SHR,DSN=HLQ.CSLIB(RRSF1)
//S04JEFT      EXEC PGM=IKJEFT01,DYNAMNBR=30,COND=(0,NE,S03JEFT)
//SORTLIB DD  DUMMY
//ISPPROF DD  DSN=&&PROFOLE,DISP=(,PASS),SPACE=(TRK,(1,1,1)),
//  DCB=(LRECL=80,BLKSIZE=6160,RECFM=FB),UNIT=SYSDA
//SYSLBC DD  DSN=SYS1.BRODCAST,DISP=SHR
//SYSPROC DD  DSN=HLQ.RACF.CLIST,DISP=SHR
//ISPPLIB DD  DSN=HLQ.ISPPLIB,DISP=SHR
//ISPMLIB DD  DSN=HLQ.ISPMLIB,DISP=SHR
//ISPSLIB DD  DSN=HLQ.ISPSLIB,DISP=SHR
//ISPTLIB DD  DSN=HLQ.ISPTLIB,DISP=SHR
//ISPCTL1 DD  DSN=&&CTLC,DISP=(,PASS),UNIT=SYSDA,SPACE=(TRK,(5,5)),
//  DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//ISPCTL2 DD  DSN=&&CTLD,DISP=(,PASS),UNIT=SYSDA,SPACE=(TRK,(5,5)),
//  DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN DD  DISP=SHR,DSN=HLQ.CSLIB(RRSF4)

******** HLQ.CSLIB(SDSF1)

 SET DELAY 99
 /RACF TARGET LIST
 ULOG
 PRINT ULOG
 AFD REFRESH
 PRINT CLOSE

******** HLQ.CSLIB(RRSF1)

 ISPSTART CMD(RRSF2) BDISPMAX(500)

******** HLQ.RACF.CLIST(RRSF2)

 CONTROL MSG LIST
 ISREDIT MACRO
 ISPEXEC EDIT DATASET ('HLQ.SDSF.RRSF') MACRO (RRSF3)
 ISREDIT END
 EXIT
******** HLQ.RACF.CLIST(RRSF3)
```

```
CONTROL MSG LIST
ISREDIT MACRO
/* DELETES ALL BUT 'TARGET LIST' OUTPUT AND ADJUSTS CONTINUED LINE.   */
ISREDIT BOUNDS = 1 15Ø
ISREDIT EX  ALL (RACF)
ISREDIT EX  ALL ((RACF))
ISREDIT EX  ALL 'STATE. '
ISREDIT EX  ALL RESPONSE=
ISREDIT DEL ALL NX
ISREDIT RESET
ISREDIT CHANGE 'RESPONSE' '     RESPONSE' ALL
ISREDIT CHANGE '((RACF))' '' ALL
ISREDIT EX  ALL '           (RACF)'
ISREDIT DEL ALL X
ISREDIT SAVE
ISREDIT CHANGE '(RACF)' '' ALL
ISREDIT SAVE
/* SETS COLUMN BOUNDARIES AND IDENTIFIES NUMBER OF LINES REMAINING.   */
ISREDIT (SCOL,ECOL) = BOUNDS
ISREDIT BOUNDS = 4 12Ø
ISREDIT (TOTAL) = LINENUM .ZLAST
WRITE &TOTAL
/* PUTS INDIVIDUAL SYSNAME INFORMATION INTO A SINGLE LINE.            */
START: ISREDIT TFLOW &EVAL(&TOTAL-1) &ECOL
    SET &TOTAL = &TOTAL - 2
    WRITE &TOTAL
    IF &TOTAL ¬= 1 THEN GOTO START
/* DELETES ALL BUT ERRONEOUS STATES.                                  */
    ELSE ISREDIT EX  ALL '               ' 4
         ISREDIT EX  ALL 'THE DEFINED STATE.'
         ISREDIT EX  ALL 'THE OPERATIVE ACTIVE STATE.'
         ISREDIT DEL ALL EX
         ISREDIT SAVE
/* IF NO ERRONEOUS STATES CONSTRUCT MESSAGE TO THAT EFFECT.           */
         ISREDIT (TOTAL2) = LINENUM .ZLAST
         IF &TOTAL2 = 1 THEN +
         ISREDIT LINE_AFTER 1 = "IRRXXXXI THERE ARE NO ERRON-
           EOUS STATES WITH REMOTE RRSF FROM THIS SYSTEM."
    ELSE ISREDIT SAVE
         ISREDIT SAVE
/* RESETS COLUMN BOUNDARIES.                                          */
         ISREDIT BOUNDS = 1 15Ø
/* WRAPS 'TSO SEND' COMMAND INFORMATION AROUND REMAINING LINES.       */
         ISREDIT CHANGE " " " TSOEXEC SE '" 1 ALL
         ISREDIT CHANGE ". " ".',USER(userid1 userid2 userid3-
           ),LOGON" ALL
         ISREDIT CHANGE "  " " ',USER(userid1 userid2 userid3-
           ),LOGON" FIRST
         ISREDIT SAVE
```

```
         ISREDIT END

******** HLQ.CSLIB(RRSF4)

 ISPSTART CMD(RRSF5)

******** HLQ.RACF.CLIST(RRSF5)

 EX  'HLQ.SDSF.RRSF'
```

Bruce Tyler
Barclays Technology Services (UK)

Using RRSF and GTF – a warning

If you have a site that exploits RRSF, you should be aware of the following situation that can occur, allowing RACF passwords to be found in clear text.

We were in discussion with IBM concerning some problems we had encountered using RRSF, and a GTF trace was requested to enable them to diagnose the problem. Once we completed the trace, and before sending this to IBM, we formatted the records to make sure the data was OK. In doing this, we found that anyone who had changed their password during the trace period had their details trapped such that it was easy to see their new password.

Following some research, it was found that the system programmers guide actually states that "*trace records can contain passwords, so be sure that trace output datasets are appropriately protected*".

Being aware of the problem, we edited the data before shipping, and ensured that all the users were notified to change their password again.

If you are concerned that this could affect your site, the trace to be wary of is for APPC events with a user record type of F44.

Vanguard Integrity Professionals has announced Version 2.1 of its Vanguard RioVision Windows interface for OS/390 Security Server administration, providing mainframe security administrators with a view of RACF and DB2 security data.

The Security Server group tree is represented in a Windows Explorer-type graphical format and the administrator can add or remove permissions, modify group or user profiles, add new users, and perform other security tasks.

New features include context-sensitive help down to the field level, and balloon help for all tool bar buttons. RioVision now shows existing 'managed' and 'peer' relationships, letting administrators define, approve, and un-define these relationships. It automatically generates the necessary RACLINK commands.

The new QUERY operator command displays the current RioVision/Security Server settings, the number and names of users currently signed onto RioVision, and the status of the maintenance that has been applied to the product.

RioVision can now apply revoke/resume changes to all connections of the group or user being administered, instead of one at a time. When a general resource class in Resource Explorer is selected, all its resources will immediately appear.

RioVision now has more diagnostics to pinpoint causes of problems. Trace capabilities have been added to components to track several levels of detail.
The Security Assurance Framework integrates the previously separate functions of security administration, reporting, and assessment. The product suite includes Administrator for automated security server administration, Reporter for security reporting, and Analyzer for system integrity analysis.

For further information contact:
Vanguard Integrity Professionals, 180 South Anita Drive, Orange, CA 92668-3306, USA.
Tel: (714) 939 0377.
Sequal Software, Eastlands Court, St Peters Road, Rugby, CV21 3QP, UK.
Tel: (01788) 551101.
URL: http://www.viplink.com.

* * *

Xephon has just launched four weekly news services covering the following subject areas:

- Data Centre
- Distributed Systems
- Networks
- Software

Each week, subscribers receive, by e-mail, a short news bulletin consisting of a list of items; each item has a link to the page on our Web site that contains the corresponding article. Each news bulletin also carries links to the main industry news stories of the week.

To subscribe to one or more of these news services, or review recent articles, point your browser at http://www.xephon.com/newz.html.

* * *