14

RACF

November 1998

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 £185.00 in the UK; $280.00 in the USA and Canada; £191.00 in Europe; £197.00 in Australasia and Japan; and £195.50 elsewhere. In all cases the price includes postage. Individual issues, starting with the August 1995 issue, are available separately to subscribers for £46.50 ($70.00) each including postage.

Printed in England.

Using ICHRCX02 after PROTECT ALL

The PROTECT ALL option is used to protect all resources being accessed in MVS (OS/390) in the active resource classes. Some shops use it because they decided it was the correct thing to do for business reasons, other shops use it because the auditors decided it was the correct thing for them to do.

However well intentioned this option may be, it can play havoc with the systems programmers' productivity. The most familiar scenario concerned SMP/E receive of software products. For a while, IBM packaged SMP/E RELFILEs with the HLQ (High Level Qualifier) being the FMID of the product. For example, if it was a product of five FMIDs for the product base and features, the systems programmer had to have the security administrator build five profile rules before any receives could be done. For OEM software vendors the situation was worse – the author recalls one OEM vendor using the PTF identifier for the RELFILE HLQ.

There had to be a way of using the PROTECT ALL feature and not impact on productivity at the same time. This was accomplished by using the RACF INSTALLATION POSTPROCESSING EXIT ICHRCX02.

The principle behind the set-up of the exit was basic. It was decided that the systems programmers were trustworthy and that they should be allowed MONITORED access to datasets otherwise protected.

For these trusted people this exit provides the following:

- If the resource being accessed was a tape dataset, security checking was bypassed with a WTO to the console log access, specifying what access was allowed and what access was actually granted.
- If the resource being accessed was not protected, access was granted with a WTO to the console log, specifying that unprotected resource access has been granted to the user.

There was also an understanding by the security administrators to

check the MVS system log daily (or weekly) for the messages generated by this exit and log the accesses for the benefit of the auditors, who then deemed that there were enough checks in place to allow this access to exist.

The example ICHRCX02 exit coded here was designed for flexibility. Both types of action (TAPE or PROTECT ALL) are supported, and the access being granted can be changed with just a few well-placed comments. In addition, while the user-ids being tested against can be generic, the user-id being logged in the WTO will be a specific user-id.

There are several important things to remember when implementing this RACF INSTALLATION EXIT:

- RACF INSTALLATION EXIT routines must be loaded into PLPA, FLPA, or MLPA.
- The RACF initialization routine loads the exit routines during MVS IPL and places the exit addresses in the RACF Communications Vector Table (RCVT). If a change needs to be made to this exit, it must occur via an IPL.
- RACF INSTALLATION EXIT routines must be AMODE(31) or AMODE(ANY) and link-edited with AC=1.

Note: from the author's personal experience while installing this exit routine, it is possible to replace this module in the LPA(MLPA) using an OEM MVS-monitoring tool, and the change will be reflected in the RCVT.

The RACF INSTALLATION EXIT routines are provided to systems programmers and administrators to enhance the flexibility of RACF protection installations and increase productivity. However, care must be taken in utilizing these powerful exits to ensure the continued protection of operating system integrity and minimized unauthorized exposure.

ICHRCX02

```
//IMSCONØ$ JOB (IMSØØØØØØØT),'CSH ICHRCXØ2',CLASS=A,MSGCLASS=X,
//    NOTIFY=IMSCONØ,REGION=4Ø96K
```

```
/*JOBPARM ROOM=8Ø14
//*
//ASM    EXEC  PGM=IEV9Ø,PARM='OBJECT,XREF(SHORT),RENT'
//SYSLIB   DD  DISP=SHR,DSN=SYS1.MACLIB
//         DD  DISP=SHR,DSN=SYS1.MODGEN
//SYSUT1   DD  UNIT=SYSALLDA,SPACE=(CYL,(1Ø,5)),DSN=&SYSUT1
//SYSPUNCH DD  DUMMY
//SYSPRINT DD  SYSOUT=*
//SYSLIN   DD  DISP=(,PASS),UNIT=SYSALLDA,SPACE=(CYL,(5,5,Ø)),       *
//         DCB=(BLKSIZE=4ØØ),DSN=&&LOADSET
//SYSIN    DD  *
 TITLE 'ICHRCXØ2 RACHECK POST PROCESSING EXIT       CSH &SYSDATE'
*
* REGISTER USAGE
*
********  CHORNG S. (JACK) HWANG
*         HSA SYSTEMS INC
*         CSHWANG@HOTMAIL.COM
*
*
* R1  - WORK
* R2  - WORK
* R3  - WORK
* R4  - WORK
* R1Ø - BASE FOR RCXPL
* R11 - BASE FOR WTO
* R12 - BASE FOR CODE
*
ICHRCXØ2 CSECT
         STM   14,12,12(13)
         LR    12,15
         USING ICHRCXØ2,12
         LR    1Ø,1              SAVE ADDRESS OF RCXPL
         USING RCXPL,1Ø          ADDRESS RCXPL
*
         L     1,RCXRCODE        GET ADDRESS OF RETURN CODE
         CLC   2(2,1),=H'4'      RESOURCE NOT DEFINED?
         BE    RCOK              YES, CONTINUE PROCESSING
         CLC   2(2,1),=H'8'      ACCESS VIOLATION?
         BE    RCOK              YES, CONTINUE PROCESSING
         B     EXIT              NEITHER, EXIT
*
RCOK     DS    ØH
         USING PSA,Ø             ADDR PSA
         L     1,PSAAOLD         GET ASCB ADDRESS
         USING ASCB,1            ADDR ASCB
         L     1,ASCBASXB        GET ASXB ADDRESS
         USING ASXB,1            ADDR ASXB
         L     1,ASXBSENV        GET ACEE ADDRESS
         LA    2,PELIST          GET STARTING ADDR
```

```
USERIDLP DS    ØH
         CLI   Ø(2),X'Ø7'        TEST LENGTH
         BH    EXIT              END REACHED - EXIT
         XR    3,3               CLEAR R3
         IC    3,Ø(2)            GET LENGTH
         EX    3,CLCUID          COMPARE UID
         USING ACEE,1            ADDR ACEE
*CLCUID  CLC   1(Ø,2),ACEEUSRI   COMPARE UID
         BE    IDOK              ID IS OK, CONTINUE
         LA    2,2(3,2)          GO TO NEXT ENTRY
         B     USERIDLP
CLCUID   CLC   1(Ø,2),ACEEUSRI   COMPARE UID
         DROP  1
*
IDOK     DS    ØH
*
*        TEST FOR RESOURCE NOT DEFINED
*
         L     1,RCXRCODE        GET ADDRESS OF RETURN CODE
         CLC   2(2,1),=H'4'      RESOURCE NOT DEFINED?
         BNE   TESTTAPE          NO, GO CHECK FOR TAPE DATASET
*
         L     1,RCXENORP        GET PROFILE ADDRESS
         CLC   =CL3'SYS',Ø(1)    PROFILE START WITH SYS?
         BE    EXIT              YES, EXIT
         CLC   =CL12'DMS.SYSPARM.',Ø(1) PROFILE START WITH DMS.SYSPARM?
         BE    EXIT              YES, EXIT
         CLC   =CL13'STGADMIN.ADR.',Ø(1) PROFILE STRT WITH DMS.SYSPARM?
         BE    EXIT              YES, EXIT
         B     CONTINUE          CONTINUE WITH PROCESSING
*
*        TEST FOR TAPE DSN
*
TESTTAPE DS    ØH
         L     1,RCXRCODE        GET ADDRESS OF RETURN CODE
         CLC   2(2,1),=H'8'      NOT AUTH?
         BNE   EXIT              NO, EXIT EXIT
         L     1,RCXFLAG3        GET FLAG3 ADDRESS
         TM    Ø(1),RCXDTYPT     DSTYPE=T?
         BNO   EXIT              NO, EXIT
         L     1,RCXFLAG         GET FLAG3 ADDRESS
         TM    Ø(1),RCXLGNOS     LOG=NOFAIL OR NOLOG?
         BZ    CONTINUE          NO, CONTINUE
         L     1,RCXFLAG2        GET FLAG2 ADDRESS
         TM    Ø(1),RCXATTAL     ALTER ATTEMPT?
         BO    CONTINUE          NO, CONTINUE
         B     EXIT
CONTINUE DS    ØH
*
         GETMAIN RU,LV=WTOL      GET WTO WORKAREA
```

```
         LR    11,1              SAVE WTO WORKAREA ADDRESS
         MVC   Ø(WTOL,11),WTO    MOVE WTO MESSAGE
         L     1,PSAAOLD         GET ASCB ADDRESS
         USING ASCB,1            ADDR ASCB
         L     1,ASCBASXB        GET ASXB ADDRESS
         USING ASXB,1            ADDR ASXB
         L     1,ASXBSENV        GET ACEE ADDRESS
         USING ACEE,1            ADDR ACEE
         MVC   13(8,11),ACEEUSRI MOVE USER-ID INTO WTO
         DROP  1                 CLEAR ADDRESSING
         L     1,RCXENORP        GET PROFILE ADDRESS
         MVC   41(44,11),Ø(1)    MOVE PROFILE
         L     1,RCXRCODE        GET ADDRESS OF RETURN CODE
*
         CLC   2(2,1),=H'4'      RESOURCE NOT DEFINED?
         BNE   TAPEDSN           NO, GO MOVE TAPE DSN REQUESTS
         MVC   22(18,11),=CL18'SECURITY BYPASS ON'
         B     DOWTO             GO DO WTO
*
TAPEDSN  DS    ØH
         L     1,RCXFLAG2        GET FLAG2 ADDRESS
         TM    Ø(1),RCXATTRE     READ ATTEMPTED?
         BNO   NEXT1             NO, NEXT 1
         MVC   22(7,11),=CL7'READ' SPECIFY READ
         B     DOACCAL
NEXT1    DS    ØH
         TM    Ø(1),RCXATTUP     UPDATE ATTEMPTED?
         BNO   NEXT2             NO, NEXT 1
         MVC   22(7,11),=CL7'UPDATE' SPECIFY UDPATE
         B     DOACCAL
NEXT2    DS    ØH
         TM    Ø(1),RCXATTCO     CONTROL ATTEMPTED?
         BNO   NEXT3             NO, NEXT 1
         MVC   22(7,11),=CL7'CONTROL' SPECIFY CONTROL
         B     DOACCAL
NEXT3    DS    ØH
         MVC   22(7,11),=CL7'ALTER'   SPECIFY ALTER
DOACCAL  DS    ØH
         L     1,RCXACC          GET ACCESS ALLOWED FLAG
         TM    Ø(1),RCXNONE      NONE ALLOWED?
         BNO   ANEXTØ            NO, NEXT 1
         MVC   3Ø(7,11),=CL7'NONE' SPECIFY NONE
         B     DOWTO
ANEXTØ   DS    ØH
         TM    Ø(1),RCXREAD      READ ALLOWED?
         BNO   ANEXT1            NO, NEXT 1
         MVC   3Ø(7,11),=CL7'READ' SPECIFY READ
         B     DOWTO
ANEXT1   DS    ØH
         TM    Ø(1),RCXUPDAT     UPDATE ALLOWED?
```

```
         BNO   ANEXT2            NO, NEXT 1
         MVC   3Ø(7,11),=CL7'UPDATE' SPECIFY UDPATE
         B     DOWTO
ANEXT2   DS    ØH
         TM    Ø(1),RCXCONTR     CONTROL ALLOWED?
         BNO   ANEXT3            NO, NEXT 1
         MVC   3Ø(7,11),=CL7'CONTROL' SPECIFY CONTROL
         B     DOWTO
ANEXT3   DS    ØH
         MVC   3Ø(7,11),=CL7'ALTER' ALTER - THIS SHOULD NEVER HAPPEN
DOWTO    DS    ØH
         L     1,RCXRCODE        GET ADDRESS OF RETURN CODE
         XC    Ø(4,1),Ø(1)       SET RETURN CODE TO Ø
         WTO   MF=(E,(11))       DO THE WTO
FREEMAIN DS    ØH
         FREEMAIN R,LV=WTOL,A=(11)
*
EXIT     DS    ØH
         LM    14,12,12(13)
         SR    15,15
         BR    14
*
WTO WTO 'ICHRCXØ2 UUUUUUUU AAAAAAA/ZZZZZZZ ON PPPPPPPPPPQQQQQQQQQQRRRRRX
               RRRRRSSSSSSSSSTTTT',MF=L
WTOL     EQU   *-WTO
PELIST   DS    ØC
         DC    AL1(5),CL6'USERØ1'
         DC    AL1(5),CL6'USERØ2'
         DC    AL1(5),CL6'USERØ3'
         DC    XL1'FF'             END OF LIST
         LTORG
         ICHRCXP
         IHAACEE
         IHAASCB
         IHAASXB
         IHAPSA
         END
//LKED   EXEC  PGM=IEWL,PARM='MAP,LET,LIST,NCAL,AC=1,RENT',
//         COND=(Ø,LE,ASM)
//SYSLIN   DD  DSN=&&LOADSET,DISP=(OLD,DELETE)
//         DD  DDNAME=SYSIN
//SYSUT1   DD  UNIT=SYSALLDA,SPACE=(CYL,(3,2)),DSN=&SYSUT1
//SYSPRINT DD  SYSOUT=*
//SYSLMOD  DD  DISP=SHR,DSN=SYS1.LINKLIB(ICHRCXØ2)
```

Editor's note: Please address any comments on this article to the author at cshwang@hotmail.com.

Chorng S (Jack) Hwang
Principal
HSA Systems (USA)

Replacement for the RACF Report Writer – part 3

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.*

```
CHOUUNDF = 'Undefined user?'
CHOUERR  = 'Token in error?'
CHOUTRST = 'User trusted?'
CHOUSEST = 'Session type'
CHOUSURO = 'Surrogate user?'
CHOURMT  = 'Remote job?'
CHOUPRVL = 'Priviliged user?'
CHOUSECL = 'User SECLABEL'
CHOUEXND = 'Execution node'
CHOUSUSR = 'Submitting user'
CHOUSNOD = 'Submitting node'
CHOUSGRP = 'Submitting group'
CHOUSPOE = 'Port of entry'
CHOUSPCL = 'Class of POE'
CHOUTUSR = 'Userid'
CHOUTGRP = 'Groupid'
CHOUTDFT = 'Default group?'
CHOUTSEC = 'Default SECLABEL?'
CHOAPPC  = 'APPC key link'
CHOAUDIT = 'Audit code'
CHOORUID = 'Old real UID'
CHOOEUID = 'Old effective UID'
CHOOSUID = 'Old saved UID'
CHOORGID = 'Old real GID'
CHOOEGID = 'Old effective GID'
CHOOSGID = 'Old saved GID'
CHOPATHN = 'Path name'
CHOFILID = 'File id'
CHOFOUID = 'Owner UID'
CHOFOGID = 'Owner GID'
CHOUID   = 'New owner UID'
CHOGID   = 'New owner GID'
CHOFILPL = 'File pool'
```

```
      CHOFILSP = 'File space'
      CHOINODE = 'Inode'
      CHOSCID  = 'File SCID'
      CHODCELK = 'DCE link'
      CHOAUTYP = 'Request type'
      ;
        OUTPUT RACF.CHOWN;
      END;
    %END;
%MEND CHOWN;
./        ADD   LIST=ALL,NAME=CLRSETID
%MACRO CLRSETID(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
      %PUT Including variables from CLRSETID extension;
      RACF.CLRSETID (KEEP=%SMFHDR
                          %SMF8ØHDR(REQ=DEFINE)
                          CSICLASS
                          CSIUSERN
                          CSIUTKNE
                          CSIUPRE
                          CSIUVFYX
                          CSIUNJEU
                          CSIUUAUD
                          CSIUSPEC
                          CSIUDFLT
                          CSIUUNDF
                          CSIUERR
                          CSIUTRST
                          CSIUSEST
                          CSIUSURO
                          CSIURMT
                          CSIUPRVL
                          CSIUSECL
                          CSIUEXND
                          CSIUSUSR
                          CSIUSNOD
                          CSIUSGRP
                          CSIUSPOE
                          CSIUSPCL
                          CSIUTUSR
                          CSIUTGRP
                          CSIUTDFT
                          CSIUTSEC
                          CSIAPPC
                          CSIAUDIT
                          CSIORUID
                          CSIOEUID
                          CSIOSUID
                          CSIORGID
```

```
                        CSIOEGID
                        CSIOSGID
                        CSIPATHN
                        CSIFILID
                        CSIFOUID
                        CSIFOGID
                        CSIOLSGI
                        CSIOLSUI
                        CSIOLSVT
                        CSIOLORD
                        CSIOLOWR
                        CSIOLOEX
                        CSIOLGRD
                        CSIOLGWR
                        CSIOLGEX
                        CSIOLWRD
                        CSIOLWWR
                        CSIOLWEX
                        CSINWSGI
                        CSINWSUI
                        CSINWSVT
                        CSINWORD
                        CSINWOWR
                        CSINWOEX
                        CSINWGRD
                        CSINWGWR
                        CSINWGEX
                        CSINWWRD
                        CSINWWWR
                        CSINWWEX
                    )
  %END;
%IF &REQ = EXTRACT %THEN
  %DO;
    %PUT Including datadefinition for CLRSETID extension;
    WHEN('CLRSETID') DO;
      INPUT %SMF80HDR(REQ=EXTRACT)
            CSICLASS $       282-289
            CSIUSERN $       291-310
            CSIUTKNE $       312-315
            CSIUPRE  $       317-320
            CSIUVFYX $       322-325
            CSIUNJEU $       327-330
            CSIUUAUD $       332-335
            CSIUSPEC $       337-340
            CSIUDFLT $       342-345
            CSIUUNDF $       347-350
            CSIUERR  $       352-355
            CSIUTRST $       357-360
            CSIUSEST $       362-369
            CSIUSURO $       371-374
```

```
                CSIURMT  $       376-379
                CSIUPRVL $       381-384
                CSIUSECL $       386-393
                CSIUEXND $       395-402
                CSIUSUSR $       404-411
                CSIUSNOD $       413-420
                CSIUSGRP $       422-429
                CSIUSPOE $       431-438
                CSIUSPCL $       440-447
                CSIUTUSR $       449-456
                CSIUTGRP $       458-465
                CSIUTDFT $       467-470
                CSIUTSEC $       472-475
                CSIAPPC  $       477-492
                CSIAUDIT $       494-504
                CSIORUID         506-515
                CSIOEUID         517-526
                CSIOSUID         528-537
                CSIORGID         539-548
                CSIOEGID         550-559
                CSIOSGID         561-570
                CSIPATHN $       572-771
                CSIFILID $      1596-1627
                CSIFOUID        1629-1638
                CSIFOGID        1640-1649
                CSIOLSGI $      1651-1654
                CSIOLSUI $      1656-1659
                CSIOLSVT $      1661-1664
                CSIOLORD $      1666-1669
                CSIOLOWR $      1671-1674
                CSIOLOEX $      1676-1679
                CSIOLGRD $      1681-1684
                CSIOLGWR $      1686-1689
                CSIOLGEX $      1691-1694
                CSIOLWRD $      1696-1699
                CSIOLWWR $      1701-1704
                CSIOLWEX $      1706-1709
                CSINWSGI $      1711-1714
                CSINWSUI $      1716-1719
                CSINWSVT $      1721-1724
                CSINWORD $      1726-1729
                CSINWOWR $      1731-1734
                CSINWOEX $      1736-1739
                CSINWGRD $      1741-1744
                CSINWGWR $      1746-1749
                CSINWGEX $      1751-1754
                CSINWWRD $      1756-1759
                CSINWWWR $      1761-1764
                CSINWWEX $      1766-1769
                 ;
LABEL CSICLASS = 'Class name'
```

```
CSIUSERN = 'User name'
CSIUTKNE = 'Utoken encr.?'
CSIUPRE  = 'Pre-1.9?'
CSIUVFYX = 'VERIFYX propagation?'
CSIUNJEU = 'Undefined NJE user?'
CSIUUAUD = 'UAUDIT?'
CSIUSPEC = 'RACF special?'
CSIUDFLT = 'Default token?'
CSIUUNDF = 'Undefined user?'
CSIUERR  = 'Token in error?'
CSIUTRST = 'User trusted?'
CSIUSEST = 'Session type'
CSIUSURO = 'Surrogate user?'
CSIURMT  = 'Remote job?'
CSIUPRVL = 'Priviliged user?'
CSIUSECL = 'User SECLABEL'
CSIUEXND = 'Execution node'
CSIUSUSR = 'Submitting user'
CSIUSNOD = 'Submitting node'
CSIUSGRP = 'Submitting group'
CSIUSPOE = 'Port of entry'
CSIUSPCL = 'Class of POE'
CSIUTUSR = 'Userid'
CSIUTGRP = 'Groupid'
CSIUTDFT = 'Default group?'
CSIUTSEC = 'Default SECLABEL?'
CSIAPPC  = 'APPC key link'
CSIAUDIT = 'Audit code'
CSIORUID = 'Old real UID'
CSIOEUID = 'Old effective UID'
CSIOSUID = 'Old saved UID'
CSIORGID = 'Old real GID'
CSIOEGID = 'Old effective GID'
CSIOSGID = 'Old saved GID'
CSIPATHN = 'Path name'
CSIFILID = 'File id'
CSIFOUID = 'Owner UID'
CSIFOGID = 'Owner GID'
CSIOLSGI = 'Old S_ISGID requested?'
CSIOLSUI = 'Old S_ISUID requested?'
CSIOLSVT = 'Old S_ISVTX requested?'
CSIOLORD = 'Old Owner read?'
CSIOLOWR = 'Old Owner write?'
CSIOLOEX = 'Old Owner exec?'
CSIOLGRD = 'Old Group read?'
CSIOLGWR = 'Old Group write?'
CSIOLGEX = 'Old Group exec?'
CSIOLWRD = 'Old Other read?'
CSIOLWWR = 'Old Other write?'
CSIOLWEX = 'Old Other exec?'
CSINWSGI = 'New S_ISGID requested?'
```

```
      CSINWSUI = 'New S_ISUID requested?'
      CSINWSVT = 'New S_ISVTX requested?'
      CSINWORD = 'New Owner read?'
      CSINWOWR = 'New Owner write?'
      CSINWOEX = 'New Owner exec?'
      CSINWGRD = 'New Group read?'
      CSINWGWR = 'New Group write?'
      CSINWGEX = 'New Group exec?'
      CSINWWRD = 'New Other read?'
      CSINWWWR = 'New Other write?'
      CSINWWEX = 'New Other exec?'
      ;
        OUTPUT RACF.CLRSETID;
      END;
    %END;
%MEND CLRSETID;
./        ADD   LIST=ALL,NAME=EXESETID
%MACRO EXESETID(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
      %PUT Including variables from EXESETID extension;
      RACF.EXESETID (KEEP=%SMFHDR
                          %SMF8ØHDR(REQ=DEFINE)
                          ESICLASS
                          ESIUSERN
                          ESIUTKNE
                          ESIUPRE
                          ESIUVFYX
                          ESIUNJEU
                          ESIUUAUD
                          ESIUSPEC
                          ESIUDFLT
                          ESIUUNDF
                          ESIUERR
                          ESIUTRST
                          ESIUSEST
                          ESIUSURO
                          ESIURMT
                          ESIUPRVL
                          ESIUSECL
                          ESIUEXND
                          ESIUSUSR
                          ESIUSNOD
                          ESIUSGRP
                          ESIUSPOE
                          ESIUSPCL
                          ESIUTUSR
                          ESIUTGRP
                          ESIUTDFT
                          ESIUTSEC
```

```
                        ESIAPPC
                        ESIAUDIT
                        ESIORUID
                        ESIOEUID
                        ESIOSUID
                        ESIORGID
                        ESIOEGID
                        ESIOSGID
                        ESINRUID
                        ESINEUID
                        ESINSUID
                        ESINRGID
                        ESINEGID
                        ESINSGID
                    )
  %END;
%IF &REQ = EXTRACT %THEN
  %DO;
    %PUT Including datadefinition for EXESETID extension;
    WHEN('EXESETID') DO;
      INPUT %SMF80HDR(REQ=EXTRACT)
            ESICLASS $       282-289
            ESIUSERN $       291-310
            ESIUTKNE $       312-315
            ESIUPRE  $       317-320
            ESIUVFYX $       322-325
            ESIUNJEU $       327-330
            ESIUUAUD $       332-335
            ESIUSPEC $       337-340
            ESIUDFLT $       342-345
            ESIUUNDF $       347-350
            ESIUERR  $       352-355
            ESIUTRST $       357-360
            ESIUSEST $       362-369
            ESIUSURO $       371-374
            ESIURMT  $       376-379
            ESIUPRVL $       381-384
            ESIUSECL $       386-393
            ESIUEXND $       395-402
            ESIUSUSR $       404-411
            ESIUSNOD $       413-420
            ESIUSGRP $       422-429
            ESIUSPOE $       431-438
            ESIUSPCL $       440-447
            ESIUTUSR $       449-456
            ESIUTGRP $       458-465
            ESIUTDFT $       467-470
            ESIUTSEC $       472-475
            ESIAPPC  $       477-492
```

```
                ESIAUDIT $       494-504
                ESIORUID         506-515
                ESIOEUID         517-526
                ESIOSUID         528-537
                ESIORGID         539-548
                ESIOEGID         550-559
                ESIOSGID         561-570
                ESINRUID         572-581
                ESINEUID         583-592
                ESINSUID         594-603
                ESINRGID         605-614
                ESINEGID         616-625
                ESINSGID         627-636
                 ;
LABEL ESICLASS = 'Class name'
      ESIUSERN = 'User name'
      ESIUTKNE = 'Utoken encr.?'
      ESIUPRE  = 'Pre-1.9?'
      ESIUVFYX = 'VERIFYX propagation?'
      ESIUNJEU = 'Undefined NJE user?'
      ESIUUAUD = 'UAUDIT?'
      ESIUSPEC = 'RACF special?'
      ESIUDFLT = 'Default token?'
      ESIUUNDF = 'Undefined user?'
      ESIUERR  = 'Token in error?'
      ESIUTRST = 'User trusted?'
      ESIUSEST = 'Session type'
      ESIUSURO = 'Surrogate user?'
      ESIURMT  = 'Remote job?'
      ESIUPRVL = 'Priviliged user?'
      ESIUSECL = 'User SECLABEL'
      ESIUEXND = 'Execution node'
      ESIUSUSR = 'Submitting user'
      ESIUSNOD = 'Submitting node'
      ESIUSGRP = 'Submitting group'
      ESIUSPOE = 'Port of entry'
      ESIUSPCL = 'Class of POE'
      ESIUTUSR = 'Userid'
      ESIUTGRP = 'Groupid'
      ESIUTDFT = 'Default group?'
      ESIUTSEC = 'Default SECLABEL?'
      ESIAPPC  = 'APPC key link'
      ESIAUDIT = 'Audit code'
      ESIORUID = 'Old real UID'
      ESIOEUID = 'Old effective UID'
      ESIOSUID = 'Old saved UID'
      ESIORGID = 'Old real GID'
      ESIOEGID = 'Old effective GID'
      ESIOSGID = 'Old saved GID'
      ESINRUID = 'New real UID'
      ESINEUID = 'New effective UID'
```

```
      ESINSUID = 'New saved UID'
      ESINRGID = 'New real GID'
      ESINEGID = 'New effective GID'
      ESINSGID = 'New saved GID'
      ;
        OUTPUT RACF.EXESETID;
      END;
    %END;
%MEND EXESETID;
./        ADD   LIST=ALL,NAME=GETPSENT
%MACRO GETPSENT(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
      %PUT Including variables from GETPSENT extension;
      RACF.GETPSENT (KEEP=%SMFHDR
                          %SMF8ØHDR(REQ=DEFINE)
                          GPSCLASS
                          GPSUSERN
                          GPSUTKNE
                          GPSUPRE
                          GPSUVFYX
                          GPSUNJEU
                          GPSUUAUD
                          GPSUSPEC
                          GPSUDFLT
                          GPSUUNDF
                          GPSUERR
                          GPSUTRST
                          GPSUSEST
                          GPSUSURO
                          GPSURMT
                          GPSUPRVL
                          GPSUSECL
                          GPSUEXND
                          GPSUSUSR
                          GPSUSNOD
                          GPSUSGRP
                          GPSUSPOE
                          GPSUSPCL
                          GPSUTUSR
                          GPSUTGRP
                          GPSUTDFT
                          GPSUTSEC
                          GPSAPPC
                          GPSAUDIT
                          GPSORUID
                          GPSOEUID
                          GPSOSUID
                          GPSORGID
                          GPSOEGID
```

```
                        GPSOSGID
                        GPSTRUID
                        GPSTEUID
                        GPSTSUID
                        GPSTPID
                    )
  %END;
%IF &REQ = EXTRACT %THEN
  %DO;
    %PUT Including datadefinition for GETPSENT extension;
    WHEN('GETPSENT') DO;
      INPUT %SMF80HDR(REQ=EXTRACT)
            GPSCLASS $       282-289
            GPSUSERN $       291-310
            GPSUTKNE $       312-315
            GPSUPRE  $       317-320
            GPSUVFYX $       322-325
            GPSUNJEU $       327-330
            GPSUUAUD $       332-335
            GPSUSPEC $       337-340
            GPSUDFLT $       342-345
            GPSUUNDF $       347-350
            GPSUERR  $       352-355
            GPSUTRST $       357-360
            GPSUSEST $       362-369
            GPSUSURO $       371-374
            GPSURMT  $       376-379
            GPSUPRVL $       381-384
            GPSUSECL $       386-393
            GPSUEXND $       395-402
            GPSUSUSR $       404-411
            GPSUSNOD $       413-420
            GPSUSGRP $       422-429
            GPSUSPOE $       431-438
            GPSUSPCL $       440-447
            GPSUTUSR $       449-456
            GPSUTGRP $       458-465
            GPSUTDFT $       467-470
            GPSUTSEC $       472-475
            GPSAPPC  $       477-492
            GPSAUDIT $       494-504
            GPSORUID         506-515
            GPSOEUID         517-526
            GPSOSUID         528-537
            GPSORGID         539-548
            GPSOEGID         550-559
            GPSOSGID         561-570
            GPSTRUID         572-581
            GPSTEUID         583-592
            GPSTSUID         594-603
            GPSTPID          605-614
```

```
                ;
LABEL GPSCLASS = 'Class name'
      GPSUSERN = 'User name'
      GPSUTKNE = 'Utoken encr.?'
      GPSUPRE  = 'Pre-1.9?'
      GPSUVFYX = 'VERIFYX propagation?'
      GPSUNJEU = 'Undefined NJE user?'
      GPSUUAUD = 'UAUDIT?'
      GPSUSPEC = 'RACF special?'
      GPSUDFLT = 'Default token?'
      GPSUUNDF = 'Undefined user?'
      GPSUERR  = 'Token in error?'
      GPSUTRST = 'User trusted?'
      GPSUSEST = 'Session type'
      GPSUSURO = 'Surrogate user?'
      GPSURMT  = 'Remote job?'
      GPSUPRVL = 'Priviliged user?'
      GPSUSECL = 'User SECLABEL'
      GPSUEXND = 'Execution node'
      GPSUSUSR = 'Submitting user'
      GPSUSNOD = 'Submitting node'
      GPSUSGRP = 'Submitting group'
      GPSUSPOE = 'Port of entry'
      GPSUSPCL = 'Class of POE'
      GPSUTUSR = 'Userid'
      GPSUTGRP = 'Groupid'
      GPSUTDFT = 'Default group?'
      GPSUTSEC = 'Default SECLABEL?'
      GPSAPPC  = 'APPC key link'
      GPSAUDIT = 'Audit code'
      GPSORUID = 'Old real UID'
      GPSOEUID = 'Old effective UID'
      GPSOSUID = 'Old saved UID'
      GPSORGID = 'Old real GID'
      GPSOEGID = 'Old effective GID'
      GPSOSGID = 'Old saved GID'
      GPSTRUID = 'Tgt. real UID'
      GPSTEUID = 'Tgt. effective UID'
      GPSTSUID = 'Tgt. saved UID'
      GPSTPID  = 'Tgt. process ID'
      ;
        OUTPUT RACF.GETPSENT;
      END;
    %END;
%MEND GETPSENT;
./        ADD   LIST=ALL,NAME=INITOEDP
%MACRO INITOEDP(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
      %PUT Including variables from INITOEDP extension;
```

```
    RACF.INITOEDP (KEEP=%SMFHDR
                        %SMF8ØHDR(REQ=DEFINE)
                        IOECLASS
                        IOEUSERN
                        IOEUTKNE
                        IOEUPRE
                        IOEUVFYX
                        IOEUNJEU
                        IOEUUAUD
                        IOEUSPEC
                        IOEUDFLT
                        IOEUUNDF
                        IOEUERR
                        IOEUTRST
                        IOEUSEST
                        IOEUSURO
                        IOEURMT
                        IOEUPRVL
                        IOEUSECL
                        IOEUEXND
                        IOEUSUSR
                        IOEUSNOD
                        IOEUSGRP
                        IOEUSPOE
                        IOEUSPCL
                        IOEUTUSR
                        IOEUTGRP
                        IOEUTDFT
                        IOEUTSEC
                        IOEAPPC
                        IOEAUDIT
                        IOEORUID
                        IOEOEUID
                        IOEOSUID
                        IOEORGID
                        IOEOEGID
                        IOEOSGID
                   )
  %END;
%IF &REQ = EXTRACT %THEN
  %DO;
    %PUT Including datadefinition for INITOEDP extension;
    WHEN('INITOEDP') DO;
      INPUT %SMF8ØHDR(REQ=EXTRACT)
            IOECLASS $       282-289
            IOEUSERN $       291-31Ø
            IOEUTKNE $       312-315
            IOEUPRE  $       317-32Ø
            IOEUVFYX $       322-325
            IOEUNJEU $       327-33Ø
            IOEUUAUD $       332-335
```

```
                IOEUSPEC $       337-340
                IOEUDFLT $       342-345
                IOEUUNDF $       347-350
                IOEUERR  $       352-355
                IOEUTRST $       357-360
                IOEUSEST $       362-369
                IOEUSURO $       371-374
                IOEURMT  $       376-379
                IOEUPRVL $       381-384
                IOEUSECL $       386-393
                IOEUEXND $       395-402
                IOEUSUSR $       404-411
                IOEUSNOD $       413-420
                IOEUSGRP $       422-429
                IOEUSPOE $       431-438
                IOEUSPCL $       440-447
                IOEUTUSR $       449-456
                IOEUTGRP $       458-465
                IOEUTDFT $       467-470
                IOEUTSEC $       472-475
                IOEAPPC  $       477-492
                IOEAUDIT $       494-504
                IOEORUID         506-515
                IOEOEUID         517-526
                IOEOSUID         528-537
                IOEORGID         539-548
                IOEOEGID         550-559
                IOEOSGID         561-570
                 ;
LABEL IOECLASS = 'Class name'
      IOEUSERN = 'User name'
      IOEUTKNE = 'Utoken encr.?'
      IOEUPRE  = 'Pre-1.9?'
      IOEUVFYX = 'VERIFYX propagation?'
      IOEUNJEU = 'Undefined NJE user?'
      IOEUUAUD = 'UAUDIT?'
      IOEUSPEC = 'RACF special?'
      IOEUDFLT = 'Default token?'
      IOEUUNDF = 'Undefined user?'
      IOEUERR  = 'Token in error?'
      IOEUTRST = 'User trusted?'
      IOEUSEST = 'Session type'
      IOEUSURO = 'Surrogate user?'
      IOEURMT  = 'Remote job?'
      IOEUPRVL = 'Priviliged user?'
      IOEUSECL = 'User SECLABEL'
      IOEUEXND = 'Execution node'
      IOEUSUSR = 'Submitting user'
      IOEUSNOD = 'Submitting node'
      IOEUSGRP = 'Submitting group'
```

```
      IOEUSPOE = 'Port of entry'
      IOEUSPCL = 'Class of POE'
      IOEUTUSR = 'Userid'
      IOEUTGRP = 'Groupid'
      IOEUTDFT = 'Default group?'
      IOEUTSEC = 'Default SECLABEL?'
      IOEAPPC  = 'APPC key link'
      IOEAUDIT = 'Audit code'
      IOEORUID = 'Old real UID'
      IOEOEUID = 'Old effective UID'
      IOEOSUID = 'Old saved UID'
      IOEORGID = 'Old real GID'
      IOEOEGID = 'Old effective GID'
      IOEOSGID = 'Old saved GID'
      ;
        OUTPUT RACF.INITOEDP;
      END;
    %END;
%MEND INITOEDP;
./        ADD   LIST=ALL,NAME=TERMOEDP
%MACRO TERMOEDP(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
      %PUT Including variables from TERMOEDP extension;
      RACF.TERMOEDP (KEEP=%SMFHDR
                          %SMF8ØHDR(REQ=DEFINE)
                          TOECLASS
                          TOEUSERN
                          TOEUTKNE
                          TOEUPRE
                          TOEUVFYX
                          TOEUNJEU
                          TOEUUAUD
                          TOEUSPEC
                          TOEUDFLT
                          TOEUUNDF
                          TOEUERR
                          TOEUTRST
                          TOEUSEST
                          TOEUSURO
                          TOEURMT
                          TOEUPRVL
                          TOEUSECL
                          TOEUEXND
                          TOEUSUSR
                          TOEUSNOD
                          TOEUSGRP
                          TOEUSPOE
                          TOEUSPCL
```

```
                        TOEUTUSR
                        TOEUTGRP
                        TOEUTDFT
                        TOEUTSEC
                        TOEAPPC
                        TOEAUDIT
                        TOEORUID
                        TOEOEUID
                        TOEOSUID
                        TOEORGID
                        TOEOEGID
                        TOEOSGID
                    )
  %END;
%IF &REQ = EXTRACT %THEN
  %DO;
    %PUT Including datadefinition for TERMOEDP extension;
    WHEN('TERMOEDP') DO;
      INPUT %SMF80HDR(REQ=EXTRACT)
            TOECLASS $       282-289
            TOEUSERN $       291-310
            TOEUTKNE $       312-315
            TOEUPRE  $       317-320
            TOEUVFYX $       322-325
            TOEUNJEU $       327-330
            TOEUUAUD $       332-335
            TOEUSPEC $       337-340
            TOEUDFLT $       342-345
            TOEUUNDF $       347-350
            TOEUERR  $       352-355
            TOEUTRST $       357-360
            TOEUSEST $       362-369
            TOEUSURO $       371-374
            TOEURMT  $       376-379
            TOEUPRVL $       381-384
            TOEUSECL $       386-393
            TOEUEXND $       395-402
            TOEUSUSR $       404-411
            TOEUSNOD $       413-420
            TOEUSGRP $       422-429
            TOEUSPOE $       431-438
            TOEUSPCL $       440-447
            TOEUTUSR $       449-456
            TOEUTGRP $       458-465
            TOEUTDFT $       467-470
            TOEUTSEC $       472-475
            TOEAPPC  $       477-492
            TOEAUDIT $       494-504
            TOEORUID         506-515
            TOEOEUID         517-526
```

```
              TOEOSUID         528-537
              TOEORGID         539-548
              TOEOEGID         55Ø-559
              TOEOSGID         561-57Ø
               ;
LABEL TOECLASS = 'Class name'
      TOEUSERN = 'User name'
      TOEUTKNE = 'Utoken encr.?'
      TOEUPRE  = 'Pre-1.9?'
      TOEUVFYX = 'VERIFYX propagation?'
      TOEUNJEU = 'Undefined NJE user?'
      TOEUUAUD = 'UAUDIT?'
      TOEUSPEC = 'RACF special?'
      TOEUDFLT = 'Default token?'
      TOEUUNDF = 'Undefined user?'
      TOEUERR  = 'Token in error?'
      TOEUTRST = 'User trusted?'
      TOEUSEST = 'Session type'
      TOEUSURO = 'Surrogate user?'
      TOEURMT  = 'Remote job?'
      TOEUPRVL = 'Priviliged user?'
      TOEUSECL = 'User SECLABEL'
      TOEUEXND = 'Execution node'
      TOEUSUSR = 'Submitting user'
      TOEUSNOD = 'Submitting node'
      TOEUSGRP = 'Submitting group'
      TOEUSPOE = 'Port of entry'
      TOEUSPCL = 'Class of POE'
      TOEUTUSR = 'Userid'
      TOEUTGRP = 'Groupid'
      TOEUTDFT = 'Default group?'
      TOEUTSEC = 'Default SECLABEL?'
      TOEAPPC  = 'APPC key link'
      TOEAUDIT = 'Audit code'
      TOEORUID = 'Old real UID'
      TOEOEUID = 'Old effective UID'
      TOEOSUID = 'Old saved UID'
      TOEORGID = 'Old real GID'
      TOEOEGID = 'Old effective GID'
      TOEOSGID = 'Old saved GID'
      ;
        OUTPUT RACF.TERMOEDP;
      END;
    %END;
%MEND TERMOEDP;
./        ADD   LIST=ALL,NAME=KILL
%MACRO KILL(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
      %PUT Including variables from KILL extension;
```

```
    RACF.KILL (KEEP=%SMFHDR
                    %SMF8ØHDR(REQ=DEFINE)
                    KILCLASS
                    KILUSERN
                    KILUTKNE
                    KILUPRE
                    KILUVFYX
                    KILUNJEU
                    KILUUAUD
                    KILUSPEC
                    KILUDFLT
                    KILUUNDF
                    KILUERR
                    KILUTRST
                    KILUSEST
                    KILUSURO
                    KILURMT
                    KILUPRVL
                    KILUSECL
                    KILUEXND
                    KILUSUSR
                    KILUSNOD
                    KILUSGRP
                    KILUSPOE
                    KILUSPCL
                    KILUTUSR
                    KILUTGRP
                    KILUTDFT
                    KILUTSEC
                    KILAPPC
                    KILAUDIT
                    KILORUID
                    KILOEUID
                    KILOSUID
                    KILORGID
                    KILOEGID
                    KILOSGID
                    KILTRUID
                    KILTEUID
                    KILTSUID
                    KILTPID
                    KILSGNAL
               )
  %END;
%IF &REQ = EXTRACT %THEN
  %DO;
    %PUT Including datadefinition for KILL extension;
    WHEN('KILL') DO;
      INPUT %SMF8ØHDR(REQ=EXTRACT)
            KILCLASS $       282-289
```

```
                KILUSERN $       291-310
                KILUTKNE $       312-315
                KILUPRE  $       317-320
                KILUVFYX $       322-325
                KILUNJEU $       327-330
                KILUUAUD $       332-335
                KILUSPEC $       337-340
                KILUDFLT $       342-345
                KILUUNDF $       347-350
                KILUERR  $       352-355
                KILUTRST $       357-360
                KILUSEST $       362-369
                KILUSURO $       371-374
                KILURMT  $       376-379
                KILUPRVL $       381-384
                KILUSECL $       386-393
                KILUEXND $       395-402
                KILUSUSR $       404-411
                KILUSNOD $       413-420
                KILUSGRP $       422-429
                KILUSPOE $       431-438
                KILUSPCL $       440-447
                KILUTUSR $       449-456
                KILUTGRP $       458-465
                KILUTDFT $       467-470
                KILUTSEC $       472-475
                KILAPPC  $       477-492
                KILAUDIT $       494-504
                KILORUID         506-515
                KILOEUID         517-526
                KILOSUID         528-537
                KILORGID         539-548
                KILOEGID         550-559
                KILOSGID         561-570
                KILTRUID         572-581
                KILTEUID         583-592
                KILTSUID         594-603
                KILTPID          605-614
                KILSGNAL         616-625
                 ;
LABEL KILCLASS = 'Class name'
      KILUSERN = 'User name'
      KILUTKNE = 'Utoken encr.?'
      KILUPRE  = 'Pre-1.9?'
      KILUVFYX = 'VERIFYX propagation?'
      KILUNJEU = 'Undefined NJE user?'
      KILUUAUD = 'UAUDIT?'
      KILUSPEC = 'RACF special?'
      KILUDFLT = 'Default token?'
      KILUUNDF = 'Undefined user?'
```

```
      KILUERR  = 'Token in error?'
      KILUTRST = 'User trusted?'
      KILUSEST = 'Session type'
      KILUSURO = 'Surrogate user?'
      KILURMT  = 'Remote job?'
      KILUPRVL = 'Priviliged user?'
      KILUSECL = 'User SECLABEL'
      KILUEXND = 'Execution node'
      KILUSUSR = 'Submitting user'
      KILUSNOD = 'Submitting node'
      KILUSGRP = 'Submitting group'
      KILUSPOE = 'Port of entry'
      KILUSPCL = 'Class of POE'
      KILUTUSR = 'Userid'
      KILUTGRP = 'Groupid'
      KILUTDFT = 'Default group?'
      KILUTSEC = 'Default SECLABEL?'
      KILAPPC  = 'APPC key link'
      KILAUDIT = 'Audit code'
      KILORUID = 'Old real UID'
      KILOEUID = 'Old effective UID'
      KILOSUID = 'Old saved UID'
      KILORGID = 'Old real GID'
      KILOEGID = 'Old effective GID'
      KILOSGID = 'Old saved GID'
      KILTRUID = 'Tgt. real UID'
      KILTEUID = 'Tgt. effective UID'
      KILTSUID = 'Tgt. saved UID'
      KILTPID  = 'Tgt. process ID'
      KILSGNAL = 'Kill signal code'
      ;
        OUTPUT RACF.KILL;
      END;
    %END;
%MEND KILL;
./        ADD   LIST=ALL,NAME=LINK
%MACRO LINK(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
      %PUT Including variables from LINK extension;
      RACF.LINK (KEEP=%SMFHDR
                      %SMF8ØHDR(REQ=DEFINE)
                      CHDCLASS
                      CHDUSERN
                      CHDUTKNE
                      CHDUPRE
                      CHDUVFYX
                      CHDUNJEU
                      CHDUUAUD
                      CHDUSPEC
```

```
                    CHDUDFLT
                    CHDUUNDF
                    CHDUERR
                    CHDUTRST
                    CHDUSEST
                    CHDUSURO
                    CHDURMT
                    CHDUPRVL
                    CHDUSECL
                    CHDUEXND
                    CHDUSUSR
                    CHDUSNOD
                    CHDUSGRP
                    CHDUSPOE
                    CHDUSPCL
                    CHDUTUSR
                    CHDUTGRP
                    CHDUTDFT
                    CHDUTSEC
                    CHDAPPC
                    CHDAUDIT
                    CHDORUID
                    CHDOEUID
                    CHDOSUID
                    CHDORGID
                    CHDOEGID
                    CHDOSGID
                    CHDPATHN
                    CHDFILID
                    CHDFOUID
                    CHDFOGID
                    CHDREQP2
                    CHDPTHTP
                    CHDFILPL
                    CHDFILSP
                    CHDINODE
                    CHDSCID
                    CHDDCELK
                    CHDAUTYP
                )
  %END;
%IF &REQ = EXTRACT %THEN
  %DO;
    %PUT Including datadefinition for LINK extension;
    WHEN('LINK') DO;
      INPUT %SMF8ØHDR(REQ=EXTRACT)
            CHDCLASS $       282-289
            CHDUSERN $       291-31Ø
            CHDUTKNE $       312-315
            CHDUPRE  $       317-32Ø
            CHDUVFYX $       322-325
```

```
                CHDUNJEU $       327-330
                CHDUUAUD $       332-335
                CHDUSPEC $       337-340
                CHDUDFLT $       342-345
                CHDUUNDF $       347-350
                CHDUERR  $       352-355
                CHDUTRST $       357-360
                CHDUSEST $       362-369
                CHDUSURO $       371-374
                CHDURMT  $       376-379
                CHDUPRVL $       381-384
                CHDUSECL $       386-393
                CHDUEXND $       395-402
                CHDUSUSR $       404-411
                CHDUSNOD $       413-420
                CHDUSGRP $       422-429
                CHDUSPOE $       431-438
                CHDUSPCL $       440-447
                CHDUTUSR $       449-456
                CHDUTGRP $       458-465
                CHDUTDFT $       467-470
                CHDUTSEC $       472-475
                CHDAPPC  $       477-492
                CHDAUDIT $       494-504
                CHDORUID         506-515
                CHDOEUID         517-526
                CHDOSUID         528-537
                CHDORGID         539-548
                CHDOEGID         550-559
                CHDOSGID         561-570
                CHDPATHN $       572-771
                CHDFILID $      1596-1627
                CHDFOUID        1629-1638
                CHDFOGID        1640-1649
                CHDREQP2 $      1651-1850
                CHDPTHTP $      2675-2678
                CHDFILPL $      2680-2687
                CHDFILSP $      2689-2696
                CHDINODE        2698-2707
                CHDSCID         2709-2718
                CHDDCELK $      2720-2735
                CHDAUTYP $      2737-2749
                 ;
LABEL CHDCLASS = 'Class name'
      CHDUSERN = 'User name'
      CHDUTKNE = 'Utoken encr.?'
      CHDUPRE  = 'Pre-1.9?'
      CHDUVFYX = 'VERIFYX propagation?'
      CHDUNJEU = 'Undefined NJE user?'
      CHDUUAUD = 'UAUDIT?'
```

```
      CHDUSPEC = 'RACF special?'
      CHDUDFLT = 'Default token?'
      CHDUUNDF = 'Undefined user?'
      CHDUERR  = 'Token in error?'
      CHDUTRST = 'User trusted?'
      CHDUSEST = 'Session type'
      CHDUSURO = 'Surrogate user?'
      CHDURMT  = 'Remote job?'
      CHDUPRVL = 'Priviliged user?'
      CHDUSECL = 'User SECLABEL'
      CHDUEXND = 'Execution node'
      CHDUSUSR = 'Submitting user'
      CHDUSNOD = 'Submitting node'
      CHDUSGRP = 'Submitting group'
      CHDUSPOE = 'Port of entry'
      CHDUSPCL = 'Class of POE'
      CHDUTUSR = 'Userid'
      CHDUTGRP = 'Groupid'
      CHDUTDFT = 'Default group?'
      CHDUTSEC = 'Default SECLABEL?'
      CHDAPPC  = 'APPC key link'
      CHDAUDIT = 'Audit code'
      CHDORUID = 'Old real UID'
      CHDOEUID = 'Old effective UID'
      CHDOSUID = 'Old saved UID'
      CHDORGID = 'Old real GID'
      CHDOEGID = 'Old effective GID'
      CHDOSGID = 'Old saved GID'
      CHDPATHN = 'Path name'
      CHDFILID = 'File id'
      CHDFOUID = 'Owner UID'
      CHDFOGID = 'Owner GID'
      CHDREQP2 = '2nd path name'
      CHDPTHTP = 'Path type'
      CHDFILPL = 'File pool'
      CHDFILSP = 'File space'
      CHDINODE = 'Inode'
      CHDSCID  = 'File SCID'
      CHDDCELK = 'DCE link'
      CHDAUTYP = 'Request type'
      ;
        OUTPUT RACF.LINK;
      END;
    %END;
%MEND LINK;
./        ADD   LIST=ALL,NAME=MKDIR
%MACRO MKDIR(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
      %PUT Including variables from MKDIR extension;
```

```
RACF.MKDIR (KEEP=%SMFHDR
                 %SMF8ØHDR(REQ=DEFINE)
                 MKDCLASS
                 MKDUSERN
                 MKDUTKNE
                 MKDUPRE
                 MKDUVFYX
                 MKDUNJEU
                 MKDUUAUD
                 MKDUSPEC
                 MKDUDFLT
                 MKDUUNDF
                 MKDUERR
                 MKDUTRST
                 MKDUSEST
                 MKDUSURO
                 MKDURMT
                 MKDUPRVL
                 MKDUSECL
                 MKDUEXND
                 MKDUSUSR
                 MKDUSNOD
                 MKDUSGRP
                 MKDUSPOE
                 MKDUSPCL
                 MKDUTUSR
                 MKDUTGRP
                 MKDUTDFT
                 MKDUTSEC
                 MKDAPPC
                 MKDAUDIT
                 MKDORUID
                 MKDOEUID
                 MKDOSUID
                 MKDORGID
                 MKDOEGID
                 MKDOSGID
                 MKDPATHN
                 MKDFILID
                 MKDFOUID
                 MKDFOGID
                 MKDOLSGI
                 MKDOLSUI
                 MKDOLSVT
                 MKDOLORD
                 MKDOLOWR
                 MKDOLOEX
                 MKDOLGRD
                 MKDOLGWR
                 MKDOLGEX
                 MKDOLWRD
```

```
                    MKDOLWWR
                    MKDOLWEX
                    MKDNWSGI
                    MKDNWSUI
                    MKDNWSVT
                    MKDNWORD
                    MKDNWOWR
                    MKDNWOEX
                    MKDNWGRD
                    MKDNWGWR
                    MKDNWGEX
                    MKDNWWRD
                    MKDNWWWR
                    MKDNWWEX
                    MKDNWURE
                    MKDNWUWR
                    MKDNWUEX
                    MKDNWARE
                    MKDNWAWR
                    MKDNWAEX
                    MKDRQSGI
                    MKDRQSUI
                    MKDRQSVT
                    MKDRQORD
                    MKDRQOWR
                    MKDRQOEX
                    MKDRQGRD
                    MKDRQGWR
                    MKDRQGEX
                    MKDRQWRD
                    MKDRQWWR
                    MKDRQWEX
                    MKDFILPL
                    MKDFILSP
                    MKDINODE
                    MKDSCID
                )
  %END;
%IF &REQ = EXTRACT %THEN
  %DO;
    %PUT Including datadefinition for MKDIR extension;
    WHEN('MKDIR') DO;
      INPUT %SMF8ØHDR(REQ=EXTRACT)
            MKDCLASS $       282-289
            MKDUSERN $       291-31Ø
            MKDUTKNE $       312-315
            MKDUPRE  $       317-32Ø
            MKDUVFYX $       322-325
            MKDUNJEU $       327-33Ø
            MKDUUAUD $       332-335
```

```
MKDUSPEC $       337-340
MKDUDFLT $       342-345
MKDUUNDF $       347-350
MKDUERR  $       352-355
MKDUTRST $       357-360
MKDUSEST $       362-369
MKDUSURO $       371-374
MKDURMT  $       376-379
MKDUPRVL $       381-384
MKDUSECL $       386-393
MKDUEXND $       395-402
MKDUSUSR $       404-411
MKDUSNOD $       413-420
MKDUSGRP $       422-429
MKDUSPOE $       431-438
MKDUSPCL $       440-447
MKDUTUSR $       449-456
MKDUTGRP $       458-465
MKDUTDFT $       467-470
MKDUTSEC $       472-475
MKDAPPC  $       477-492
MKDAUDIT $       494-504
MKDORUID         506-515
MKDOEUID         517-526
MKDOSUID         528-537
MKDORGID         539-548
MKDOEGID         550-559
MKDOSGID         561-570
MKDPATHN $       572-771
MKDFILID $      1596-1627
MKDFOUID        1629-1638
MKDFOGID        1640-1649
MKDOLSGI $      1651-1654
MKDOLSUI $      1656-1659
MKDOLSVT $      1661-1664
MKDOLORD $      1666-1669
MKDOLOWR $      1671-1674
MKDOLOEX $      1676-1679
MKDOLGRD $      1681-1684
MKDOLGWR $      1686-1689
MKDOLGEX $      1691-1694
MKDOLWRD $      1696-1699
MKDOLWWR $      1701-1704
MKDOLWEX $      1706-1709
MKDNWSGI $      1711-1714
MKDNWSUI $      1716-1719
MKDNWSVT $      1721-1724
MKDNWORD $      1726-1729
MKDNWOWR $      1731-1734
MKDNWOEX $      1736-1739
MKDNWGRD $      1741-1744
```

```
                MKDNWGWR $      1746-1749
                MKDNWGEX $      1751-1754
                MKDNWWRD $      1756-1759
                MKDNWWWR $      1761-1764
                MKDNWWEX $      1766-1769
                MKDNWURE $      1771-1778
                MKDNWUWR $      178Ø-1787
                MKDNWUEX $      1789-1796
                MKDNWARE $      1798-18Ø5
                MKDNWAWR $      18Ø7-1814
                MKDNWAEX $      1816-1823
                MKDRQSGI $      1825-1828
                MKDRQSUI $      183Ø-1833
                MKDRQSVT $      1835-1838
                MKDRQORD $      184Ø-1843
                MKDRQOWR $      1845-1848
                MKDRQOEX $      185Ø-1853
                MKDRQGRD $      1855-1858
                MKDRQGWR $      186Ø-1863
                MKDRQGEX $      1865-1868
                MKDRQWRD $      187Ø-1873
                MKDRQWWR $      1875-1878
                MKDRQWEX $      188Ø-1883
                MKDFILPL $      1885-1892
                MKDFILSP $      1894-19Ø1
                MKDINODE        19Ø3-1912
                MKDSCID         1914-1923
                 ;
LABEL MKDCLASS = 'Class name'
      MKDUSERN = 'User name'
      MKDUTKNE = 'Utoken encr.?'
      MKDUPRE  = 'Pre-1.9?'
      MKDUVFYX = 'VERIFYX propagation?'
      MKDUNJEU = 'Undefined NJE user?'
      MKDUUAUD = 'UAUDIT?'
      MKDUSPEC = 'RACF special?'
      MKDUDFLT = 'Default token?'
      MKDUUNDF = 'Undefined user?'
      MKDUERR  = 'Token in error?'
      MKDUTRST = 'User trusted?'
      MKDUSEST = 'Session type'
      MKDUSURO = 'Surrogate user?'
      MKDURMT  = 'Remote job?'
      MKDUPRVL = 'Priviliged user?'
      MKDUSECL = 'User SECLABEL'
      MKDUEXND = 'Execution node'
      MKDUSUSR = 'Submitting user'
      MKDUSNOD = 'Submitting node'
      MKDUSGRP = 'Submitting group'
      MKDUSPOE = 'Port of entry'
      MKDUSPCL = 'Class of POE'
```

```
MKDUTUSR = 'Userid'
MKDUTGRP = 'Groupid'
MKDUTDFT = 'Default group?'
MKDUTSEC = 'Default SECLABEL?'
MKDAPPC  = 'APPC key link'
MKDAUDIT = 'Audit code'
MKDORUID = 'Old real UID'
MKDOEUID = 'Old effective UID'
MKDOSUID = 'Old saved UID'
MKDORGID = 'Old real GID'
MKDOEGID = 'Old effective GID'
MKDOSGID = 'Old saved GID'
MKDPATHN = 'Path name'
MKDFILID = 'File id'
MKDFOUID = 'Owner UID'
MKDFOGID = 'Owner GID'
MKDOLSGI = 'Old S_ISGID requested?'
MKDOLSUI = 'Old S_ISUID requested?'
MKDOLSVT = 'Old S_ISVTX requested?'
MKDOLORD = 'Old Owner read?'
MKDOLOWR = 'Old Owner write?'
MKDOLOEX = 'Old Owner exec?'
MKDOLGRD = 'Old Group read?'
MKDOLGWR = 'Old Group write?'
MKDOLGEX = 'Old Group exec?'
MKDOLWRD = 'Old Other read?'
MKDOLWWR = 'Old Other write?'
MKDOLWEX = 'Old Other exec?'
MKDNWSGI = 'New S_ISGID requested?'
MKDNWSUI = 'New S_ISUID requested?'
MKDNWSVT = 'New S_ISVTX requested?'
MKDNWORD = 'New Owner read?'
MKDNWOWR = 'New Owner write?'
MKDNWOEX = 'New Owner exec?'
MKDNWGRD = 'New Group read?'
MKDNWGWR = 'New Group write?'
MKDNWGEX = 'New Group exec?'
MKDNWWRD = 'New Other read?'
MKDNWWWR = 'New Other write?'
MKDNWWEX = 'New Other exec?'
MKDNWURE = 'New user aud read'
MKDNWUWR = 'New user aud write'
MKDNWUEX = 'New user aud exec'
MKDNWARE = 'New auditor aud read'
MKDNWAWR = 'New auditor aud write'
MKDNWAEX = 'New auditor aud exec'
MKDRQSGI = 'Req S_ISGID?'
MKDRQSUI = 'Req S_ISUID?'
MKDRQSVT = 'Req S_ISVTX?'
MKDRQORD = 'Req Owner read?'
```

```
      MKDRQOWR = 'Req Owner write?'
      MKDRQOEX = 'Req Owner exec?'
      MKDRQGRD = 'Req Group read?'
      MKDRQGWR = 'Req Group write?'
      MKDRQGEX = 'Req Group exec?'
      MKDRQWRD = 'Req Other read?'
      MKDRQWWR = 'Req Other write?'
      MKDRQWEX = 'Req Other exec?'
      MKDFILPL = 'File pool'
      MKDFILSP = 'File space'
      MKDINODE = 'Inode'
      MKDSCID  = 'File SCID'
      ;
        OUTPUT RACF.MKDIR;
      END;
    %END;
%MEND MKDIR;
./        ADD   LIST=ALL,NAME=MKNOD
%MACRO MKNOD(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
      %PUT Including variables from MKNOD extension;
      RACF.MKNOD (KEEP=%SMFHDR
                       %SMF8ØHDR(REQ=DEFINE)
                       MKNCLASS
                       MKNUSERN
                       MKNUTKNE
                       MKNUPRE
                       MKNUVFYX
                       MKNUNJEU
                       MKNUUAUD
                       MKNUSPEC
                       MKNUDFLT
                       MKNUUNDF
                       MKNUERR
                       MKNUTRST
                       MKNUSEST
                       MKNUSURO
                       MKNURMT
                       MKNUPRVL
                       MKNUSECL
                       MKNUEXND
                       MKNUSUSR
                       MKNUSNOD
                       MKNUSGRP
                       MKNUSPOE
                       MKNUSPCL
                       MKNUTUSR
                       MKNUTGRP
```

```
MKNUTDFT
MKNUTSEC
MKNAPPC
MKNAUDIT
MKNORUID
MKNOEUID
MKNOSUID
MKNORGID
MKNOEGID
MKNOSGID
MKNPATHN
MKNFILID
MKNFOUID
MKNFOGID
MKNOLSGI
MKNOLSUI
MKNOLSVT
MKNOLORD
MKNOLOWR
MKNOLOEX
MKNOLGRD
MKNOLGWR
MKNOLGEX
MKNOLWRD
MKNOLWWR
MKNOLWEX
MKNNWSGI
MKNNWSUI
MKNNWSVT
MKNNWORD
MKNNWOWR
MKNNWOEX
MKNNWGRD
MKNNWGWR
MKNNWGEX
MKNNWWRD
MKNNWWWR
MKNNWWEX
MKNNWURE
MKNNWUWR
MKNNWUEX
MKNNWARE
MKNNWAWR
MKNNWAEX
MKNRQSGI
MKNRQSUI
MKNRQSVT
MKNRQORD
MKNRQOWR
MKNRQOEX
```

```
                    MKNRQGRD
                    MKNRQGWR
                    MKNRQGEX
                    MKNRQWRD
                    MKNRQWWR
                    MKNRQWEX
                    MKNFILPL
                    MKNFILSP
                    MKNINODE
                    MKNSCID
                )
  %END;
%IF &REQ = EXTRACT %THEN
  %DO;
    %PUT Including datadefinition for MKNOD extension;
    WHEN('MKNOD') DO;
      INPUT %SMF80HDR(REQ=EXTRACT)
            MKNCLASS $       282-289
            MKNUSERN $       291-310
            MKNUTKNE $       312-315
            MKNUPRE  $       317-320
            MKNUVFYX $       322-325
            MKNUNJEU $       327-330
            MKNUUAUD $       332-335
            MKNUSPEC $       337-340
            MKNUDFLT $       342-345
            MKNUUNDF $       347-350
            MKNUERR  $       352-355
            MKNUTRST $       357-360
            MKNUSEST $       362-369
            MKNUSURO $       371-374
            MKNURMT  $       376-379
            MKNUPRVL $       381-384
            MKNUSECL $       386-393
            MKNUEXND $       395-402
            MKNUSUSR $       404-411
            MKNUSNOD $       413-420
            MKNUSGRP $       422-429
            MKNUSPOE $       431-438
            MKNUSPCL $       440-447
            MKNUTUSR $       449-456
            MKNUTGRP $       458-465
            MKNUTDFT $       467-470
            MKNUTSEC $       472-475
            MKNAPPC  $       477-492
            MKNAUDIT $       494-504
            MKNORUID         506-515
            MKNOEUID         517-526
            MKNOSUID         528-537
            MKNORGID         539-548
```

```
MKNOEGID          55Ø-559
MKNOSGID          561-57Ø
MKNPATHN $        572-771
MKNFILID $       1596-1627
MKNFOUID         1629-1638
MKNFOGID         164Ø-1649
MKNOLSGI $       1651-1654
MKNOLSUI $       1656-1659
MKNOLSVT $       1661-1664
MKNOLORD $       1666-1669
MKNOLOWR $       1671-1674
MKNOLOEX $       1676-1679
MKNOLGRD $       1681-1684
MKNOLGWR $       1686-1689
MKNOLGEX $       1691-1694
MKNOLWRD $       1696-1699
MKNOLWWR $       17Ø1-17Ø4
MKNOLWEX $       17Ø6-17Ø9
MKNNWSGI $       1711-1714
MKNNWSUI $       1716-1719
MKNNWSVT $       1721-1724
MKNNWORD $       1726-1729
MKNNWOWR $       1731-1734
MKNNWOEX $       1736-1739
MKNNWGRD $       1741-1744
MKNNWGWR $       1746-1749
MKNNWGEX $       1751-1754
MKNNWWRD $       1756-1759
MKNNWWWR $       1761-1764
MKNNWWEX $       1766-1769
MKNNWURE $       1771-1778
MKNNWUWR $       178Ø-1787
MKNNWUEX $       1789-1796
MKNNWARE $       1798-18Ø5
MKNNWAWR $       18Ø7-1814
MKNNWAEX $       1816-1823
MKNRQSGI $       1825-1828
MKNRQSUI $       183Ø-1833
MKNRQSVT $       1835-1838
MKNRQORD $       184Ø-1843
MKNRQOWR $       1845-1848
MKNRQOEX $       185Ø-1853
MKNRQGRD $       1855-1858
MKNRQGWR $       186Ø-1863
MKNRQGEX $       1865-1868
MKNRQWRD $       187Ø-1873
MKNRQWWR $       1875-1878
MKNRQWEX $       188Ø-1883
MKNFILPL $       1885-1892
MKNFILSP $       1894-19Ø1
```

```
               MKNINODE        19Ø3-1912
               MKNSCID         1914-1923
                ;
LABEL MKNCLASS = 'Class name'
      MKNUSERN = 'User name'
      MKNUTKNE = 'Utoken encr.?'
      MKNUPRE  = 'Pre-1.9?'
      MKNUVFYX = 'VERIFYX propagation?'
      MKNUNJEU = 'Undefined NJE user?'
      MKNUUAUD = 'UAUDIT?'
      MKNUSPEC = 'RACF special?'
      MKNUDFLT = 'Default token?'
      MKNUUNDF = 'Undefined user?'
      MKNUERR  = 'Token in error?'
      MKNUTRST = 'User trusted?'
      MKNUSEST = 'Session type'
      MKNUSURO = 'Surrogate user?'
      MKNURMT  = 'Remote job?'
      MKNUPRVL = 'Priviliged user?'
      MKNUSECL = 'User SECLABEL'
      MKNUEXND = 'Execution node'
      MKNUSUSR = 'Submitting user'
      MKNUSNOD = 'Submitting node'
      MKNUSGRP = 'Submitting group'
      MKNUSPOE = 'Port of entry'
      MKNUSPCL = 'Class of POE'
      MKNUTUSR = 'Userid'
      MKNUTGRP = 'Groupid'
      MKNUTDFT = 'Default group?'
      MKNUTSEC = 'Default SECLABEL?'
      MKNAPPC  = 'APPC key link'
      MKNAUDIT = 'Audit code'
      MKNORUID = 'Old real UID'
      MKNOEUID = 'Old effective UID'
      MKNOSUID = 'Old saved UID'
      MKNORGID = 'Old real GID'
      MKNOEGID = 'Old effective GID'
      MKNOSGID = 'Old saved GID'
      MKNPATHN = 'Path name'
      MKNFILID = 'File id'
      MKNFOUID = 'Owner UID'
      MKNFOGID = 'Owner GID'
      MKNOLSGI = 'Old S_ISGID requested?'
      MKNOLSUI = 'Old S_ISUID requested?'
      MKNOLSVT = 'Old S_ISVTX requested?'
      MKNOLORD = 'Old Owner read?'
      MKNOLOWR = 'Old Owner write?'
      MKNOLOEX = 'Old Owner exec?'
      MKNOLGRD = 'Old Group read?'
      MKNOLGWR = 'Old Group write?'
```

```
      MKNOLGEX = 'Old Group exec?'
      MKNOLWRD = 'Old Other read?'
      MKNOLWWR = 'Old Other write?'
      MKNOLWEX = 'Old Other exec?'
      MKNNWSGI = 'New S_ISGID requested?'
      MKNNWSUI = 'New S_ISUID requested?'
      MKNNWSVT = 'New S_ISVTX requested?'
      MKNNWORD = 'New Owner read?'
      MKNNWOWR = 'New Owner write?'
      MKNNWOEX = 'New Owner exec?'
      MKNNWGRD = 'New Group read?'
      MKNNWGWR = 'New Group write?'
      MKNNWGEX = 'New Group exec?'
      MKNNWWRD = 'New Other read?'
      MKNNWWWR = 'New Other write?'
      MKNNWWEX = 'New Other exec?'
      MKNNWURE = 'New user aud read'
      MKNNWUWR = 'New user aud write'
      MKNNWUEX = 'New user aud exec'
      MKNNWARE = 'New auditor aud read'
      MKNNWAWR = 'New auditor aud write'
      MKNNWAEX = 'New auditor aud exec'
      MKNRQSGI = 'Req S_ISGID?'
      MKNRQSUI = 'Req S_ISUID?'
      MKNRQSVT = 'Req S_ISVTX?'
      MKNRQORD = 'Req Owner read?'
      MKNRQOWR = 'Req Owner write?'
      MKNRQOEX = 'Req Owner exec?'
      MKNRQGRD = 'Req Group read?'
      MKNRQGWR = 'Req Group write?'
      MKNRQGEX = 'Req Group exec?'
      MKNRQWRD = 'Req Other read?'
      MKNRQWWR = 'Req Other write?'
      MKNRQWEX = 'Req Other exec?'
      MKNFILPL = 'File pool'
      MKNFILSP = 'File space'
      MKNINODE = 'Inode'
      MKNSCID  = 'File SCID'
      ;
        OUTPUT RACF.MKNOD;
      END;
    %END;
%MEND MKNOD;
./        ADD   LIST=ALL,NAME=MNTFSYS
%MACRO MNTFSYS(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
      %PUT Including variables from MNTFSYS extension;
      RACF.MNTFSYS (KEEP=%SMFHDR
```

```
                        %SMF8ØHDR(REQ=DEFINE)
                        MFSCLASS
                        MFSUSERN
                        MFSUTKNE
                        MFSUPRE
                        MFSUVFYX
                        MFSUNJEU
                        MFSUUAUD
                        MFSUSPEC
                        MFSUDFLT
                        MFSUUNDF
                        MFSUERR
                        MFSUTRST
                        MFSUSEST
                        MFSUSURO
                        MFSURMT
                        MFSUPRVL
                        MFSUSECL
                        MFSUEXND
                        MFSUSUSR
                        MFSUSNOD
                        MFSUSGRP
                        MFSUSPOE
                        MFSUSPCL
                        MFSUTUSR
                        MFSUTGRP
                        MFSUTDFT
                        MFSUTSEC
                        MFSAPPC
                        MFSAUDIT
                        MFSORUID
                        MFSOEUID
                        MFSOSUID
                        MFSORGID
                        MFSOEGID
                        MFSOSGID
                        MFSPATHN
                        MFSFILID
                        MFSFOUID
                        MFSFOGID
                        MFSHFSDS
                        MFSDCELK
                        MFSAUTYP
                )
  %END;
%IF &REQ = EXTRACT %THEN
  %DO;
    %PUT Including datadefinition for MNTFSYS extension;
    WHEN('MNTFSYS') DO;
      INPUT %SMF8ØHDR(REQ=EXTRACT)
```

```
                MFSCLASS $       282-289
                MFSUSERN $       291-310
                MFSUTKNE $       312-315
                MFSUPRE  $       317-320
                MFSUVFYX $       322-325
                MFSUNJEU $       327-330
                MFSUUAUD $       332-335
                MFSUSPEC $       337-340
                MFSUDFLT $       342-345
                MFSUUNDF $       347-350
                MFSUERR  $       352-355
                MFSUTRST $       357-360
                MFSUSEST $       362-369
                MFSUSURO $       371-374
                MFSURMT  $       376-379
                MFSUPRVL $       381-384
                MFSUSECL $       386-393
                MFSUEXND $       395-402
                MFSUSUSR $       404-411
                MFSUSNOD $       413-420
                MFSUSGRP $       422-429
                MFSUSPOE $       431-438
                MFSUSPCL $       440-447
                MFSUTUSR $       449-456
                MFSUTGRP $       458-465
                MFSUTDFT $       467-470
                MFSUTSEC $       472-475
                MFSAPPC  $       477-492
                MFSAUDIT $       494-504
                MFSORUID         506-515
                MFSOEUID         517-526
                MFSOSUID         528-537
                MFSORGID         539-548
                MFSOEGID         550-559
                MFSOSGID         561-570
                MFSPATHN $       572-771
                MFSFILID $      1596-1627
                MFSFOUID        1629-1638
                MFSFOGID        1640-1649
                MFSHFSDS $      1651-1694
                MFSDCELK $      1696-1711
                MFSAUTYP $      1713-1725
                 ;
LABEL MFSCLASS = 'Class name'
      MFSUSERN = 'User name'
      MFSUTKNE = 'Utoken encr.?'
      MFSUPRE  = 'Pre-1.9?'
      MFSUVFYX = 'VERIFYX propagation?'
      MFSUNJEU = 'Undefined NJE user?'
      MFSUUAUD = 'UAUDIT?'
```

```
      MFSUSPEC = 'RACF special?'
      MFSUDFLT = 'Default token?'
      MFSUUNDF = 'Undefined user?'
      MFSUERR  = 'Token in error?'
      MFSUTRST = 'User trusted?'
      MFSUSEST = 'Session type'
      MFSUSURO = 'Surrogate user?'
      MFSURMT  = 'Remote job?'
      MFSUPRVL = 'Priviliged user?'
      MFSUSECL = 'User SECLABEL'
      MFSUEXND = 'Execution node'
      MFSUSUSR = 'Submitting user'
      MFSUSNOD = 'Submitting node'
      MFSUSGRP = 'Submitting group'
      MFSUSPOE = 'Port of entry'
      MFSUSPCL = 'Class of POE'
      MFSUTUSR = 'Userid'
      MFSUTGRP = 'Groupid'
      MFSUTDFT = 'Default group?'
      MFSUTSEC = 'Default SECLABEL?'
      MFSAPPC  = 'APPC key link'
      MFSAUDIT = 'Audit code'
      MFSORUID = 'Old real UID'
      MFSOEUID = 'Old effective UID'
      MFSOSUID = 'Old saved UID'
      MFSORGID = 'Old real GID'
      MFSOEGID = 'Old effective GID'
      MFSOSGID = 'Old saved GID'
      MFSPATHN = 'Path name'
      MFSFILID = 'File id'
      MFSFOUID = 'Owner UID'
      MFSFOGID = 'Owner GID'
      MFSHFSDS = 'HFS datasetname'
      MFSDCELK = 'DCE link'
      MFSAUTYP = 'Request type'
      ;
        OUTPUT RACF.MNTFSYS;
      END;
    %END;
%MEND MNTFSYS;
./        ADD   LIST=ALL,NAME=OPENFILE
%MACRO OPENFILE(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
      %PUT Including variables from OPENFILE extension;
      RACF.OPENFILE (KEEP=%SMFHDR
                          %SMF8ØHDR(REQ=DEFINE)
                          OPNCLASS
                          OPNUSERN
```

```
OPNUTKNE
OPNUPRE
OPNUVFYX
OPNUNJEU
OPNUUAUD
OPNUSPEC
OPNUDFLT
OPNUUNDF
OPNUERR
OPNUTRST
OPNUSEST
OPNUSURO
OPNURMT
OPNUPRVL
OPNUSECL
OPNUEXND
OPNUSUSR
OPNUSNOD
OPNUSGRP
OPNUSPOE
OPNUSPCL
OPNUTUSR
OPNUTGRP
OPNUTDFT
OPNUTSEC
OPNAPPC
OPNAUDIT
OPNORUID
OPNOEUID
OPNOSUID
OPNORGID
OPNOEGID
OPNOSGID
OPNPATHN
OPNFILID
OPNFOUID
OPNFOGID
OPNOLSGI
OPNOLSUI
OPNOLSVT
OPNOLORD
OPNOLOWR
OPNOLOEX
OPNOLGRD
OPNOLGWR
OPNOLGEX
OPNOLWRD
OPNOLWWR
OPNOLWEX
OPNNWSGI
```

```
                        OPNNWSUI
                        OPNNWSVT
                        OPNNWORD
                        OPNNWOWR
                        OPNNWOEX
                        OPNNWGRD
                        OPNNWGWR
                        OPNNWGEX
                        OPNNWWRD
                        OPNNWWWR
                        OPNNWWEX
                        OPNNWURE
                        OPNNWUWR
                        OPNNWUEX
                        OPNNWARE
                        OPNNWAWR
                        OPNNWAEX
                        OPNRQSGI
                        OPNRQSUI
                        OPNRQSVT
                        OPNRQORD
                        OPNRQOWR
                        OPNRQOEX
                        OPNRQGRD
                        OPNRQGWR
                        OPNRQGEX
                        OPNRQWRD
                        OPNRQWWR
                        OPNRQWEX
                        OPNFILPL
                        OPNFILSP
                        OPNINODE
                        OPNSCID
                    )
  %END;
%IF &REQ = EXTRACT %THEN
  %DO;
    %PUT Including datadefinition for OPENFILE extension;
    WHEN('OPENFILE') DO;
      INPUT %SMF8ØHDR(REQ=EXTRACT)
            OPNCLASS $       282-289
            OPNUSERN $       291-31Ø
            OPNUTKNE $       312-315
            OPNUPRE  $       317-32Ø
            OPNUVFYX $       322-325
            OPNUNJEU $       327-33Ø
            OPNUUAUD $       332-335
            OPNUSPEC $       337-34Ø
            OPNUDFLT $       342-345
            OPNUUNDF $       347-35Ø
```

```
OPNUERR  $       352-355
OPNUTRST $       357-360
OPNUSEST $       362-369
OPNUSURO $       371-374
OPNURMT  $       376-379
OPNUPRVL $       381-384
OPNUSECL $       386-393
OPNUEXND $       395-402
OPNUSUSR $       404-411
OPNUSNOD $       413-420
OPNUSGRP $       422-429
OPNUSPOE $       431-438
OPNUSPCL $       440-447
OPNUTUSR $       449-456
OPNUTGRP $       458-465
OPNUTDFT $       467-470
OPNUTSEC $       472-475
OPNAPPC  $       477-492
OPNAUDIT $       494-504
OPNORUID         506-515
OPNOEUID         517-526
OPNOSUID         528-537
OPNORGID         539-548
OPNOEGID         550-559
OPNOSGID         561-570
OPNPATHN $       572-771
OPNFILID $      1596-1627
OPNFOUID        1629-1638
OPNFOGID        1640-1649
OPNOLSGI $      1651-1654
OPNOLSUI $      1656-1659
OPNOLSVT $      1661-1664
OPNOLORD $      1666-1669
OPNOLOWR $      1671-1674
OPNOLOEX $      1676-1679
OPNOLGRD $      1681-1684
OPNOLGWR $      1686-1689
OPNOLGEX $      1691-1694
OPNOLWRD $      1696-1699
OPNOLWWR $      1701-1704
OPNOLWEX $      1706-1709
OPNNWSGI $      1711-1714
OPNNWSUI $      1716-1719
OPNNWSVT $      1721-1724
OPNNWORD $      1726-1729
OPNNWOWR $      1731-1734
OPNNWOEX $      1736-1739
OPNNWGRD $      1741-1744
OPNNWGWR $      1746-1749
OPNNWGEX $      1751-1754
```

```
OPNNWWRD $      1756-1759
OPNNWWWR $      1761-1764
OPNNWWEX $      1766-1769
OPNNWURE $      1771-1778
OPNNWUWR $      178Ø-1787
OPNNWUEX $      1789-1796
OPNNWARE $      1798-18Ø5
OPNNWAWR $      18Ø7-1814
OPNNWAEX $      1816-1823
OPNRQSGI $      1825-1828
OPNRQSUI $      183Ø-1833
OPNRQSVT $      1835-1838
OPNRQORD $      184Ø-1843
OPNRQOWR $      1845-1848
OPNRQOEX $      185Ø-1853
OPNRQGRD $      1855-1858
OPNRQGWR $      186Ø-1863
OPNRQGEX $      1865-1868
OPNRQWRD $      187Ø-1873
OPNRQWWR $      1875-1878
OPNRQWEX $      188Ø-1883
OPNFILPL $      1885-1892
OPNFILSP $      1894-19Ø1
OPNINODE        19Ø3-1912
OPNSCID         1914-1923
```

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

System ‘hacks’

Some time ago I was approached by our internal audit team. They were looking into ways in which an MVS system could be compromised and assumed that I, as the senior systems programmer, was best placed to help them – either that or they decided that I was the biggest risk!

Following our discussions, it seemed that a number of situations had been documented which appeared to be outside the normal auditor checklist. As a result, I thought it would be worthwhile to pass on this information so that it might be of use to others.

Note that this is not meant to be a definitive list of potential exposures, it is simply meant as a starting point for extra things to review when checking your own security set-up. (Our system is MVS/ESA Version 4+ and TSO/E.) I hope you find them useful:

- LPAR access – if your site has multiple LPARs (for example a test machine and a production machine) then you need to be sure that access to volumes not normally attached to an LPAR is controlled. There are several reasons for this:
  - From a technical viewpoint, if the device is not defined as shared, it is possible to accidentally corrupt the VTOC.
  - From an access viewpoint, it is important to ensure that RACF controls the data correctly across all partitions. Note that although this may be covered by a shared database, or an RRSF ring, there may still be loopholes for (say) a test machine, ie the access to RACF facilities may be greater on the test machine thus allowing people to grant access to data they may not otherwise have.
  - Do not assume that OPERCMDS will prevent devices being brought on-line. Products such as OMEGAMON or SYSVIEW may have access and allow a user access as a result.
- One of the significant areas of concern for security and audit people is that of access to APF libraries. While this is undoubtedly

valid – because of the risk that access to APF libraries poses – the simple act of controlling access may be insufficient and you also need to investigate the following:

  - Special SVCs that enable APF authority dynamically. If you have any they are probably there for a good reason – but it is essential to be sure that they are only being used for the 'right' reasons.
  - Products (or home-grown code) that allow users access to facilities that they would not normally have. For example, products such as OMEGAMON, SYSVIEW, CMF, PDSMAN, etc, all have functions that can manipulate the operating system (eg dynamic system library modification, storage manipulation, etc).
  - The main reason for being worried about APF libraries is that it allows code to exploit the MODESET SVC. This SVC permits the user to change storage keys, and to switch in to supervisor state. In other words, it can permit a coder to write code that can attack or bypass virtually anything. This of course assumes that the MODESET SVC actually needs to be issued in APF authorized state! Given enough access in some products, it is possible to switch off the APF bit in the SVC table. As a result, non-authorized programs could issue this SVC. It is essential that the SVC table is monitored for such corruption.

- Be aware of products that allow storage altering (eg CMF, OMEGAMON, SYSVIEW, etc). Apart from the possibility of such products being used to modify code dynamically, or indeed to crash the system, it is easy to use such a product to modify a user's ACEE and grant special access dynamically. The availability of such facilities at a site must be identified and controlled.
- Check for the existence of the TSO LOGON exit IKJEFLD (or IKJEFLD1) at your site. Through these, it is a simple matter to get TSO to store the RACF password used to sign on in clear text in the TSB control block. Note that this is documented in the IBM

manuals as an acceptable process if the password needs to be included by the submit exit, IKJEFF10. It may therefore be an acceptable situation, but it can also be a risky one when combined with products that permit storage display, or if the submit exit was used inappropriately.

- Check the set-up of STGADMIN at your site. There have been several cases where all the facilities were set to a default of read by mistake. As a result, users with access to this can bypass normal RACF control (ie it would be possible to delete a dataset to which you do not normally have access, by running a DFDSS job with the ADMIN parameter).
- Superzap (IMASPZAP) is often documented as being a 'dangerous' program and is protected accordingly. However these days it is a minor danger compared to the potential of certain products that allow the 'zap' process to be completed more easily (eg OMEGAMON, SYSVIEW, PDSMAN, etc). Therefore, it is important to check for zap-abilities in products before allowing access to users.
- REXX – most flavours of REXX have a STORAGE function – often with an alteration capability. It is important to check whether this is allowed. Check the standard REXX, and also check any other system that may have a REXX interface, such as automation products (AF operator, NetView, etc).
- Automation – check who has access to your scheduling or console automation facilities. Incorrect modification of these areas could result in a dramatic impact on your site's ability to process its workload correctly.

Expiring users' passwords

When RACF administrators reset a user's password, the user is forced to change it at the next log-on, having been informed of the value that the administrator has set. However, sometimes it would be useful simply to expire the password.

This can be useful after a security threat, or when a team member leaves, to ensure that all users in the affected area change their passwords, but avoiding the need to manually reset them and re-issue the new values (which itself is often a security risk).

The TSO command processor reproduced here allows a RACF administrator to expire the password (that is change the PASSDATE to zeros) for any user within his/her scope of control. The user will then be forced to change password at the next log-on. Another reason for this is to immediately enforce new password content rules.

This module needs to be APF authorized and in the system linklist. It can be used in the foreground or in TSO batch.

ZEXPUSER

```
************************************************************
* MODULE : ZEXPUSER
*
* CUSTOM WRITTEN RACF COMMAND TO SET PASSDATE TO ZEROS SO
* USER IS FORCED TO CHANGED PASSWORD AT NEXT LOG-ON. THE
* CURRENT PASSWORD IS NOT CHANGED DURING THIS PROCESS.
*
* SYNTAX : ZEXPUSER USERID
*
* ALLOWS UPDATE IF ISSUER IS GLOBAL SPECIAL OR HAS GROUP
* SPECIAL ATTRIBUTE FOR THE TARGET USERS OWNING GROUP OR
* FOR THE SUPERIOR GROUP TO THAT AND SO ON UP TO SYS1.
*
* MUST BE APF AUTHORIZED (IN IKJTSO00).
* RETURN CODES
*
* 0  - SUCCESSFUL COMPLETION
* 4  - NO ACTION DONE
* 8  - NOT AUTHORIZED FOR ISSUER ON TARGET USER
* 12 - PGM ERROR
```

```
* 16 - PGM NOT APF AUTH
************************************************************
ZEXPUSER CSECT
ZEXPUSER AMODE 31
ZEXPUSER RMODE 24
         SPLEVEL SET=2
         STM   R14,R12,12(R13)   SAVE REGISTERS
         LR    R12,R15           ADDRESSABILITY
         LA    R11,2Ø48(R12)     LOAD R11
         LA    R11,2Ø48(R11)     LOAD R11
         LR    R4,R1             SAVE CPPL ADDRESS
         USING CPPL,R4           ADDRESSABILITY FOR CPPL
         USING ZEXPUSER,R12,R11
         GETMAIN R,LV=WORKLEN    GETMAIN DYNAMIC AREA
         LR    R1Ø,R1            R1Ø -> DYNAMIC AREA
         USING WORKAREA,R1Ø      ADDRESS DYNAMIC AREA
         ST    R13,SAVEAREA+4    SAVE CALLERS SAVEAREA ADDRESS
         ST    R1Ø,8(R13)        SAVE SAVEAREA ADDRESS
         LR    R13,R1Ø           SAVE AREA PTR
*
* INITIALIZE PUTLINE PARAMETER LIST
*
         LA    R1,PUTIWORK              ADDR IOPL
         USING IOPL,R1
         MVC   IOPLUPT,CPPLUPT          STORE ADDR USER PROFILE TAB
         MVC   IOPLECT,CPPLECT          STORE ADDR ENVIR CONTROL TAB
         LA    R2,LOCECB
         ST    R2,IOPLECB               STORE ADDR ECB
         DROP  R1
*
* CHECK THAT THIS CMD IS RUNNING APF AUTHORISED
*
         TESTAUTH FCTN=1                TEST APF
         LTR   R15,R15                  RC
         BNZ   ENDNOTAF                 NOT APF
*
* ESTABLISH PARSE PARAMETER LIST (PPL)
*
         LA    R6,LOCPPL                OUR LOCAL PPL TO BE BUILT
         USING PPL,R6
         L     R1,CPPLUPT               USER PROFILE TABLE
         ST    R1,PPLUPT
         L     R1,CPPLECT               ENVIRONMENT CONTROL TABLE
         ST    R1,PPLECT
         XC    LOCECB,LOCECB            ZERO LOCAL ECB
         LA    R1,LOCECB                ADDRESS OF LOCAL ECB
         ST    R1,PPLECB
         L     R1,=V(PCLPDL)            PARAM CONTROL/DESCRIPTOR LIST
         ST    R1,PPLPCL
         LA    R1,LOCANS                RETURNED PDL POINTER
```

```
         ST    R1,PPLANS
         L     R1,CPPLCBUF              CPPL COMMAND BUFFER
         ST    R1,PPLCBUF
         SR    R1,R1                    NO WORK AREA
         ST    R1,PPLUWA
         DROP  R4,R6
*
* PARSE THE TSO COMMAND BUFFER
*
         CALLTSSR EP=IKJPARS,MF=(E,(R6))
         B     *+4(R15)
         B     PARSEOK                   Ø - PARSED OK
         B     ENDNOACT                  4 - PARMS INCOMPLETE
         B     ENDNOACT                  8 - ATTENTION INTERRUPT
         B     ENDPARS                  12 - INVALID CONTROL BLOCK
         B     ENDNOST                  16 - INSUFFICIENT STORAGE
         B     ENDNOACT                 2Ø - VALIDITY CHECK RTN FAILED
         B     ENDPARS                  24 - INVALID PARAMETERS
         B     ENDNOACT                 28 - TERMINAL DISCONNECTED
*
* PARSE WAS SUCCESSFUL, EXTRACT USERID
*
PARSEOK  EQU   *
         L     R4,LOCANS                PROCESS DESCRIPTOR LIST
         USING IKJPARMD,R4              ADDRESSABILITY FOR PDL
         ICM   R1,B'1111',IKJUID        SOURCE FIELD
         BZ    ENDNOACT                 NOT PRESENT, EXIT
         XR    R2,R2                    CLEAR
         ICM   R2,B'ØØØ1',IKJUID+5      LENGTH OF SOURCE FIELD
         BZ    ENDNOACT                 ZERO LENGTH, EXIT
         MVC   USERID,BLANKS            CLEAR
         BCTR  R2,Ø                     DOWN ONE
         EX    R2,MUSR                  MOVE USER-ID
         B     GAUTH                    JUMP
MUSR     MVC   USERID(Ø),Ø(R1)          COPY USER-ID
*
* CHECK USER'S AUTHORITY THROUGH GLOBAL SPECIAL
*
GAUTH    EQU   *
         USING PSA,RØ                   ADDRESSABILITY FOR PSA
         L     R1,PSATOLD               ADDR OUR TCB
         USING TCB,R1
         ICM   R1,15,TCBSENV            ADDR THE ACEE
         BNZ   GOTACEE                  BRANCH IF PRESENT
         L     R1,PSAAOLD               ADDR OUR ASCB
         USING ASCB,R1
         L     R1,ASCBASXB              ADDR THE ASXB
         USING ASXB,R1
         L     R1,ASXBSENV              ADDR THE ACEE
*
```

```
* ACEE FOUND, TEST GLOBAL SPECIAL ATTRIBUTE
*
GOTACEE  EQU   *
         LTR   R1,R1                    TEST ACEE
         BZ    ENDNOTAU                 NOT THERE
         USING ACEE,R1
         MVC   ISSUER,ACEEUSRI          SAVE ISSUING USER-ID
         ST    R1,ADDRACEE              SAVE ACEE ADDR
         TM    ACEEFLG1,ACEESPEC        TEST FOR SPECIAL USER
         BO    AUTHOK                   YES, THEN AUTHORIZED
         DROP  R1
*
* NON-GLOBAL SPECIAL OBTAIN OWNING GROUP OF USER TO BE MODIFIED
*
         MVC   ITEM,USERID       CURRENT REQUEST
         LA    R8,USERID         ADDR USERID FIELD
         LA    R6,RFLD1A         ADDR FIELDS TO BE OBTAINED
         LA    R9,RACWORK        RACROUTE WORK AREA
         MVC   RX4(RX4L),RX3     INITIALIZE REENTRANT AREA
         RACROUTE REQUEST=EXTRACT,WORKA=(9),RELEASE=1.8.1,            X
               TYPE=EXTRACT,FIELDS=(6),ENTITY=(8),MF=(E,RX4)
         L     R4,RX4            LOAD RACXTRT RETURN CODE
         L     R5,RX4+4          LOAD RACXTRT REASON CODE
         LTR   R15,R15           TEST RACROUTE RETURN CODE
         BNZ   ENDBRAC           RACXTRT FAILED
         LR    R7,R1             RETURN AREA ADDR
*
* PROCESS THE RETURNED INFOMATION, SEG AREA IS LEN(4), DATA(8)
*
         XR    R4,R4             CLEAR
         IC    R4,Ø(R7)          SUBPOOL OF GETMAINED AREA
         XR    R5,R5             CLEAR
         ICM   R5,B'Ø111',1(R7)  LENGTH OF GETMAINED AREA
         LH    R6,4(R7)          OFFSET TO SEGMENT AREA
         AR    R6,R7             ADD BASE ADDR FOR SEGMENT AREA
         MVC   GROUPN,4(R6)      COPY USER PROFILE OWNER (8)
         FREEMAIN R,LV=(5),A=(7),SP=(4) FREE RACXTRT AREA
         LA    R2,2ØØ            PREVENT INFINITE LOOP
*
         CLC   ISSUER,GROUPN     IS ISSUER THE OWNER OF ID
         BE    AUTHOK            YES HE IS
*
* CHECK ISSUER HAS GROUP SPECIAL FOR THE USERS OWNING GROUP
*
CHECKG   EQU   *
         L     R4,ADDRACEE             LOAD ACEE ADDR
         USING ACEE,R4
         L     R5,ACEECGRP             ADDR OF CONNECT TABLE, 1
         LTR   R5,R5                   TEST
         BNZ   OKCG                    OK, THERE
```

```
         L     R5,ACEEFCGP              ADDR OF CONNECT TABLE, 2
         LTR   R5,R5                    TEST
         BZ    ENDNOTAU                 NOT THERE
OKCG     EQU   *
         USING CGRP,R5                  ADDR CONNECT TABLE
         LA    R6,CGRPENT               ADDR FIRST ENTRY
         LH    R7,CGRPNUM               NUMBER OF ENTRIES
         USING CGRPENTD,R6
LOOPG    EQU   *
         CLC   GROUPN,CGRPNAME          IS THIS THE GROUP
         BE    MATCHG                   YES
         LA    R6,L'CGRPENT(R6)         INCREMENT ENTRY
         BCT   R7,LOOPG                 CHECK NEXT
         B     NEXTG                    NOT AUTHORIZED
MATCHG   EQU   *
         TM    CGRPAUTH,CGRPSPEC        GROUP SPECIAL FLAG
         BO    AUTHOK                   ON, AUTH
*
* GET SUPERIOR GROUP TO LAST GROUP AND CHECK AGAIN UNTIL SYS1
*
NEXTG    EQU   *
         CLC   GROUPN,=CL8'SYS1' IS IT THE FINAL GROUP
         BE    ENDNOTAU           NO CHANCE
         MVC   ITEM,GROUPN        CURRENT REQUEST
         LA    R8,GROUPN          ADDR GROUP FIELD
         LA    R6,RFLD2A          ADDR FIELDS TO BE OBTAINED
         LA    R9,RACWORK         RACROUTE WORK AREA
         MVC   RX6(RX6L),RX5      INITIALIZE REENTRANT AREA
         RACROUTE REQUEST=EXTRACT,WORKA=(9),RELEASE=1.8.1,            X
               TYPE=EXTRACT,FIELDS=(6),ENTITY=(8),MF=(E,RX6)
         L     R4,RX6             LOAD RACXTRT RETURN CODE
         L     R5,RX6+4           LOAD RACXTRT REASON CODE
         LTR   R15,R15            TEST RACROUTE RETURN CODE
         BNZ   ENDBRAC            RACXTRT FAILED
         LR    R7,R1              RETURN AREA ADDR
*
* PROCESS THE RETURNED INFOMATION, SEG AREA IS LEN(4), DATA(8)
*
         XR    R4,R4              CLEAR
         IC    R4,Ø(R7)           SUBPOOL OF GETMAINED AREA
         XR    R5,R5              CLEAR
         ICM   R5,B'Ø111',1(R7)   LENGTH OF GETMAINED AREA
         LH    R6,4(R7)           OFFSET TO SEGMENT AREA
         AR    R6,R7              ADD BASE ADDR FOR SEGMENT AREA
         MVC   GROUPN,4(R6)       COPY GROUP SUPERIOR GROUP
         FREEMAIN R,LV=(5),A=(7),SP=(4) FREE RACXTRT AREA
         BCT   R2,CHECKG          CHECK FOR GROUP SPECIAL AGAIN
         B     ENDNOTAU           NO CHANCE (AFTER 2ØØ LOOPS)
         DROP  R4,R5,R6
*
```

```
* PERFORM UPDATE ON LOCAL SYSTEM WITH ICHEINTY.
*
AUTHOK   EQU   *
         LA    R4,USERID         ADDR USERNAME
         LA    R5,8              MAX LENGTH
         LA    R6,Ø              COUNTER
LOOPU    EQU   *
         CLI   Ø(R4),C' '        END YET?
         BE    ENDU              YES
         LA    R4,1(R4)          UP PTR
         LA    R6,1(R6)          UP COUNTER
         BCT   R5,LOOPU          LOOP IF NOT END
ENDU     EQU   *
         STC   R6,NUSER          SET LENGTH OF USER-ID
         MVC   NUSER+1(8),USERID SET VALUE OF USER-ID
         MVC   NDATE,=XL4'ØØØØØØØF'   SET TO ZEROS (PACK DEC)
*
         LA    R4,NDATE+1        ADDRESS PASSDATE FOR ACT2
         MVC   ACT2B(ACT2BL),ACT2A    TO REENTRANT AREA
         ICHEACTN FLDATA=(3,(4)),MF=(E,ACT2B)
*
         LA    R4,NUSER          ADDRESS USER-ID
         MVC   ICH2(ICH2L),ICH1  COPY TO REENTRANT AREA
         ICHEINTY ALTER,TYPE='USR',ENTRY=(4),ACTIONS=(ACT2B),          X
               OPTIONS=(FLDEF),MF=(E,ICH2)
         LR    R4,R15            LOAD RETURN CODE
         LR    R5,RØ             LOAD REASON CODE
         LTR   R15,R15           TEST RETURN CODE
         BNZ   RACXMSG           ICHEINTY FAILED, SHOW MSG
*
* TERMINATE AFTER SUCESSFUL PROCESSING
*
ENDRCØ   EQU   *
         LA    R3,Ø
*
* QUIT COMMAND
*
EXIT     EQU   *
         L     R13,SAVEAREA+4    RESTORE R13
         FREEMAIN R,LV=WORKLEN,A=(1Ø) FREE DYNAMIC AREA
         LR    R15,R3            RETURN CODE TO R15
         L     R14,12(R13)       RESTORE R14
         LM    RØ,R12,2Ø(R13)    RESTORE RØ TO R12
         BR    R14               RETURN
*
* TSO PUTLINE MESSAGES
*
ENDNOTAF EQU   *
         LA    R2,MSGØOLD              PGM NOT APF AUTHORIZED
         LA    R3,16                   SET RETURN CODE
```

```
         B     ENDMSG
*
ENDNOTAU EQU   *
         LA    R2,MSG1OLD               USER NOT AUTHORIZED
         LA    R3,8                     SET RETURN CODE
         B     ENDMSG
*
ENDNOACT EQU   *
         LA    R2,MSG2OLD               NO ACTION ATTEMPTED
         LA    R3,4                     SET RETURN CODE
         B     ENDMSG
*
ENDPARS  EQU   *
         LA    R2,MSG3OLD               PROGRAM ERROR DURING PARSE
         LA    R3,12                    SET RETURN CODE
         B     ENDMSG
*
ENDNOST  EQU   *
         LA    R2,MSG4OLD               INSUFFICIENT STORAGE FOR PARSE
         LA    R3,12                    SET RETURN CODE
         B     ENDMSG
*
ENDBRAC  EQU   *
         MVC   RMSG,BLANKS             CLEAR MSG
         CVD   R4,WORK1                CONVERT TO DEC
         CVD   R5,WORK2                CONVERT TO DEC
         UNPK  WORK3,WORK1             UNPACK
         MVZ   WORK3+7(1),=X'FØ'       SET ZONE
         MVC   RMSG(9),=CL9'RETURN = ' INTO MSG
         MVC   RMSG+9(8),WORK3         INTO MSG
         UNPK  WORK4,WORK2             UNPACK
         MVZ   WORK4+7(1),=X'FØ'       SET ZONE
         MVC   RMSG+9+9(9),=CL9'REASON = ' INTO MSG
         MVC   RMSG+9+9+9(8),WORK4         INTO MSG
         MVC   WTO2(WTO2L),WTO1        COPY WTO
         MVC   WTO2+4(4Ø),RMSG         INTO WTO
         WTO   'ZEXPUSER - RACXTRT MACRO OPERATION FAILED',ROUTCDE=11
         WTO   MF=(E,WTO2)             ISSUE
         TPUT  ITEM,8                  CURRENT REQUEST
         LA    R2,MSG5OLD              RACXTRT FAILED
         LA    R3,2Ø                   SET RETURN CODE
         C     R4,=F'8'                IS RC 8 (NOT FOUND ITEM)
         BNE   ENDMSG                  NO, EXIT
         TPUT  =CL4Ø'ABOVE USER/GROUP DOES NOT EXIST',4Ø
         B     ENDMSG                  EXIT
*
* ISSUE PUTLINE
*
ENDMSG   EQU   *
         PUTLINE PARM=PUTPWORK,                                    XXXXX
```

```
               OUTPUT=((R2),TERM,SINGLE,INFOR),                    XXXXX
               MF=(E,PUTIWORK)
         B     EXIT
*
* RACF ERROR MESSAGE
*
RACXMSG  EQU   *
         MVC   RMSG,BLANKS             CLEAR MSG
         CVD   R4,WORK1                CONVERT TO DEC
         CVD   R5,WORK2                CONVERT TO DEC
         UNPK  WORK3,WORK1             UNPACK
         MVZ   WORK3+7(1),=X'FØ'       SET ZONE
         MVC   RMSG(9),=CL9'RETURN = ' INTO MSG
         MVC   RMSG+9(8),WORK3         INTO MSG
         UNPK  WORK4,WORK2             UNPACK
         MVZ   WORK4+7(1),=X'FØ'       SET ZONE
         MVC   RMSG+9+9(9),=CL9'REASON = ' INTO MSG
         MVC   RMSG+9+9+9(8),WORK4         INTO MSG
         MVC   WTO2(WTO2L),WTO1        COPY WTO
         MVC   WTO2+4(4Ø),RMSG         INTO WTO
         WTO   'ZEXPUSER - RACF MACRO OPERATION FAILED',ROUTCDE=11
         WTO   MF=(E,WTO2)             ISSUE
         LA    R2,MSG6OLD              RACF FAILED
         LA    R3,24                   SET RETURN CODE
         C     R4,=F'12'               IS RC 12 (NOT FOUND USER)
         BNE   ENDMSG                  NO, EXIT
         TPUT  =CL4Ø'SPECIFIED USERID DOES NOT EXIST',4Ø
         B     ENDMSG                  EXIT
*
         LTORG
*
* CONSTANTS, NON-MODIFIABLE
*
MARKER   DC    C'ZEXPUSER &SYSDATE' DUMP AID
PATCHA   DC    4ØS(*)
ZEROS    DC    256XL1'ØØ'        ZEROS
BLANKS   DC    CL255' '          BLANKS
*
RFLD1A   DC    A(1)              NUMBER OF FIELDS FOLLOWING
RFLD1B   DC    CL8'AUTHOR'       NAME OF FIELD (OWNER)
*
RFLD2A   DC    A(1)              NUMBER OF FIELDS FOLLOWING
RFLD2B   DC    CL8'SUPGROUP'     NAME OF FIELD (SUPERIOR GROUP)
*
RX3      RACROUTE REQUEST=EXTRACT,WORKA=ZEROS,RELEASE=1.8.1,           X
               TYPE=EXTRACT,FIELDS=ZEROS,ENTITY=ZEROS,CLASS='USER',MF=L
*
RX5      RACROUTE REQUEST=EXTRACT,WORKA=ZEROS,RELEASE=1.8.1,           X
               TYPE=EXTRACT,FIELDS=ZEROS,ENTITY=ZEROS,                X
               CLASS='GROUP',MF=L
```

```
*
ICH1     ICHEINTY ALTER,TYPE='USR',ENTRY=ZEROS,ACTIONS=(ZEROS),        X
               MF=L
ACT2A    ICHEACTN FIELD=PASSDATE,FLDATA=(3,ZEROS),MF=L
*
WTOX1    DS    ØF
WTO1     WTO '                                        ',ROUTCDE=11,MF=L
*
* COMMAND BUFFER PARSE CONTROL BLOCKS
*
PCLPDL   IKJPARM
IKJUID   IKJIDENT 'USERID',FIRST=ANY,OTHER=ANY,MAXLNTH=8,          XXXXX
               PROMPT='USERID',                                    XXXXX
               HELP=('RACF USERID')
         IKJENDP
*
* PUTLINE MESSAGES
*
MSGØOLD  DS    ØF                      OUTPUT LINE DESCRIPTOR
         DC    A(1)                    NO OF SEGMENTS
         DC    A(MSGØSEG)              ADDRESS OF SEGMENT
MSGØSEG  DC    Y(MSGØLEN)              SEGMENT LENGTH
         DC    H'Ø'                    RESERVED
         DC    C'ZEXPUSER: COMMAND NOT APF AUTHORIZED'
MSGØLEN  EQU   *-MSGØSEG
*
MSG1OLD  DS    ØF                      OUTPUT LINE DESCRIPTOR
         DC    A(1)                    NO OF SEGMENTS
         DC    A(MSG1SEG)              ADDRESS OF SEGMENT
MSG1SEG  DC    Y(MSG1LEN)              SEGMENT LENGTH
         DC    H'Ø'                    RESERVED
         DC    C'ZEXPUSER: NOT AUTHORIZED TO UPDATE THIS USER'
MSG1LEN  EQU   *-MSG1SEG
*
MSG2OLD  DS    ØF                      OUTPUT LINE DESCRIPTOR
         DC    A(1)                    NO OF SEGMENTS
         DC    A(MSG2SEG)              ADDRESS OF SEGMENT
MSG2SEG  DC    Y(MSG2LEN)              SEGMENT LENGTH
         DC    H'Ø'                    RESERVED
         DC    C'ZEXPUSER: NO ACTION ATTEMPTED, CHECK PARAMETERS'
MSG2LEN  EQU   *-MSG2SEG
*
MSG3OLD  DS    ØF                      OUTPUT LINE DESCRIPTOR
         DC    A(1)                    NO OF SEGMENTS
         DC    A(MSG3SEG)              ADDRESS OF SEGMENT
MSG3SEG  DC    Y(MSG3LEN)              SEGMENT LENGTH
         DC    H'Ø'                    RESERVED
         DC    C'ZEXPUSER: PROGRAM ERROR DURING COMMAND PARSE'
MSG3LEN  EQU   *-MSG3SEG
*
```

```
MSG4OLD  DS    ØF                       OUTPUT LINE DESCRIPTOR
         DC    A(1)                     NO OF SEGMENTS
         DC    A(MSG4SEG)               ADDRESS OF SEGMENT
MSG4SEG  DC    Y(MSG4LEN)               SEGMENT LENGTH
         DC    H'Ø'                     RESERVED
         DC    C'ZEXPUSER: INSUFFICIENT STORAGE FOR PARSE'
MSG4LEN  EQU   *-MSG4SEG
*
MSG5OLD  DS    ØF                       OUTPUT LINE DESCRIPTOR
         DC    A(1)                     NO OF SEGMENTS
         DC    A(MSG5SEG)               ADDRESS OF SEGMENT
MSG5SEG  DC    Y(MSG5LEN)               SEGMENT LENGTH
         DC    H'Ø'                     RESERVED
         DC    C'ZEXPUSER: RACXTRT MACRO HAS FAILED'
MSG5LEN  EQU   *-MSG5SEG
*
MSG6OLD  DS    ØF                       OUTPUT LINE DESCRIPTOR
         DC    A(1)                     NO OF SEGMENTS
         DC    A(MSG6SEG)               ADDRESS OF SEGMENT
MSG6SEG  DC    Y(MSG6LEN)               SEGMENT LENGTH
         DC    H'Ø'                     RESERVED
         DC    C'ZEXPUSER: RACF ICHEINTY MACRO HAS FAILED'
MSG6LEN  EQU   *-MSG6SEG
*
* PROGRAM DATA AREA
*
WORKAREA DSECT
SAVEAREA DS    18F              SAVE AREA
ALIGND   DS    ØD               ALIGN DECS
WORK1    DS    PL8              WORK AREA FOR RETURN/REASON CODES
WORK2    DS    PL8              WORK AREA FOR RETURN/REASON CODES
WORK3    DS    CL8              WORK AREA FOR RETURN/REASON CODES
WORK4    DS    CL8              WORK AREA FOR RETURN/REASON CODES
RMSG     DS    CL4Ø             RACF MSG FOR WTO RETURN/REASON CODES
ITEM     DS    CL8              CURRENT ITEM FOR RACX
*
ADDRACEE DS    F                ACEE ADDR
ISSUER   DS    CL8              ISSUING USER-ID
*
NUSER    DS    AL1,CL8          USER-ID NAME IN LEN, VALUE FORMAT
NDATE    DS    F                NEW PASSWORD DATE
*
USERID   DS    CL8              TARGET USER-ID
GROUPN   DS    CL8              USER/GRP OWNING GROUP
*
RX4      RACROUTE REQUEST=EXTRACT,WORKA=ZEROS,RELEASE=1.8.1,           X
               TYPE=EXTRACT,FIELDS=ZEROS,ENTITY=ZEROS,CLASS='USER',MF=L
RX4L     EQU   *-RX4            RACXTRT LENGTH
*
RX6      RACROUTE REQUEST=EXTRACT,WORKA=ZEROS,RELEASE=1.8.1,           X
```

```
               TYPE=EXTRACT,FIELDS=ZEROS,ENTITY=ZEROS,                X
               CLASS='GROUP',MF=L
RX6L     EQU   *-RX6              RACXTRT LENGTH
*
ICH2     ICHEINTY ALTER,TYPE='USR',ENTRY=ZEROS,ACTIONS=(ZEROS),       X
               MF=L
ICH2L    EQU   *-ICH2             ICHEINTY LENGTH
*
ACT2B    ICHEACTN FIELD=PASSDATE,FLDATA=(3,ZEROS),MF=L
ACT2BL   EQU   *-ACT2B
*
WTOX2    DS    ØF
WTO2     WTO '                        ',ROUTCDE=11,MF=L
WTO2L    EQU   *-WTO2             WTO LENGTH
*
* TSO WORK AREAS AND RACROUTE AREA
*
LOCPPL   DS    XL(PPLLEN)         PARSE PARAMETER LIST
LOCANS   DS    F                  ADDR OF PARM DESCRIPTOR LIST
LOCECB   DS    F                  ECB FOR PROCESSOR
PUTPWORK PUTLINE MF=L             PTPB
PUTPLEN  EQU   *-PUTPWORK
PUTIWORK DS    4F                 FOR IOPL
PUTILEN  EQU   *-PUTIWORK
RACWORK  DS    CL512              RACROUTE WORK AREA
WORKLEN  EQU   *-WORKAREA
*
* DSECTS
*
         PRINT NOGEN
         YREGS
         IKJCPPL                  CPPL
         IKJPSCB                  PSCB
         CVT DSECT=YES            CVT
         IEESMCA                  SMCA
         IHAACEE                  ACEE
         ICHPCGRP                 CGRP
         IHAASCB                  ASCB
         IHAASXB                  ASXB
         IHAPSA                   PSA
         IKJIOPL                  IOPL
         IKJTCB                   TCB
         IKJPPL                   PPL
PPLLEN   EQU   *-PPL
         END
```

Cloning resources

This is a small utility we use in our centre to deal with the definition of a new group of transactions in the RACF classes GCICSTRN (for the production environment) or G$TESTRN (for the test environment).

To administer security, we use a product that replicates the definition of the group in only a few aspects (owner, installation data, etc), but not the members of the group. So when we have to copy a definition of a test group to production, or create a new group on the basis of an existing one, we have to insert the members in the new group by hand.

This utility is able to clone the definition of a group of transactions (main data, members, permits) and can also add more transactions and permissions to the cloned ones. Finally, you can create a definition from scratch, explicitly adding the transactions and permissions you want.

After preparing the necessary RACF commands in a temporary dataset, you can view, execute, save, or discard them. The structure of this utility is generic and can be adapted easily to any kind of RACF resource. It's written in REXX and uses three panels, and so is independent from any external vendor product.

Just a note about the permit duplication. This code is written according to our standards – we have three types of permit:

- For Cedacri Ovest groups (RACF administrators, programmers, Help Desk, operators, etc).
- For decentralized administrators, who need access to alter the resources whose access list they control.
- For groups of external users, who need read-only access.

This can be adapted to your standards. This utility is composed of a REXX (RACFRGEN), five panels (RACFPGEN, RACFHGEN, RACFPGO, RACFPTRN, RACFPPER), and one file of messages (RACFM00).

RACFM00 MESSAGES

```
RACFMØØ1 'DEFINITION CREATED          ' .HELP = *        .ALARM=YES
'RACFMØØ1: DEFINITION READY TO EXECUTE'           '

RACFMØØ2 'INSTRUCTIONS SAVED          ' .HELP = *        .ALARM=YES
'RACFMØØ2: CIAO CIAO            '          '

RACFMØØ3 'TRANSACTIONS INSERTED       ' .HELP = *        .ALARM=YES
'RACFMØØ3:                      '          '

RACFMØØ4 'QUALIFIERS INSERTED         ' .HELP = *        .ALARM=YES
'RACFMØØ4:                      '          '

RACFMØØ5 'INSTRUCTIONS SAVED          ' .HELP = *        .ALARM=YES
'RACFMØØ5:                      '          '

RACFMØØ6 'INSTRUCTIONS EXECUTED       ' .HELP = *        .ALARM=YES
'RACFMØØ6:                      '          '

RACFMØØ7 'INSTRUCTIONS DISCARDED      ' .HELP = *        .ALARM=YES
'RACFMØØ7:                      '          '

RACFMØØ8 'GROUP NON-EXISTENT          ' .HELP = *        .ALARM=YES
'RACFMØØ8:                      '          '

RACFMØØ9 'INSTRUCTION CANCELLATION    ' .HELP = *        .ALARM=YES
'RACFMØØ9:                      '          '
```

RACFPTRN PANEL

```
)ATTR DEFAULT(%$_)
/*************************************************************/
/* insertion of transactions in a new RACF resource        */
/*************************************************************/
¬ TYPE(TEXT)  INTENS(HIGH) COLOR(PINK)
£ TYPE(TEXT)  INTENS(HIGH) COLOR(YELLOW)
% TYPE(TEXT)  INTENS(HIGH)
$ TYPE(TEXT)  INTENS(LOW)
_ TYPE(INPUT) INTENS(HIGH) CAPS(ON) JUST(LEFT) PADC(_)
# TYPE(INPUT) INTENS(HIGH) CAPS(ON) JUST(LEFT)
)BODY
$──────────¬INSERTION OF TRANSACTIONS$──────────
$===>#ZCMD
$
$   £Transactions$:
$
$
$   _T1  $_T2  $_T3  $_T4  $_T5  $_T6  $_T7  $_T8  $_T9  $_T1Ø $
$
```

```
$   _T11 $_T12 $_T13 $_T14 $_T15 $_T16 $_T17 $_T18 $_T19 $_T2Ø $
$
$   _T21 $_T22 $_T23 $_T24 $_T25 $_T26 $_T27 $_T28 $_T29 $_T3Ø $
$
$   _T31 $_T32 $_T33 $_T34 $_T35 $_T36 $_T37 $_T38 $_T39 $_T4Ø $
$
$   _T41 $_T42 $_T43 $_T44 $_T45 $_T46 $_T47 $_T48 $_T49 $_T5Ø $
$
$   _T51 $_T52 $_T53 $_T54 $_T55 $_T56 $_T57 $_T58 $_T59 $_T6Ø $
$
$
$  £More$==>_OT$     (Si/No)
)INIT
&OT = No
)REINIT
)PROC
ver(&OT,List,Si,SI,NO)
)END
```

RACFPPER PANEL

```
)ATTR DEFAULT(%$_)
/*************************************************************/
/* insert permissions in a new group of transactions        */
/*************************************************************/
¬ TYPE(TEXT)  INTENS(HIGH) COLOR(PINK)
£ TYPE(TEXT)  INTENS(HIGH) COLOR(YELLOW)
% TYPE(TEXT)  INTENS(HIGH)
$ TYPE(TEXT)  INTENS(LOW)
_ TYPE(INPUT) INTENS(HIGH) CAPS(ON) JUST(LEFT) PADC(_)
# TYPE(INPUT) INTENS(HIGH) CAPS(ON) JUST(LEFT)
)BODY
$──────────¬INSERTION OF PERMISSIONS$────────────
$===>#ZCMD
$
$   £CEDACRI Permit      $:
$
$    CEDOPE:_Z$CEDSVI:_Z$CEDNET:_Z$
$
$
$   £Devolved administrators permits$:
$
$    AL:_Z$ AT:_Z$ BI:_Z$ SV:_Z$ RO:_Z$ CC:_Z$ CG:_Z$ CH:_Z$ BA:_Z$
$
$
$   £Base groups permits$:
$
$    BA:_Z$ CB:_Z$ CD:_Z$ RO:_Z$
$
)INIT
```

```
.ZVARS = '(CEDOPE CEDSVI CEDNET AL AT BI SV RO CC CG CH BA BB CB CD RB)'
&CEDOPE = ''
&CEDSVI = ''
&CEDNET = ''
&AL     = ''
&AT     = ''
&BI     = ''
&SV     = ''
&RO     = ''
&CC     = ''
&CG     = ''
&CH     = ''
&BA     = ''
&BB     = ''
&CB     = ''
&CD     = ''
&RB     = ''
)REINIT
)END
```

RACFPGO PANEL

```
)ATTR
 _ TYPE(INPUT) CAPS(ON)
 £ TYPE(TEXT) INTENS(HIGH) COLOR(YELLOW)
 < TYPE(TEXT) INTENS(HIGH) COLOR(BLUE)
 { TYPE(TEXT) INTENS(HIGH) COLOR(TURQ)
)BODY EXPAND(//)
%-/-/- RACF COMMANDS PENDING -/-/-
%COMMAND ===>_PCMD
+
£       ************************************************************
£       *%                                                        £*
£       *%    COMMANDS GENERATED FOR RACF STILL PEND EXECUTION    £*
£       *%                                                        £*
£       ************************************************************
%
+          CHOOSE ONE OF THE FOLLOWING COMMANDS:
%
%      (£1%) {VIEW + REVIEW THE GENERATED RACF COMMANDS
%
%      (£2%) {SAVE + SAVE THE GENERATED RACF COMMANDS
%
%      (£3%) {DEL  + DELETE THE GENERATED RACF COMMANDS
%
%      (£4%) {EXEC + EXEC THE GENERATED RACF COMMANDS ONLINE
%
%      (£4%) {END  + DESCARD GENERATED COMMANDS
%
)PROC
```

```
 VER(&PCMD,LIST,VIEW,SAVE,EXEC,END,1,2,3,4)
)END
```

RACFHGEN PANEL

```
)ATTR DEFAULT(%+_)
¬ TYPE(TEXT)  INTENS(HIGH) COLOR(PINK)
£ TYPE(TEXT)  INTENS(HIGH) COLOR(YELLOW)
% TYPE(TEXT)  INTENS(HIGH)
)BODY
+----------¬CREATION OF A GROUP OF TRANSACTIONS+-------- --
%COMMAND ===>_ZCMD
+
%
%£Res. name
+   It's the resource name of the new group of transactions
+
%£Class
+   It's the RACF class in which the resource must be defined.
+   Test is G$TESTRN and Prod is GCICSTRN.
+
%£UACC
+   It's the Universal Access for the new resource being defined
+
%£Description
+   It's the installation data of the new resource being defined
+
%£Copy res.
+   It's the resource you want to clone completely
+
%£Transactions
+   If you want to add more transactions, specify yes
+
%£Permits
+   If you want to add more permissions, specify yes
+
+
+
)PROC
)END
```

RACFPGEN PANEL

```
)ATTR DEFAULT(%$_)
/*************************************************************/
/* creation of a new RACF group of transactions              */
/*************************************************************/
¬ TYPE(TEXT)  INTENS(HIGH) COLOR(PINK)
```

```
£ TYPE(TEXT)  INTENS(HIGH) COLOR(YELLOW)
% TYPE(TEXT)  INTENS(HIGH)
$ TYPE(TEXT)  INTENS(LOW)
_ TYPE(INPUT) INTENS(HIGH) CAPS(ON) JUST(LEFT) PADC(_)
# TYPE(INPUT) INTENS(HIGH) CAPS(ON) JUST(LEFT)
)BODY
$———————¬CREATION OF A GROUP OF TRANSACTIONS$———————
$===>#ZCMD
$
$   £Res. name   $===>_GRPNAME $    £Class   $===>_ENV $ (Test/Prod)
$
$
$   £UACC$(N/R)   ===>_U$
$
$
$   £Description$ ===>_DESC                                      $
$
$
$   £Copy res.   $===>_FROMGRP $    £Class   $===>_FENV$ (Test/Prod)
$
$
$   £Transactions$===>_TQ$ (Si/No)   insert more transactions ?
$
$
$   £Permits     $===>_PQ$ (Si/No)   insert more permits ?
$
$
$
$
)INIT
.help = racfhgen
)PROC
ver(&U,nb,LIST,R,N)
ver(&GRPNAME,PICT,AANNCCCC)
ver(&ENV,nb,list,TEST,PROD)
ver(&FENV,list,TEST,PROD)
ver(&TQ,list,SI,NO)
ver(&PQ,list,SI,NO)
)END
```

RACFRGEN

```
/* rexx */
/*************************************************/
/* Creation of a new RACF group of transactions */
/*************************************************/
msg = ''
ISPP = "'SYSO.RACF.ISPPLIB'"
ISPM = "'SYSO.RACF.ISPMLIB'"
ADDRESS ISPEXEC "LIBDEF ISPPLIB DATASET ID("ISPP") COND"
```

```
ADDRESS ISPEXEC "LIBDEF ISPMLIB DATASET ID("ISPM") COND"

filename = sysvar(sysuid)||'.T'||time(S)||'.RACFRGEN'
call msg 'off'
ok='Ø 4 8';"free fi(cmds)"
call msg 'on'
ok='Ø';"allocate fi (cmds) da ('"||filename||"') new space(2,1) cyl ,
   blksize(Ø) unit(339Ø) catalog lrecl(8Ø) recfm(f,b) dsorg(ps)"

cre = 'no'
ok='Ø 4 8';address ispexec "display panel(racfpgen)"
fine = rc
first = 'si'
do while fine = Ø
   cre = 'si'
   if first = 'si' then do
      /********** Creation of the group ********************/
      first = 'no'
      if env = 'PROD' then class = 'gcicstrn'
      else                 class = 'g$testrn'
      queue 'rdef' class grpname 'owner(cedsys) uacc('||u||')'
      ok='Ø';"execio 1 diskw cmds"
      if desc <> '' then do
         queue 'ralt' class grpname "data('" desc "')"
         ok='Ø';"execio 1 diskw cmds"
         end
      msg = 'RACFMØØ1'
      end
   /********** Copy  transactions from group ************/
   if fromgrp <> '' then do
      if fenv = 'PROD' then fclass = 'gcicstrn'
      else                  fclass = 'g$testrn'
      z = outtrap('fromline.')
      ok='Ø';"rl " fclass fromgrp "auth"
      esito = rc
      z = outtrap('off')
      if esito > Ø then do
         msg = 'RACFMØØ8'
         ok='Ø';address ispexec "control display refresh"
     ok='Ø 4 8';address ispexec "display panel(racfpgen) msg("||msg||")"
         fine = rc
         iterate
         end
      finetran = 'no'
      i = 1
      do while finetran = 'no'
         if index(fromline.i,'RESOURCES IN GROUP') > Ø then do
            i = i + 2
            do while substr(fromline.i,2,4) <> '    '
               tran = substr(fromline.i,1,4)
```

```
            queue 'ralt '||class||' '||grpname||' addmem('||tran||')'
            ok='Ø';"execio 1 diskw cmds"
            i = i + 1
            end
         finetran = 'si'
         end
      i = i + 1
      end  /* fine do finetran */
   fineauth = 'no'
   do while fineauth = 'no'
      if index(fromline.i,'USER      ACCESS  ') > Ø then do
         i = i + 2
         do while substr(fromline.i,2,4) <> '    '
            parse var fromline.i authid access .
            com = 'pe '||grpname||' class('||class||') id('||authid
            queue com||') access('||access||')'
            ok='Ø';"execio 1 diskw cmds"
            i = i + 1
            end
         fineauth = 'si'
         end
      i = i + 1
      end  /* fine do fineauth */
   end     /* fine if fromgrp  */
/********** Insert    transactions in group  *********/
if tq = 'SI' then do
   tq = ''
   ok='Ø 4 8';address ispexec "display panel(racfptrn)"
   finetrn = rc
   do while finetrn = Ø
      do j = 1 to 6Ø
         indtrn = T||j
         trnname = value(indtrn,'')
         if trnname = '' then iterate
         queue 'ralt' class grpname 'addmem('||trnname||')'
         ok='Ø';"execio 1 diskw cmds"
         end
      if ot = 'SI' then do
         msg = 'RACFMØØ3'
         ok='Ø';address ispexec "control display refresh"
     ok='Ø';address ispexec "display panel(racfptrn) msg("||msg||")"
         finetrn = rc
         end
      else leave
      end
   end
/************** Insert permit in group **************/
riga = 'pe '||grpname||' class('||class||') id('
if pq = 'SI' then do
   pq = ''
   queue 'pe' grpname ' class(' class ') id(cedsys) access(alter)'
```

```
      ok='Ø';"execio 1 diskw cmds"
      ok='Ø 4 8';address ispexec "display panel(racfpper)"
      fineper = rc
      do while fineper = Ø
         if cedope <> '' then queue riga||'cedope) access(read)'
         if cedsvi <> '' then queue riga||'cedsvi) access(read)'
         if cednet <> '' then queue riga||'cednet) access(read)'
         if al <> ''     then queue riga||'cralci) access(alter)'
         if at <> ''     then queue riga||'cratci) access(alter)'
         if bi <> ''     then queue riga||'crbici) access(alter)'
         if sv <> ''     then queue riga||'crsvci) access(alter)'
         if ro <> ''     then queue riga||'crroci) access(alter)'
         if cc <> ''     then queue riga||'crccci) access(alter)'
         if cg <> ''     then queue riga||'crcgci) access(alter)'
         if ch <> ''     then queue riga||'crcpci) access(alter)'
         if ba <> ''     then queue riga||'crbaci) access(alter)'
         if bb <> ''     then queue riga||'crbaciØØ) access(read)'
         if cb <> ''     then queue riga||'crcbciØØ) access(read)'
         if cd <> ''     then queue riga||'crcdciØØ) access(read)'
         if rb <> ''     then queue riga||'crrociØØ) access(read)'
         ok='Ø';"execio * diskw cmds"
         msg = 'RACFMØØ4'
         ok='Ø';address ispexec "control display refresh"
       ok='Ø 4 8';address ispexec "display panel(racfpper) msg("||msg||")"
         fineper = rc
         end
       end
   ok='Ø';address ispexec "control display refresh"
   ok='Ø 4 8';address ispexec "display panel(racfpgen) msg("||msg||")"
   fine = rc
   end      /* fine do principale */
ok='Ø';"execio Ø diskw cmds (finis"

/************ execution of pending commands *************/
if cre = 'si' then do
   erase = 'si'
   ok='Ø 4 8';address ispexec "display panel(racfpgo)"
   fine = rc
   do while fine = Ø
      select
         when pcmd = 1 then do
              ok='Ø';"rev " filename
              end
         when pcmd = 2 then do
              erase = 'no'
              msg = 'racfmØØ5'
              end
         when pcmd = 3 then do
              "delete '"||filename||"'"
              msg = 'racfmØØ9'
              end
```

```
         when pcmd = 4 then do
              "exec '"||filename||"'"
              msg = 'racfm006'
              end
         when pcmd = 5 then do
              erase = 'no'
              msg = 'racfm007'
              end
         otherwise
         end
      pcmd = ''
      ok='0';address ispexec "control display refresh"
      ok='0 4 8';address ispexec "display panel(racfpgo) msg("||msg||")"
      fine = rc
      end
   end
/*********************************************************************/
call cleanup
exit
/*********************************************************************/

cleanup:
ADDRESS ISPEXEC "LIBDEF ISPPLIB"
ADDRESS ISPEXEC "LIBDEF ISPMLIB"
delstack
call msg 'off'
if erase = 'si' then do
   ok='0';"delete " filename
   end
ok='0';"free fi (cmds)"
call msg 'on'
return

errproc:
if condition('c')='error' & symbol('ok')='var' then,
   if wordpos(rc,ok)>0 | ok='*' then return
signal off syntax
signal off novalue
call off error
error_type = condition('c')
say error_type 'alla linea' sigl ':' condition('d')
if error_type = 'syntax' then say errortext(rc)
if error_type = 'error' & symbol('zerrlm') = 'var',
   then say zerrlm
call cleanup
```

Maria Elena Campidoglio
Systems Programmer
Cedacri Ovest (Italy)

RACF news

RACF users can benefit from Release 14 of IBM's DFSORT sort, merge, copy, analysis, and reporting option for OS/390 and MVS/ESA. The new release includes a range of enhancements for productivity, performance, capacity, and storage usage, and also simplified installation and customization.

Users can create and use symbols for their own data, and use symbols from IBM for data associated with RACF.

More INCLUDE/OMIT conditions and SUM fields allow users to write more complex filtering and totalling applications, while new OUTFIL features support multiple output records using the fields of each input record, split records, double and triple space in reports, and pad short fields.

For further information contact your local IBM representative.

* * *

Treehouse has announced Version 3.1.0 of Securitre, its security interface between ADABAS/NATURAL and RACF.

Securitre allows the storage of all security 'rules' in a single rule base of the security system and interfaces with RACF to determine user access to any ADABAS/ NATURAL resource. Access is controlled by user-id rather than password. The program pathing feature enables access to be restricted to specific users, programs, jobs, etc.

The Securitre NATURAL Security System Conversion Facility produces Securitre parameters and RACF rules from the NATURAL security system data, allowing transition to the single rule base.

For further information contact:
Treehouse Software, 409 Broad Street, Suite 140, Sewickly, PA 15143, USA.
Tel: (412) 741 1677.
URL: http://www.treehouse.com.

* * *

RACF users can benefit from Version 2.2 of Computer Associates' Unicenter TNG and its underlying TNG Framework. Enhancements include bi-directional policy synchronization with RACF, and improved cross-platform scheduling agents for AS/400 and NetWare.

Built-in network management capabilities include support for multi-homed devices, instant DHCP discovery and identification, and RMON analysis and MIB II support. It monitors and manages the health and performance between any two end-points in the network. It also manages and monitors ATM and Frame Relay networks.

For further information contact:
Computer Associates, One Computer Associates Plaza, Islandia, NY 11788-7000, USA.
Tel: (516) 342 5224.
Computer Associates, Computer Associates House, 183-187 Bath Road, Slough, SL1 4AA, UK.
Tel: (01753) 5777733
URL: http://www.cai.com.