15

RACF update

February 1999

In this issue

RACF Update

Published by

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

North American office

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

Contributions

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

RACF Update on-line

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

Editor

Robert Burgess

Disclaimer

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

Subscriptions and back-issues

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

Printed in England.

Retrieving revoked users

SETTING UP NETWORK SECURITY

There are two main concerns for network security – protecting the system against attacks from the outside world and protecting against attacks from within the system. Protection against the outside world requires the prevention of unauthorized access by other systems, and prevention of unauthorized use of TPs by users. Protection against attacks from the inside covers the prevention of an intruder masquerading as a local LU or a server TP for a local LU.

Protection is primarily accomplished with RACF profiles; however, there are three exceptions:

- The default level of conversation security accepted by the LU (on the APPL definition statement).
- The conditions under which LU-LU verification takes place (on the APPL definition statement).
- The security level of the TP (on the TP profile).

RACF provides two commands (DELUSER and DELGROUP) for removing user-ids or groups from the RACF database. However, not all occurrences of the specified IDs are removed by these commands. The entries in the access lists of resource profiles, ownership of profiles, etc are retained even after removing the user or group. If all the information relating to an ID is not removed, unwanted information can accumulate in the database. Moreover, if the same user-id is re-allocated later, the user may get access to a number of resources which are not required.

I have written some code, using standard RACF utilities, to retrieve revoked users. This should be submitted to JES.

```
//RACFREVO JOB  SYS,
//         'psy',
//         MSGLEVEL=(1,1),
//         CLASS=W,
//         MSGCLASS=9,
//         COND=(Ø,LT),
//         NOTIFY=&SYSUID
//*******************************************************************
//*        REVOKES all user-ids  without log-on for..               *
//*        45 days                                                  *
//*        output in aclist file USERID.EXEC.RACF.CLIST             *
//*******************************************************************
//SEARCHA  EXEC PGM=IKJEFT1A
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
PROFILE PREFIX(psy)
SEARCH CLASS(USER) AGE(45) FILTER(YSVØØA*) CLIST(' ALTUSER ' ' REVOKE')
/*
//REVOKEA  EXEC PGM=IKJEFT1A
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 EXEC 'psy.EXEC.RACF.CLIST'
/*
//SEARCHB  EXEC PGM=IKJEFT1A
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
PROFILE PREFIX(psy)
SEARCH CLASS(USER) AGE(45) FILTER(YSVØØB*) CLIST(' ALTUSER ' ' REVOKE')
/*
//REVOKEB  EXEC PGM=IKJEFT1A
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 EXEC 'psy.EXEC.RACF.CLIST'
/*
//SEARCHC  EXEC PGM=IKJEFT1A
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
PROFILE PREFIX(psy)
SEARCH CLASS(USER) AGE(45) FILTER(YSVØØC*) CLIST(' ALTUSER ' ' REVOKE')
/*
//REVOKEC  EXEC PGM=IKJEFT1A
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 EXEC 'psy.EXEC.RACF.CLIST'
/*
//SEARCHD  EXEC PGM=IKJEFT1A
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
PROFILE PREFIX(psy)
SEARCH CLASS(USER) AGE(45) FILTER(YSVØØD*) CLIST(' ALTUSER ' ' REVOKE')
/*
//REVOKED  EXEC PGM=IKJEFT1A
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 EXEC 'psy.EXEC.RACF.CLIST'
```

```
/*
//SEARCHE  EXEC PGM=IKJEFT1A
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
PROFILE PREFIX(psy)
SEARCH CLASS(USER) AGE(45) FILTER(YSVØØE*) CLIST(' ALTUSER ' ' REVOKE')
/*
//REVOKEE  EXEC PGM=IKJEFT1A
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 EXEC 'psy.EXEC.RACF.CLIST'
/*
//SEARCHF  EXEC PGM=IKJEFT1A
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
PROFILE PREFIX(psy)
SEARCH CLASS(USER) AGE(45) FILTER(YSVØØF*) CLIST(' ALTUSER ' ' REVOKE')
/*
//REVOKEF  EXEC PGM=IKJEFT1A
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 EXEC 'psy.EXEC.RACF.CLIST'
/*
//SEARCHK  EXEC PGM=IKJEFT1A
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
PROFILE PREFIX(psy)
SEARCH CLASS(USER) AGE(45) FILTER(YSVØØK*) CLIST(' ALTUSER ' ' REVOKE')
/*
//REVOKEK  EXEC PGM=IKJEFT1A
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 EXEC 'psy.EXEC.RACF.CLIST'
/*
//SEARCHL  EXEC PGM=IKJEFT1A
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
PROFILE PREFIX(psy)
SEARCH CLASS(USER) AGE(45) FILTER(YSVØØL*) CLIST(' ALTUSER ' ' REVOKE')
/*
//REVOKEL  EXEC PGM=IKJEFT1A
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 EXEC 'psy.EXEC.RACF.CLIST'
/*
//SEARCHO  EXEC PGM=IKJEFT1A
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
PROFILE PREFIX(psy)
SEARCH CLASS(USER) AGE(45) FILTER(YSVØØO*) CLIST(' ALTUSER ' ' REVOKE')
```

```
/*
//REVOKEO  EXEC PGM=IKJEFT1A
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 EXEC 'psy.EXEC.RACF.CLIST'
/*
//SEARCHR  EXEC PGM=IKJEFT1A
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
PROFILE PREFIX(psy)
SEARCH CLASS(USER) AGE(45) FILTER(YSVØØR*) CLIST(' ALTUSER ' ' REVOKE')
/*
//REVOKER  EXEC PGM=IKJEFT1A
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 EXEC 'psy.EXEC.RACF.CLIST'
/*
//SEARCHZ  EXEC PGM=IKJEFT1A
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
PROFILE PREFIX(psy)
SEARCH CLASS(USER) AGE(45) FILTER(YSVØØZ*) CLIST(' ALTUSER ' ' REVOKE')
/*
//REVOKEZ  EXEC PGM=IKJEFT1A
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 EXEC 'psy.EXEC.RACF.CLIST'
/*
//
//
//
//
//
//
SEARCH CLASS(USER) AGE(45) FILTER(YSVØØS*) CLIST(' LISTUSER ' ' ')

    users not used for one year (in TSO command mode)
/*****************************************************/
/*  REVOKES all users not connected for one year       */
/*  output in a clist file USERID.EXEC.RACF.CLIST      */
/*****************************************************/
SEARCH CLASS(USER) AGE(362) CLIST(' ALTUSER ' ' REVOKE')
```

The following code is for retrieving activity from an SMF file for a particular terminal, where T69000CT is the netname:

```
//AUDITRAC EXEC PGM=IKJEFTØ1
//SYSPRINT DD SYSOUT=9
//SYSTSPRT DD SYSOUT=9
//SORTLIB  DD DSN=SYS1.SORTLIB,DISP=SHR
```

```
//RSMFIN   DD DSN=your.smf.file,DISP=SHR
//SYSTSIN  DD *,DLM=XX
RACFRW TITLE ('LIST netname access for terminal T69000CT')
SELECT DATE(97001:97351) TERMINAL(T69000CT)
EVENT LOGON
LIST SORT(USER DATE TIME)
END
XX
/*
```

Claude Dunand (France) © Xephon 1999

Replacement for the RACF Report Writer – part 4

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 obtained in the usual way. This service is free to subscribers.*

```
                     ;
LABEL OPNCLASS = 'Class name'
      OPNUSERN = 'User name'
      OPNUTKNE = 'Utoken encr.?'
      OPNUPRE  = 'Pre-1.9?'
      OPNUVFYX = 'VERIFYX propagation?'
      OPNUNJEU = 'Undefined NJE user?'
      OPNUUAUD = 'UAUDIT?'
      OPNUSPEC = 'RACF special?'
      OPNUDFLT = 'Default token?'
      OPNUUNDF = 'Undefined user?'
      OPNUERR  = 'Token in error?'
      OPNUTRST = 'User trusted?'
      OPNUSEST = 'Session type'
      OPNUSURO = 'Surrogate user?'
      OPNURMT  = 'Remote job?'
      OPNUPRVL = 'Privileged user?'
      OPNUSECL = 'User SECLABEL'
      OPNUEXND = 'Execution node'
```

```
OPNUSUSR = 'Submitting user'
OPNUSNOD = 'Submitting node'
OPNUSGRP = 'Submitting group'
OPNUSPOE = 'Port of entry'
OPNUSPCL = 'Class of POE'
OPNUTUSR = 'Userid'
OPNUTGRP = 'Groupid'
OPNUTDFT = 'Default group?'
OPNUTSEC = 'Default SECLABEL?'
OPNAPPC  = 'APPC key link'
OPNAUDIT = 'Audit code'
OPNORUID = 'Old real UID'
OPNOEUID = 'Old effective UID'
OPNOSUID = 'Old saved UID'
OPNORGID = 'Old real GID'
OPNOEGID = 'Old effective GID'
OPNOSGID = 'Old saved GID'
OPNPATHN = 'Path name'
OPNFILID = 'File id'
OPNFOUID = 'Owner UID'
OPNFOGID = 'Owner GID'
OPNOLSGI = 'Old S_ISGID requested?'
OPNOLSUI = 'Old S_ISUID requested?'
OPNOLSVT = 'Old S_ISVTX requested?'
OPNOLORD = 'Old Owner read?'
OPNOLOWR = 'Old Owner write?'
OPNOLOEX = 'Old Owner exec?'
OPNOLGRD = 'Old Group read?'
OPNOLGWR = 'Old Group write?'
OPNOLGEX = 'Old Group exec?'
OPNOLWRD = 'Old Other read?'
OPNOLWWR = 'Old Other write?'
OPNOLWEX = 'Old Other exec?'
OPNNWSGI = 'New S_ISGID requested?'
OPNNWSUI = 'New S_ISUID requested?'
OPNNWSVT = 'New S_ISVTX requested?'
OPNNWORD = 'New Owner read?'
OPNNWOWR = 'New Owner write?'
OPNNWOEX = 'New Owner exec?'
OPNNWGRD = 'New Group read?'
OPNNWGWR = 'New Group write?'
OPNNWGEX = 'New Group exec?'
OPNNWWRD = 'New Other read?'
OPNNWWWR = 'New Other write?'
OPNNWWEX = 'New Other exec?'
OPNNWURE = 'New user aud read'
OPNNWUWR = 'New user aud write'
OPNNWUEX = 'New user aud exec'
OPNNWARE = 'New auditor aud read'
OPNNWAWR = 'New auditor aud write'
OPNNWAEX = 'New auditor aud exec'
```

```
      OPNRQSGI = 'Req S_ISGID?'
      OPNRQSUI = 'Req S_ISUID?'
      OPNRQSVT = 'Req S_ISVTX?'
      OPNRQORD = 'Req Owner read?'
      OPNRQOWR = 'Req Owner write?'
      OPNRQOEX = 'Req Owner exec?'
      OPNRQGRD = 'Req Group read?'
      OPNRQGWR = 'Req Group write?'
      OPNRQGEX = 'Req Group exec?'
      OPNRQWRD = 'Req Other read?'
      OPNRQWWR = 'Req Other write?'
      OPNRQWEX = 'Req Other exec?'
      OPNFILPL = 'File pool'
      OPNFILSP = 'File space'
      OPNINODE = 'Inode'
      OPNSCID  = 'File SCID'
      ;
        OUTPUT RACF.OPENFILE;
      END;
    %END;
%MEND OPENFILE;
./        ADD   LIST=ALL,NAME=PTRACE
%MACRO PTRACE(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
      %PUT Including variables from PTRACE extension;
      RACF.PTRACE (KEEP=%SMFHDR
                        %SMF8ØHDR(REQ=DEFINE)
                        PTRCLASS
                        PTRUSERN
                        PTRUTKNE
                        PTRUPRE
                        PTRUVFYX
                        PTRUNJEU
                        PTRUUAUD
                        PTRUSPEC
                        PTRUDFLT
                        PTRUUNDF
                        PTRUERR
                        PTRUTRST
                        PTRUSEST
                        PTRUSURO
                        PTRURMT
                        PTRUPRVL
                        PTRUSECL
                        PTRUEXND
                        PTRUSUSR
                        PTRUSNOD
                        PTRUSGRP
                        PTRUSPOE
```

```
                      PTRUSPCL
                      PTRUTUSR
                      PTRUTGRP
                      PTRUTDFT
                      PTRUTSEC
                      PTRAPPC
                      PTRAUDIT
                      PTRORUID
                      PTROEUID
                      PTROSUID
                      PTRORGID
                      PTROEGID
                      PTROSGID
                      PTRTRUID
                      PTRTEUID
                      PTRTSUID
                      PTRTRGID
                      PTRTEGID
                      PTRTSGID
                      PTRTPID
                )
  %END;
%IF &REQ = EXTRACT %THEN
  %DO;
    %PUT Including datadefinition for PTRACE extension;
    WHEN('PTRACE') DO;
      INPUT %SMF8ØHDR(REQ=EXTRACT)
            PTRCLASS $       282-289
            PTRUSERN $       291-31Ø
            PTRUTKNE $       312-315
            PTRUPRE  $       317-32Ø
            PTRUVFYX $       322-325
            PTRUNJEU $       327-33Ø
            PTRUUAUD $       332-335
            PTRUSPEC $       337-34Ø
            PTRUDFLT $       342-345
            PTRUUNDF $       347-35Ø
            PTRUERR  $       352-355
            PTRUTRST $       357-36Ø
            PTRUSEST $       362-369
            PTRUSURO $       371-374
            PTRURMT  $       376-379
            PTRUPRVL $       381-384
            PTRUSECL $       386-393
            PTRUEXND $       395-4Ø2
            PTRUSUSR $       4Ø4-411
            PTRUSNOD $       413-42Ø
            PTRUSGRP $       422-429
            PTRUSPOE $       431-438
            PTRUSPCL $       44Ø-447
            PTRUTUSR $       449-456
```

```
              PTRUTGRP $        458-465
              PTRUTDFT $        467-47Ø
              PTRUTSEC $        472-475
              PTRAPPC  $        477-492
              PTRAUDIT $        494-5Ø4
              PTRORUID          5Ø6-515
              PTROEUID          517-526
              PTROSUID          528-537
              PTRORGID          539-548
              PTROEGID          55Ø-559
              PTROSGID          561-57Ø
              PTRTRUID          572-581
              PTRTEUID          583-592
              PTRTSUID          594-6Ø3
              PTRTRGID          6Ø5-614
              PTRTEGID          616-625
              PTRTSGID          627-636
              PTRTPID           638-647
               ;
LABEL PTRCLASS = 'Class name'
      PTRUSERN = 'User name'
      PTRUTKNE = 'Utoken encr.?'
      PTRUPRE  = 'Pre-1.9?'
      PTRUVFYX = 'VERIFYX propagation?'
      PTRUNJEU = 'Undefined NJE user?'
      PTRUUAUD = 'UAUDIT?'
      PTRUSPEC = 'RACF special?'
      PTRUDFLT = 'Default token?'
      PTRUUNDF = 'Undefined user?'
      PTRUERR  = 'Token in error?'
      PTRUTRST = 'User trusted?'
      PTRUSEST = 'Session type'
      PTRUSURO = 'Surrogate user?'
      PTRURMT  = 'Remote job?'
      PTRUPRVL = 'Privileged user?'
      PTRUSECL = 'User SECLABEL'
      PTRUEXND = 'Execution node'
      PTRUSUSR = 'Submitting user'
      PTRUSNOD = 'Submitting node'
      PTRUSGRP = 'Submitting group'
      PTRUSPOE = 'Port of entry'
      PTRUSPCL = 'Class of POE'
      PTRUTUSR = 'Userid'
      PTRUTGRP = 'Groupid'
      PTRUTDFT = 'Default group?'
      PTRUTSEC = 'Default SECLABEL?'
      PTRAPPC  = 'APPC key link'
      PTRAUDIT = 'Audit code'
      PTRORUID = 'Old real UID'
      PTROEUID = 'Old effective UID'
      PTROSUID = 'Old saved UID'
      PTRORGID = 'Old real GID'
      PTROEGID = 'Old effective GID'
      PTROSGID = 'Old saved GID'
      PTRTRUID = 'Tgt. real UID'
      PTRTEUID = 'Tgt. effective UID'
      PTRTSUID = 'Tgt. saved UID'
      PTRTRGID = 'Tgt. real GID'
      PTRTEGID = 'Tgt. effective GID'
      PTRTSGID = 'Tgt. saved GID'
      PTRTPID  = 'Tgt. process ID'
      ;
        OUTPUT RACF.PTRACE;
      END;
    %END;
%MEND PTRACE;
./        ADD   LIST=ALL,NAME=RENAMEF
%MACRO RENAMEF(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
      %PUT Including variables from RENAMEF extension;
      RACF.RENAMEF (KEEP=%SMFHDR
                         %SMF8ØHDR(REQ=DEFINE)
                         RENCLASS
                         RENUSERN
                         RENUTKNE
                         RENUPRE
                         RENUVFYX
                         RENUNJEU
                         RENUUAUD
                         RENUSPEC
                         RENUDFLT
                         RENUUNDF
                         RENUERR
                         RENUTRST
                         RENUSEST
                         RENUSURO
                         RENURMT
                         RENUPRVL
                         RENUSECL
                         RENUEXND
                         RENUSUSR
                         RENUSNOD
                         RENUSGRP
                         RENUSPOE
                         RENUSPCL
                         RENUTUSR
                         RENUTGRP
                         RENUTDFT
```

```
                       RENUTSEC
                       RENAPPC
                       RENAUDIT
                       RENORUID
                       RENOEUID
                       RENOSUID
                       RENORGID
                       RENOEGID
                       RENOSGID
                       RENPATHN
                       RENFILID
                       RENFOUID
                       RENFOGID
                       RENPATH2
                       RENFID2
                       RENDOUID
                       RENDOGID
                       RENPTHTP
                       RENLSTDL
                       RENFILPL
                       RENFILSP
                       RENINODE
                       RENSCID
                       RENFILP2
                       RENFILS2
                       RENINOD2
                       RENSCID2
                       RENDCELK
                       RENAUTYP
                )
  %END;
%IF &REQ = EXTRACT %THEN
  %DO;
    %PUT Including datadefinition for RENAMEF extension;
    WHEN('RENAMEF') DO;
      INPUT %SMF8ØHDR(REQ=EXTRACT)
            RENCLASS $       282-289
            RENUSERN $       291-31Ø
            RENUTKNE $       312-315
            RENUPRE  $       317-32Ø
            RENUVFYX $       322-325
            RENUNJEU $       327-33Ø
            RENUUAUD $       332-335
            RENUSPEC $       337-34Ø
            RENUDFLT $       342-345
            RENUUNDF $       347-35Ø
            RENUERR  $       352-355
            RENUTRST $       357-36Ø
            RENUSEST $       362-369
            RENUSURO $       371-374
            RENURMT  $       376-379
```

```
              RENUPRVL $       381-384
              RENUSECL $       386-393
              RENUEXND $       395-402
              RENUSUSR $       404-411
              RENUSNOD $       413-420
              RENUSGRP $       422-429
              RENUSPOE $       431-438
              RENUSPCL $       440-447
              RENUTUSR $       449-456
              RENUTGRP $       458-465
              RENUTDFT $       467-470
              RENUTSEC $       472-475
              RENAPPC  $       477-492
              RENAUDIT $       494-504
              RENORUID         506-515
              RENOEUID         517-526
              RENOSUID         528-537
              RENORGID         539-548
              RENOEGID         550-559
              RENOSGID         561-570
              RENPATHN $       572-771
              RENFILID $      1596-1627
              RENFOUID        1629-1638
              RENFOGID        1640-1649
              RENPATH2 $      1651-1850
              RENFID2  $      2675-2706
              RENDOUID        2708-2717
              RENDOGID        2719-2728
              RENPTHTP $      2730-2733
              RENLSTDL $      2735-2738
              RENFILPL $      2740-2747
              RENFILSP $      2749-2756
              RENINODE        2758-2767
              RENSCID         2769-2778
              RENFILP2 $      2780-2787
              RENFILS2 $      2789-2796
              RENINOD2        2798-2807
              RENSCID2        2809-2818
              RENDCELK $      2820-2835
              RENAUTYP $      2837-2849
               ;
LABEL RENCLASS = 'Class name'
      RENUSERN = 'User name'
      RENUTKNE = 'Utoken encr.?'
      RENUPRE  = 'Pre-1.9?'
      RENUVFYX = 'VERIFYX propagation?'
      RENUNJEU = 'Undefined NJE user?'
      RENUUAUD = 'UAUDIT?'
      RENUSPEC = 'RACF special?'
      RENUDFLT = 'Default token?'
      RENUUNDF = 'Undefined user?'
      RENUERR  = 'Token in error?'
      RENUTRST = 'User trusted?'
      RENUSEST = 'Session type'
      RENUSURO = 'Surrogate user?'
      RENURMT  = 'Remote job?'
      RENUPRVL = 'Privileged user?'
      RENUSECL = 'User SECLABEL'
      RENUEXND = 'Execution node'
      RENUSUSR = 'Submitting user'
      RENUSNOD = 'Submitting node'
      RENUSGRP = 'Submitting group'
      RENUSPOE = 'Port of entry'
      RENUSPCL = 'Class of POE'
      RENUTUSR = 'Userid'
      RENUTGRP = 'Groupid'
      RENUTDFT = 'Default group?'
      RENUTSEC = 'Default SECLABEL?'
      RENAPPC  = 'APPC key link'
      RENAUDIT = 'Audit code'
      RENORUID = 'Old real UID'
      RENOEUID = 'Old effective UID'
      RENOSUID = 'Old saved UID'
      RENORGID = 'Old real GID'
      RENOEGID = 'Old effective GID'
      RENOSGID = 'Old saved GID'
      RENPATHN = 'Path name'
      RENFILID = 'File id'
      RENFOUID = 'Owner UID'
      RENFOGID = 'Owner GID'
      RENPATH2 = '2nd path name'
      RENFID2  = '2nd file id'
      RENDOUID = 'UID of del. file'
      RENDOGID = 'GID of del. file'
      RENPTHTP = 'Path type'
      RENLSTDL = 'Last link del.?'
      RENFILPL = 'File pool'
      RENFILSP = 'File space'
      RENINODE = 'Inode'
      RENSCID  = 'File SCID'
      RENFILP2 = '2nd file pool'
      RENFILS2 = '2nd file space'
      RENINOD2 = '2nd inode'
      RENSCID2 = '2nd file SCID'
      RENDCELK = 'DCE link'
      RENAUTYP = 'Request type'
      ;
        OUTPUT RACF.RENAMEF;
      END;
    %END;
%MEND RENAMEF;
```

```
./        ADD   LIST=ALL,NAME=RMDIR
%MACRO RMDIR(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
      %PUT Including variables from RMDIR extension;
      RACF.RMDIR (KEEP=%SMFHDR
                       %SMF8ØHDR(REQ=DEFINE)
                       RMDCLASS
                       RMDUSERN
                       RMDUTKNE
                       RMDUPRE
                       RMDUVFYX
                       RMDUNJEU
                       RMDUUAUD
                       RMDUSPEC
                       RMDUDFLT
                       RMDUUNDF
                       RMDUERR
                       RMDUTRST
                       RMDUSEST
                       RMDUSURO
                       RMDURMT
                       RMDUPRVL
                       RMDUSECL
                       RMDUEXND
                       RMDUSUSR
                       RMDUSNOD
                       RMDUSGRP
                       RMDUSPOE
                       RMDUSPCL
                       RMDUTUSR
                       RMDUTGRP
                       RMDUTDFT
                       RMDUTSEC
                       RMDAPPC
                       RMDAUDIT
                       RMDORUID
                       RMDOEUID
                       RMDOSUID
                       RMDORGID
                       RMDOEGID
                       RMDOSGID
                       RMDPATHN
                       RMDFILID
                       RMDFOUID
                       RMDFOGID
                       RMDFILPL
                       RMDFILSP
                       RMDINODE
```

```
                     RMDSCID
                     RMDDCELK
                     RMDAUTYP
                )
  %END;
%IF &REQ = EXTRACT %THEN
  %DO;
    %PUT Including datadefinition for RMDIR extension;
    WHEN('RMDIR') DO;
      INPUT %SMF80HDR(REQ=EXTRACT)
            RMDCLASS $       282-289
            RMDUSERN $       291-310
            RMDUTKNE $       312-315
            RMDUPRE  $       317-320
            RMDUVFYX $       322-325
            RMDUNJEU $       327-330
            RMDUUAUD $       332-335
            RMDUSPEC $       337-340
            RMDUDFLT $       342-345
            RMDUUNDF $       347-350
            RMDUERR  $       352-355
            RMDUTRST $       357-360
            RMDUSEST $       362-369
            RMDUSURO $       371-374
            RMDURMT  $       376-379
            RMDUPRVL $       381-384
            RMDUSECL $       386-393
            RMDUEXND $       395-402
            RMDUSUSR $       404-411
            RMDUSNOD $       413-420
            RMDUSGRP $       422-429
            RMDUSPOE $       431-438
            RMDUSPCL $       440-447
            RMDUTUSR $       449-456
            RMDUTGRP $       458-465
            RMDUTDFT $       467-470
            RMDUTSEC $       472-475
            RMDAPPC  $       477-492
            RMDAUDIT $       494-504
            RMDORUID         506-515
            RMDOEUID         517-526
            RMDOSUID         528-537
            RMDORGID         539-548
            RMDOEGID         550-559
            RMDOSGID         561-570
            RMDPATHN $       572-771
            RMDFILID $      1596-1627
            RMDFOUID        1629-1638
            RMDFOGID        1640-1649
            RMDFILPL $      1651-1658
```

```
              RMDFILSP $      166Ø-1667
              RMDINODE        1669-1678
              RMDSCID         168Ø-1689
              RMDDCELK $      1691-17Ø6
              RMDAUTYP $      17Ø8-172Ø
               ;
LABEL RMDCLASS = 'Class name'
      RMDUSERN = 'User name'
      RMDUTKNE = 'Utoken encr.?'
      RMDUPRE  = 'Pre-1.9?'
      RMDUVFYX = 'VERIFYX propagation?'
      RMDUNJEU = 'Undefined NJE user?'
      RMDUUAUD = 'UAUDIT?'
      RMDUSPEC = 'RACF special?'
      RMDUDFLT = 'Default token?'
      RMDUUNDF = 'Undefined user?'
      RMDUERR  = 'Token in error?'
      RMDUTRST = 'User trusted?'
      RMDUSEST = 'Session type'
      RMDUSURO = 'Surrogate user?'
      RMDURMT  = 'Remote job?'
      RMDUPRVL = 'Privileged user?'
      RMDUSECL = 'User SECLABEL'
      RMDUEXND = 'Execution node'
      RMDUSUSR = 'Submitting user'
      RMDUSNOD = 'Submitting node'
      RMDUSGRP = 'Submitting group'
      RMDUSPOE = 'Port of entry'
      RMDUSPCL = 'Class of POE'
      RMDUTUSR = 'Userid'
      RMDUTGRP = 'Groupid'
      RMDUTDFT = 'Default group?'
      RMDUTSEC = 'Default SECLABEL?'
      RMDAPPC  = 'APPC key link'
      RMDAUDIT = 'Audit code'
      RMDORUID = 'Old real UID'
      RMDOEUID = 'Old effective UID'
      RMDOSUID = 'Old saved UID'
      RMDORGID = 'Old real GID'
      RMDOEGID = 'Old effective GID'
      RMDOSGID = 'Old saved GID'
      RMDPATHN = 'Path name'
      RMDFILID = 'File id'
      RMDFOUID = 'Owner UID'
      RMDFOGID = 'Owner GID'
      RMDFILPL = 'File pool'
      RMDFILSP = 'File space'
      RMDINODE = 'Inode'
      RMDSCID  = 'File SCID'
      RMDDCELK = 'DCE link'
      RMDAUTYP = 'Request type'
      ;
        OUTPUT RACF.RMDIR;
      END;
    %END;
%MEND RMDIR;
./        ADD   LIST=ALL,NAME=SETEGID
%MACRO SETEGID(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
      %PUT Including variables from SETEGID extension;
      RACF.SETEGID (KEEP=%SMFHDR
                         %SMF8ØHDR(REQ=DEFINE)
                         SEGCLASS
                         SEGUSERN
                         SEGUTKNE
                         SEGUPRE
                         SEGUVFYX
                         SEGUNJEU
                         SEGUUAUD
                         SEGUSPEC
                         SEGUDFLT
                         SEGUUNDF
                         SEGUERR
                         SEGUTRST
                         SEGUSEST
                         SEGUSURO
                         SEGURMT
                         SEGUPRVL
                         SEGUSECL
                         SEGUEXND
                         SEGUSUSR
                         SEGUSNOD
                         SEGUSGRP
                         SEGUSPOE
                         SEGUSPCL
                         SEGUTUSR
                         SEGUTGRP
                         SEGUTDFT
                         SEGUTSEC
                         SEGAPPC
                         SEGAUDIT
                         SEGORUID
                         SEGOEUID
                         SEGOSUID
                         SEGORGID
                         SEGOEGID
                         SEGOSGID
                         SEGNRGID
                         SEGNEGID
                         SEGNSGID
                         SEGGID
                   )
    %END;
  %IF &REQ = EXTRACT %THEN
    %DO;
      %PUT Including datadefinition for SETEGID extension;
      WHEN('SETEGID') DO;
        INPUT %SMF8ØHDR(REQ=EXTRACT)
              SEGCLASS $       282-289
              SEGUSERN $       291-31Ø
              SEGUTKNE $       312-315
              SEGUPRE  $       317-32Ø
              SEGUVFYX $       322-325
              SEGUNJEU $       327-33Ø
              SEGUUAUD $       332-335
              SEGUSPEC $       337-34Ø
              SEGUDFLT $       342-345
              SEGUUNDF $       347-35Ø
              SEGUERR  $       352-355
              SEGUTRST $       357-36Ø
              SEGUSEST $       362-369
              SEGUSURO $       371-374
              SEGURMT  $       376-379
              SEGUPRVL $       381-384
              SEGUSECL $       386-393
              SEGUEXND $       395-4Ø2
              SEGUSUSR $       4Ø4-411
              SEGUSNOD $       413-42Ø
              SEGUSGRP $       422-429
              SEGUSPOE $       431-438
              SEGUSPCL $       44Ø-447
              SEGUTUSR $       449-456
              SEGUTGRP $       458-465
              SEGUTDFT $       467-47Ø
              SEGUTSEC $       472-475
              SEGAPPC  $       477-492
              SEGAUDIT $       494-5Ø4
              SEGORUID         5Ø6-515
              SEGOEUID         517-526
              SEGOSUID         528-537
              SEGORGID         539-548
              SEGOEGID         55Ø-559
              SEGOSGID         561-57Ø
              SEGNRGID         572-581
              SEGNEGID         583-592
              SEGNSGID         594-6Ø3
              SEGGID           6Ø5-614
               ;
LABEL SEGCLASS = 'Class name'
      SEGUSERN = 'User name'
      SEGUTKNE = 'Utoken encr.?'
      SEGUPRE  = 'Pre-1.9?'
      SEGUVFYX = 'VERIFYX propagation?'
      SEGUNJEU = 'Undefined NJE user?'
      SEGUUAUD = 'UAUDIT?'
      SEGUSPEC = 'RACF special?'
      SEGUDFLT = 'Default token?'
      SEGUUNDF = 'Undefined user?'
      SEGUERR  = 'Token in error?'
      SEGUTRST = 'User trusted?'
      SEGUSEST = 'Session type'
      SEGUSURO = 'Surrogate user?'
      SEGURMT  = 'Remote job?'
      SEGUPRVL = 'Privileged user?'
      SEGUSECL = 'User SECLABEL'
      SEGUEXND = 'Execution node'
      SEGUSUSR = 'Submitting user'
      SEGUSNOD = 'Submitting node'
      SEGUSGRP = 'Submitting group'
      SEGUSPOE = 'Port of entry'
      SEGUSPCL = 'Class of POE'
      SEGUTUSR = 'Userid'
      SEGUTGRP = 'Groupid'
      SEGUTDFT = 'Default group?'
      SEGUTSEC = 'Default SECLABEL?'
      SEGAPPC  = 'APPC key link'
      SEGAUDIT = 'Audit code'
      SEGORUID = 'Old real UID'
      SEGOEUID = 'Old effective UID'
      SEGOSUID = 'Old saved UID'
      SEGORGID = 'Old real GID'
      SEGOEGID = 'Old effective GID'
      SEGOSGID = 'Old saved GID'
      SEGNRGID = 'New real GID'
      SEGNEGID = 'New effective GID'
      SEGNSGID = 'New saved GID'
      SEGGID   = 'GID input parm'
      ;
        OUTPUT RACF.SETEGID;
      END;
    %END;
%MEND SETEGID;
./        ADD   LIST=ALL,NAME=SETEUID
%MACRO SETEUID(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
      %PUT Including variables from SETEUID extension;
      RACF.SETEUID (KEEP=%SMFHDR
                         %SMF8ØHDR(REQ=DEFINE)
```

```
                       SEUCLASS
                       SEUUSERN
                       SEUUTKNE
                       SEUUPRE
                       SEUUVFYX
                       SEUUNJEU
                       SEUUUAUD
                       SEUUSPEC
                       SEUUDFLT
                       SEUUUNDF
                       SEUUERR
                       SEUUTRST
                       SEUUSEST
                       SEUUSURO
                       SEUURMT
                       SEUUPRVL
                       SEUUSECL
                       SEUUEXND
                       SEUUSUSR
                       SEUUSNOD
                       SEUUSGRP
                       SEUUSPOE
                       SEUUSPCL
                       SEUUTUSR
                       SEUUTGRP
                       SEUUTDFT
                       SEUUTSEC
                       SEUAPPC
                       SEUAUDIT
                       SEUORUID
                       SEUOEUID
                       SEUOSUID
                       SEUORGID
                       SEUOEGID
                       SEUOSGID
                       SEUNRUID
                       SEUNEUID
                       SEUNSUID
                       SEUUID
                 )
  %END;
%IF &REQ = EXTRACT %THEN
  %DO;
    %PUT Including datadefinition for SETEUID extension;
    WHEN('SETEUID') DO;
      INPUT %SMF8ØHDR(REQ=EXTRACT)
            SEUCLASS $       282-289
            SEUUSERN $       291-31Ø
            SEUUTKNE $       312-315
            SEUUPRE  $       317-32Ø
            SEUUVFYX $       322-325
```

```
              SEUUNJEU $       327-330
              SEUUUAUD $       332-335
              SEUUSPEC $       337-340
              SEUUDFLT $       342-345
              SEUUUNDF $       347-350
              SEUUERR  $       352-355
              SEUUTRST $       357-360
              SEUUSEST $       362-369
              SEUUSURO $       371-374
              SEUURMT  $       376-379
              SEUUPRVL $       381-384
              SEUUSECL $       386-393
              SEUUEXND $       395-402
              SEUUSUSR $       404-411
              SEUUSNOD $       413-420
              SEUUSGRP $       422-429
              SEUUSPOE $       431-438
              SEUUSPCL $       440-447
              SEUUTUSR $       449-456
              SEUUTGRP $       458-465
              SEUUTDFT $       467-470
              SEUUTSEC $       472-475
              SEUAPPC  $       477-492
              SEUAUDIT $       494-504
              SEUORUID         506-515
              SEUOEUID         517-526
              SEUOSUID         528-537
              SEUORGID         539-548
              SEUOEGID         550-559
              SEUOSGID         561-570
              SEUNRUID         572-581
              SEUNEUID         583-592
              SEUNSUID         594-603
              SEUUID           605-614
               ;
LABEL SEUCLASS = 'Class name'
      SEUUSERN = 'User name'
      SEUUTKNE = 'Utoken encr.?'
      SEUUPRE  = 'Pre-1.9?'
      SEUUVFYX = 'VERIFYX propagation?'
      SEUUNJEU = 'Undefined NJE user?'
      SEUUUAUD = 'UAUDIT?'
      SEUUSPEC = 'RACF special?'
      SEUUDFLT = 'Default token?'
      SEUUUNDF = 'Undefined user?'
      SEUUERR  = 'Token in error?'
      SEUUTRST = 'User trusted?'
      SEUUSEST = 'Session type'
      SEUUSURO = 'Surrogate user?'
      SEUURMT  = 'Remote job?'
      SEUUPRVL = 'Privileged user?'
      SEUUSECL = 'User SECLABEL'
      SEUUEXND = 'Execution node'
      SEUUSUSR = 'Submitting user'
      SEUUSNOD = 'Submitting node'
      SEUUSGRP = 'Submitting group'
      SEUUSPOE = 'Port of entry'
      SEUUSPCL = 'Class of POE'
      SEUUTUSR = 'Userid'
      SEUUTGRP = 'Groupid'
      SEUUTDFT = 'Default group?'
      SEUUTSEC = 'Default SECLABEL?'
      SEUAPPC  = 'APPC key link'
      SEUAUDIT = 'Audit code'
      SEUORUID = 'Old real UID'
      SEUOEUID = 'Old effective UID'
      SEUOSUID = 'Old saved UID'
      SEUORGID = 'Old real GID'
      SEUOEGID = 'Old effective GID'
      SEUOSGID = 'Old saved GID'
      SEUNRUID = 'New real UID'
      SEUNEUID = 'New effective UID'
      SEUNSUID = 'New saved UID'
      SEUUID   = 'UID input parm'
      ;
        OUTPUT RACF.SETEUID;
      END;
    %END;
%MEND SETEUID;
./        ADD   LIST=ALL,NAME=SETGID
%MACRO SETGID(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
      %PUT Including variables from SETGID extension;
      RACF.SETGID (KEEP=%SMFHDR
                        %SMF8ØHDR(REQ=DEFINE)
                        SGICLASS
                        SGIUSERN
                        SGIUTKNE
                        SGIUPRE
                        SGIUVFYX
                        SGIUNJEU
                        SGIUUAUD
                        SGIUSPEC
                        SGIUDFLT
                        SGIUUNDF
                        SGIUERR
                        SGIUTRST
                        SGIUSEST
                        SGIUSURO
                        SGIURMT
```

```
                      SGIUPRVL
                      SGIUSECL
                      SGIUEXND
                      SGIUSUSR
                      SGIUSNOD
                      SGIUSGRP
                      SGIUSPOE
                      SGIUSPCL
                      SGIUTUSR
                      SGIUTGRP
                      SGIUTDFT
                      SGIUTSEC
                      SGIAPPC
                      SGIAUDIT
                      SGIORUID
                      SGIOEUID
                      SGIOSUID
                      SGIORGID
                      SGIOEGID
                      SGIOSGID
                      SGINRGID
                      SGINEGID
                      SGINSGID
                      SGIGID
                 )
  %END;
%IF &REQ = EXTRACT %THEN
  %DO;
    %PUT Including datadefinition for SETGID extension;
    WHEN('SETGID') DO;
      INPUT %SMF80HDR(REQ=EXTRACT)
            SGICLASS $       282-289
            SGIUSERN $       291-310
            SGIUTKNE $       312-315
            SGIUPRE  $       317-320
            SGIUVFYX $       322-325
            SGIUNJEU $       327-330
            SGIUUAUD $       332-335
            SGIUSPEC $       337-340
            SGIUDFLT $       342-345
            SGIUUNDF $       347-350
            SGIUERR  $       352-355
            SGIUTRST $       357-360
            SGIUSEST $       362-369
            SGIUSURO $       371-374
            SGIURMT  $       376-379
            SGIUPRVL $       381-384
            SGIUSECL $       386-393
            SGIUEXND $       395-402
            SGIUSUSR $       404-411
            SGIUSNOD $       413-420
```

```
              SGIUSGRP $       422-429
              SGIUSPOE $       431-438
              SGIUSPCL $       44Ø-447
              SGIUTUSR $       449-456
              SGIUTGRP $       458-465
              SGIUTDFT $       467-47Ø
              SGIUTSEC $       472-475
              SGIAPPC  $       477-492
              SGIAUDIT $       494-5Ø4
              SGIORUID         5Ø6-515
              SGIOEUID         517-526
              SGIOSUID         528-537
              SGIORGID         539-548
              SGIOEGID         55Ø-559
              SGIOSGID         561-57Ø
              SGINRGID         572-581
              SGINEGID         583-592
              SGINSGID         594-6Ø3
              SGIGID           6Ø5-614
               ;
LABEL SGICLASS = 'Class name'
      SGIUSERN = 'User name'
      SGIUTKNE = 'Utoken encr.?'
      SGIUPRE  = 'Pre-1.9?'
      SGIUVFYX = 'VERIFYX propagation?'
      SGIUNJEU = 'Undefined NJE user?'
      SGIUUAUD = 'UAUDIT?'
      SGIUSPEC = 'RACF special?'
      SGIUDFLT = 'Default token?'
      SGIUUNDF = 'Undefined user?'
      SGIUERR  = 'Token in error?'
      SGIUTRST = 'User trusted?'
      SGIUSEST = 'Session type'
      SGIUSURO = 'Surrogate user?'
      SGIURMT  = 'Remote job?'
      SGIUPRVL = 'Privileged user?'
      SGIUSECL = 'User SECLABEL'
      SGIUEXND = 'Execution node'
      SGIUSUSR = 'Submitting user'
      SGIUSNOD = 'Submitting node'
      SGIUSGRP = 'Submitting group'
      SGIUSPOE = 'Port of entry'
      SGIUSPCL = 'Class of POE'
      SGIUTUSR = 'Userid'
      SGIUTGRP = 'Groupid'
      SGIUTDFT = 'Default group?'
      SGIUTSEC = 'Default SECLABEL?'
      SGIAPPC  = 'APPC key link'
      SGIAUDIT = 'Audit code'
      SGIORUID = 'Old real UID'
      SGIOEUID = 'Old effective UID'
      SGIOSUID = 'Old saved UID'
      SGIORGID = 'Old real GID'
      SGIOEGID = 'Old effective GID'
      SGIOSGID = 'Old saved GID'
      SGINRGID = 'New real GID'
      SGINEGID = 'New effective GID'
      SGINSGID = 'New saved GID'
      SGIGID   = 'GID input parm'
      ;
        OUTPUT RACF.SETGID;
      END;
    %END;
%MEND SETGID;
./        ADD   LIST=ALL,NAME=SETUID
%MACRO SETUID(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
      %PUT Including variables from SETUID extension;
      RACF.SETUID (KEEP=%SMFHDR
                        %SMF8ØHDR(REQ=DEFINE)
                        SUICLASS
                        SUIUSERN
                        SUIUTKNE
                        SUIUPRE
                        SUIUVFYX
                        SUIUNJEU
                        SUIUUAUD
                        SUIUSPEC
                        SUIUDFLT
                        SUIUUNDF
                        SUIUERR
                        SUIUTRST
                        SUIUSEST
                        SUIUSURO
                        SUIURMT
                        SUIUPRVL
                        SUIUSECL
                        SUIUEXND
                        SUIUSUSR
                        SUIUSNOD
                        SUIUSGRP
                        SUIUSPOE
                        SUIUSPCL
                        SUIUTUSR
                        SUIUTGRP
                        SUIUTDFT
                        SUIUTSEC
                        SUIAPPC
```

```
                      SUIAUDIT
                      SUIORUID
                      SUIOEUID
                      SUIOSUID
                      SUIORGID
                      SUIOEGID
                      SUIOSGID
                      SUINRUID
                      SUINEUID
                      SUINSUID
                      SUIUID
                 )
  %END;
%IF &REQ = EXTRACT %THEN
  %DO;
    %PUT Including datadefinition for SETUID extension;
    WHEN('SETUID') DO;
      INPUT %SMF80HDR(REQ=EXTRACT)
            SUICLASS $       282-289
            SUIUSERN $       291-310
            SUIUTKNE $       312-315
            SUIUPRE  $       317-320
            SUIUVFYX $       322-325
            SUIUNJEU $       327-330
            SUIUUAUD $       332-335
            SUIUSPEC $       337-340
            SUIUDFLT $       342-345
            SUIUUNDF $       347-350
            SUIUERR  $       352-355
            SUIUTRST $       357-360
            SUIUSEST $       362-369
            SUIUSURO $       371-374
            SUIURMT  $       376-379
            SUIUPRVL $       381-384
            SUIUSECL $       386-393
            SUIUEXND $       395-402
            SUIUSUSR $       404-411
            SUIUSNOD $       413-420
            SUIUSGRP $       422-429
            SUIUSPOE $       431-438
            SUIUSPCL $       440-447
            SUIUTUSR $       449-456
            SUIUTGRP $       458-465
            SUIUTDFT $       467-470
            SUIUTSEC $       472-475
            SUIAPPC  $       477-492
            SUIAUDIT $       494-504
            SUIORUID         506-515
            SUIOEUID         517-526
            SUIOSUID         528-537
```

```
              SUIORGID         539-548
              SUIOEGID         55Ø-559
              SUIOSGID         561-57Ø
              SUINRUID         572-581
              SUINEUID         583-592
              SUINSUID         594-6Ø3
              SUIUID           6Ø5-614
               ;
LABEL SUICLASS = 'Class name'
      SUIUSERN = 'User name'
      SUIUTKNE = 'Utoken encr.?'
      SUIUPRE  = 'Pre-1.9?'
      SUIUVFYX = 'VERIFYX propagation?'
      SUIUNJEU = 'Undefined NJE user?'
      SUIUUAUD = 'UAUDIT?'
      SUIUSPEC = 'RACF special?'
      SUIUDFLT = 'Default token?'
      SUIUUNDF = 'Undefined user?'
      SUIUERR  = 'Token in error?'
      SUIUTRST = 'User trusted?'
      SUIUSEST = 'Session type'
      SUIUSURO = 'Surrogate user?'
      SUIURMT  = 'Remote job?'
      SUIUPRVL = 'Privileged user?'
      SUIUSECL = 'User SECLABEL'
      SUIUEXND = 'Execution node'
      SUIUSUSR = 'Submitting user'
      SUIUSNOD = 'Submitting node'
      SUIUSGRP = 'Submitting group'
      SUIUSPOE = 'Port of entry'
      SUIUSPCL = 'Class of POE'
      SUIUTUSR = 'Userid'
      SUIUTGRP = 'Groupid'
      SUIUTDFT = 'Default group?'
      SUIUTSEC = 'Default SECLABEL?'
      SUIAPPC  = 'APPC key link'
      SUIAUDIT = 'Audit code'
      SUIORUID = 'Old real UID'
      SUIOEUID = 'Old effective UID'
      SUIOSUID = 'Old saved UID'
      SUIORGID = 'Old real GID'
      SUIOEGID = 'Old effective GID'
      SUIOSGID = 'Old saved GID'
      SUINRUID = 'New real UID'
      SUINEUID = 'New effective UID'
      SUINSUID = 'New saved UID'
      SUIUID   = 'UID input parm'
      ;
        OUTPUT RACF.SETUID;
      END;
```

```
    %END;
%MEND SETUID;
./        ADD   LIST=ALL,NAME=SYMLINK
%MACRO SYMLINK(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
      %PUT Including variables from SYMLINK extension;
      RACF.SYMLINK (KEEP=%SMFHDR
                         %SMF8ØHDR(REQ=DEFINE)
                         SYMCLASS
                         SYMUSERN
                         SYMUTKNE
                         SYMUPRE
                         SYMUVFYX
                         SYMUNJEU
                         SYMUUAUD
                         SYMUSPEC
                         SYMUDFLT
                         SYMUUNDF
                         SYMUERR
                         SYMUTRST
                         SYMUSEST
                         SYMUSURO
                         SYMURMT
                         SYMUPRVL
                         SYMUSECL
                         SYMUEXND
                         SYMUSUSR
                         SYMUSNOD
                         SYMUSGRP
                         SYMUSPOE
                         SYMUSPCL
                         SYMUTUSR
                         SYMUTGRP
                         SYMUTDFT
                         SYMUTSEC
                         SYMAPPC
                         SYMAUDIT
                         SYMORUID
                         SYMOEUID
                         SYMOSUID
                         SYMORGID
                         SYMOEGID
                         SYMOSGID
                         SYMPATHN
                         SYMFILID
                         SYMFOUID
                         SYMFOGID
                         SYMSYMLK
```

```
                       SYMFILPL
                       SYMFILSP
                       SYMINODE
                       SYMSCID
                       SYMDCELK
                       SYMAUTYP
                )
  %END;
%IF &REQ = EXTRACT %THEN
  %DO;
    %PUT Including datadefinition for SYMLINK extension;
    WHEN('SYMLINK') DO;
      INPUT %SMF8ØHDR(REQ=EXTRACT)
            SYMCLASS $       282-289
            SYMUSERN $       291-31Ø
            SYMUTKNE $       312-315
            SYMUPRE  $       317-32Ø
            SYMUVFYX $       322-325
            SYMUNJEU $       327-33Ø
            SYMUUAUD $       332-335
            SYMUSPEC $       337-34Ø
            SYMUDFLT $       342-345
            SYMUUNDF $       347-35Ø
            SYMUERR  $       352-355
            SYMUTRST $       357-36Ø
            SYMUSEST $       362-369
            SYMUSURO $       371-374
            SYMURMT  $       376-379
            SYMUPRVL $       381-384
            SYMUSECL $       386-393
            SYMUEXND $       395-4Ø2
            SYMUSUSR $       4Ø4-411
            SYMUSNOD $       413-42Ø
            SYMUSGRP $       422-429
            SYMUSPOE $       431-438
            SYMUSPCL $       44Ø-447
            SYMUTUSR $       449-456
            SYMUTGRP $       458-465
            SYMUTDFT $       467-47Ø
            SYMUTSEC $       472-475
            SYMAPPC  $       477-492
            SYMAUDIT $       494-5Ø4
            SYMORUID         5Ø6-515
            SYMOEUID         517-526
            SYMOSUID         528-537
            SYMORGID         539-548
            SYMOEGID         55Ø-559
            SYMOSGID         561-57Ø
            SYMPATHN $       572-771
            SYMFILID $      1596-1627
```

```
              SYMFOUID        1629-1638
              SYMFOGID        1640-1649
              SYMSYMLK $      1651-1850
              SYMFILPL $      2675-2682
              SYMFILSP $      2684-2691
              SYMINODE        2693-2702
              SYMSCID         2704-2713
              SYMDCELK $      2715-2730
              SYMAUTYP $      2732-2744
               ;
LABEL SYMCLASS = 'Class name'
      SYMUSERN = 'User name'
      SYMUTKNE = 'Utoken encr.?'
      SYMUPRE  = 'Pre-1.9?'
      SYMUVFYX = 'VERIFYX propagation?'
      SYMUNJEU = 'Undefined NJE user?'
      SYMUUAUD = 'UAUDIT?'
      SYMUSPEC = 'RACF special?'
      SYMUDFLT = 'Default token?'
      SYMUUNDF = 'Undefined user?'
      SYMUERR  = 'Token in error?'
      SYMUTRST = 'User trusted?'
      SYMUSEST = 'Session type'
      SYMUSURO = 'Surrogate user?'
      SYMURMT  = 'Remote job?'
      SYMUPRVL = 'Privileged user?'
      SYMUSECL = 'User SECLABEL'
      SYMUEXND = 'Execution node'
      SYMUSUSR = 'Submitting user'
      SYMUSNOD = 'Submitting node'
      SYMUSGRP = 'Submitting group'
      SYMUSPOE = 'Port of entry'
      SYMUSPCL = 'Class of POE'
      SYMUTUSR = 'Userid'
      SYMUTGRP = 'Groupid'
      SYMUTDFT = 'Default group?'
      SYMUTSEC = 'Default SECLABEL?'
      SYMAPPC  = 'APPC key link'
      SYMAUDIT = 'Audit code'
      SYMORUID = 'Old real UID'
      SYMOEUID = 'Old effective UID'
      SYMOSUID = 'Old saved UID'
      SYMORGID = 'Old real GID'
      SYMOEGID = 'Old effective GID'
      SYMOSGID = 'Old saved GID'
      SYMPATHN = 'Path name'
      SYMFILID = 'File id'
      SYMFOUID = 'Owner UID'
      SYMFOGID = 'Owner GID'
      SYMSYMLK = 'SYMLINK data'
      SYMFILPL = 'File pool'
      SYMFILSP = 'File space'
      SYMINODE = 'Inode'
      SYMSCID  = 'File SCID'
      SYMDCELK = 'DCE link'
      SYMAUTYP = 'Request type'
      ;
        OUTPUT RACF.SYMLINK;
      END;
    %END;
%MEND SYMLINK;
./        ADD   LIST=ALL,NAME=UNLINK
%MACRO UNLINK(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
      %PUT Including variables from UNLINK extension;
      RACF.UNLINK (KEEP=%SMFHDR
                        %SMF8ØHDR(REQ=DEFINE)
                        UNLCLASS
                        UNLUSERN
                        UNLUTKNE
                        UNLUPRE
                        UNLUVFYX
                        UNLUNJEU
                        UNLUUAUD
                        UNLUSPEC
                        UNLUDFLT
                        UNLUUNDF
                        UNLUERR
                        UNLUTRST
                        UNLUSEST
                        UNLUSURO
                        UNLURMT
                        UNLUPRVL
                        UNLUSECL
                        UNLUEXND
                        UNLUSUSR
                        UNLUSNOD
                        UNLUSGRP
                        UNLUSPOE
                        UNLUSPCL
                        UNLUTUSR
                        UNLUTGRP
                        UNLUTDFT
                        UNLUTSEC
                        UNLAPPC
                        UNLAUDIT
                        UNLORUID
                        UNLOEUID
```

```
                      UNLOSUID
                      UNLORGID
                      UNLOEGID
                      UNLOSGID
                      UNLPATHN
                      UNLFILID
                      UNLFOUID
                      UNLFOGID
                      UNLLSTDL
                      UNLFILPL
                      UNLFILSP
                      UNLINODE
                      UNLSCID
                      UNLDCELK
                      UNLAUTYP
               )
  %END;
%IF &REQ = EXTRACT %THEN
  %DO;
    %PUT Including datadefinition for UNLINK extension;
    WHEN('UNLINK') DO;
      INPUT %SMF8ØHDR(REQ=EXTRACT)
            UNLCLASS $       282-289
            UNLUSERN $       291-31Ø
            UNLUTKNE $       312-315
            UNLUPRE  $       317-32Ø
            UNLUVFYX $       322-325
            UNLUNJEU $       327-33Ø
            UNLUUAUD $       332-335
            UNLUSPEC $       337-34Ø
            UNLUDFLT $       342-345
            UNLUUNDF $       347-35Ø
            UNLUERR  $       352-355
            UNLUTRST $       357-36Ø
            UNLUSEST $       362-369
            UNLUSURO $       371-374
            UNLURMT  $       376-379
            UNLUPRVL $       381-384
            UNLUSECL $       386-393
            UNLUEXND $       395-4Ø2
            UNLUSUSR $       4Ø4-411
            UNLUSNOD $       413-42Ø
            UNLUSGRP $       422-429
            UNLUSPOE $       431-438
            UNLUSPCL $       44Ø-447
            UNLUTUSR $       449-456
            UNLUTGRP $       458-465
            UNLUTDFT $       467-47Ø
            UNLUTSEC $       472-475
            UNLAPPC  $       477-492
```

```
              UNLAUDIT $        494-5Ø4
              UNLORUID          5Ø6-515
              UNLOEUID          517-526
              UNLOSUID          528-537
              UNLORGID          539-548
              UNLOEGID          55Ø-559
              UNLOSGID          561-57Ø
              UNLPATHN $        572-771
              UNLFILID $       1596-1627
              UNLFOUID         1629-1638
              UNLFOGID         164Ø-1649
              UNLLSTDL $       1651-1654
              UNLFILPL $       1656-1663
              UNLFILSP $       1665-1672
              UNLINODE         1674-1683
              UNLSCID          1685-1694
              UNLDCELK $       1696-1711
              UNLAUTYP $       1713-1725
               ;
LABEL UNLCLASS = 'Class name'
      UNLUSERN = 'User name'
      UNLUTKNE = 'Utoken encr.?'
      UNLUPRE  = 'Pre-1.9?'
      UNLUVFYX = 'VERIFYX propagation?'
      UNLUNJEU = 'Undefined NJE user?'
      UNLUUAUD = 'UAUDIT?'
      UNLUSPEC = 'RACF special?'
      UNLUDFLT = 'Default token?'
      UNLUUNDF = 'Undefined user?'
      UNLUERR  = 'Token in error?'
      UNLUTRST = 'User trusted?'
      UNLUSEST = 'Session type'
      UNLUSURO = 'Surrogate user?'
      UNLURMT  = 'Remote job?'
      UNLUPRVL = 'Privileged user?'
      UNLUSECL = 'User SECLABEL'
      UNLUEXND = 'Execution node'
      UNLUSUSR = 'Submitting user'
      UNLUSNOD = 'Submitting node'
      UNLUSGRP = 'Submitting group'
      UNLUSPOE = 'Port of entry'
      UNLUSPCL = 'Class of POE'
      UNLUTUSR = 'Userid'
      UNLUTGRP = 'Groupid'
      UNLUTDFT = 'Default group?'
      UNLUTSEC = 'Default SECLABEL?'
      UNLAPPC  = 'APPC key link'
      UNLAUDIT = 'Audit code'
      UNLORUID = 'Old real UID'
      UNLOEUID = 'Old effective UID'
```

```
        UNLOSUID = 'Old saved UID'
        UNLORGID = 'Old real GID'
        UNLOEGID = 'Old effective GID'
        UNLOSGID = 'Old saved GID'
        UNLPATHN = 'Path name'
        UNLFILID = 'File id'
```

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

Access the ACEE with REXX

The following REXX function is designed to turn a TSO user's ACEE into usable REXX variables which makes a lot of information about a user available to dialogs, enabling decisions to be made more easily about who gets what. RACF supplies a RACVAR function that goes some way to supplying data, but the function given here is far more information-rich and provides the following REXX variables:

- RACEE is the full ACEE.
- RUSER is the RACF user-id.
- RFLAG1 is the user attributes flag.
- RFLAG2 is the default access flag.
- RDATE is the date of log-on.
- RNAME is the RACF user name.
- RDATA is the user data.
- SECURITY_LABEL is the security label.

To explain these variables I have provided a simple REXX to allow you to check out the code and see the results. First, however, here is the Assembler code for the function:

```
**********************************************************************
*
* ACEETT: A REXX FUNCTION TO PROVIDE ACCESS TO THE ACEE FOR PROVIDING
*         INFORMATION THAT MAY BE USEFUL TO A TSO USER.
*
* USAGE: CALL ACEETT
*
* NOTE:  THE REXX VARIABLES THAT ARE CREATED ARE AS FOLLOWS:
*        SECURITY_LABEL
*        RACEE - THIS IS THE FULL ACEE WHICH IS PROVIDED FOR DEBUGGING
*                PURPOSES.
*        RUSER - THE RACF USER-ID
*        RFLAG1 - THE USER ATTRIBUTES FLAG
*        RFLAG2 - DEFAULT ACCESS FLAG
*        RDATE - DATE OF LOG-ON
*        RNAME - RACF USER NAME
*        RDATA - THE USER DATA
```

```
*
*        NOTE THERE ARE TWO POSSIBLE USER ABENDS THAT THIS CODE MIGHT
*        ISSUE IN THE EVENT OF PROBLEMS. A UØØ1 MEANS THAT THERE WAS A
*        FAILURE CREATING A REXX VARIABLE, WHILE A UØØ2 MEANS THAT THE
*        LOCASCB REQUEST FAILED WHILE TRYING TO FIND THE ACEE.
*
*
***********************************************************************
              MACRO
              SHOW &LABEL,&ASNAME,&ERR=ABENDØØ1,&LEN=,&UNPACK=NO
***********************************************************************
*
* MACRO FORMAT:
*       SHOW &LABEL,&ASNAME,&ERR=,&LEN=,&UNPACK=
* WHERE:
*       &LABEL IS THE NAME OF THE LABEL WHICH ADDRESSES THE FIELD FROM
*              WHERE THE DATA TO BE DEFINED IN A REXX VARIABLE IS
*              LOCATED
*       &ASNAME IS THE NAME TO BE ASSIGNED TO THE DATA FOR USE IN REXX
*       &ERR= IS THE LABEL TO BRANCH TO SHOULD AN ERROR OCCUR WHILE
*             CREATING THE REXX VARIABLE. BY DEFAULT IT IS ABENDØØ1
*       &LEN= IF THE DATA AT &LABEL IS NOT DEFINED SUCH THAT THE LENGTH
*             OF THE DATA IS WHAT YOU WANT, SIMPLY ENTER A NUMBER HERE
*             THAT DEFINES THE LENGTH REQUIRED. CAN ALSO BE USEFUL IF
*             NECESSARY TO DUMP OUT A LARGE AREA.
*       &UNPACK= IF THE DATA IS IN PACKED FORMAT, SET THIS TO YES IF
*             YOU WANT THE AREA UNPACKED FOR YOU. THE DEFAULT IS NO.
*
**********************************************************************
        AIF (D’SHOW_START).NONEED
        B  BY_SHOW_START
SHOW_START DS ØH
        ST R1Ø,COMRET
        LA 6,COMSHVB
        USING SHVBLOCK,R6
        XC COMSHVB(SHVBLEN),COMSHVB
        XC SHVNEXT,SHVNEXT
        MVI SHVCODE,C’S’
        BR 14
BY_SHOW_START DS ØH
LITLOC  LOCTR
@_UNPACK DC CL16' ‘
&SYSECT LOCTR
.NONEED ANOP
        BAL 14,SHOW_START
LITLOC  LOCTR
&LABCHECK SETC ‘@_&ASNAME’
         AIF   (D’&LABCHECK).BYPASS
@_&ASNAME DC C’&ASNAME’
.BYPASS ANOP
```

```
&SYSECT LOCTR
         LA 1,@_&ASNAME
         ST 1,SHVNAMA
         LA 1,L’@_&ASNAME
         ST 1,SHVNAML
         AIF (‘&UNPACK’ EQ ‘NO’).DATAAOK
         UNPK @_UNPACK,&LABEL
         OI @_UNPACK+(L’@_UNPACK-1),X’FØ'
         LA 1,@_UNPACK
         ST 1,SHVVALA
         LA 1,L’@_UNPACK
         AGO .OK
.DATAAOK ANOP
         LA 1,&LABEL
         ST 1,SHVVALA
         AIF (T’&LEN NE ‘O’).DOLEN
         LA 1,L’&LABEL
         AGO .OK
.DOLEN  ANOP
         LA 1,&LEN
.OK     ANOP
        ST 1,SHVVALL
        LR Ø,1Ø
        LA 1,COMS
        L 15,IRXEXCOM
        BALR 14,15
        LTR 15,15
        BNZ &ERR
        MEND
ACEETT TITLE ‘REXX FUNCTION TO RETRIEVE ACEE INFORMATION’
ACEETT AMODE 31
ACEETT RMODE ANY
ACEETT CSECT
RØ EQU Ø
R1 EQU 1
R2 EQU 2
R3 EQU 3
R4 EQU 4
R5 EQU 5
R6 EQU 6
R7 EQU 7
R8 EQU 8
R9 EQU 9
R1Ø EQU 1Ø
R11 EQU 11
R12 EQU 12
R13 EQU 13
R14 EQU 14
R15 EQU 15
         PRINT GEN
         BAKR 14,Ø
         LR   12,15
         USING ACEETT,12
         PRINT GEN
         LR    R1Ø,RØ                  *R1Ø -> A(ENVIRONMENT BLOCK)
         USING ENVBLOCK,R1Ø
         LR    R11,R1                  *R11 -> A(PARAM LIST (EFPL))
         USING EFPL,R11
         L     R9,ENVBLOCK_IRXEXTE     *R9 -> A(EXTERNAL EP TABLE)
         USING IRXEXTE,R9
*
* GET A WORK AREA FOR REXX OUTPUT AND MAP WITH R2
*
         STORAGE OBTAIN,LENGTH=AREALEN,ADDR=(2),LOC=BELOW
*
         USING WORKAREA,2
*
* PREPARE THE REXX AREA FOR USE
*
         XC  COMS(COMSLEN),COMS * SET TO LOW VALUES
         LA  15,COMID
         ST  15,COMS
         LA  15,COMDUMMY
         ST  15,COMS+4
         ST  15,COMS+8
         LA  15,COMSHVB
         ST  15,COMS+12
         LA  15,COMRET
         ST  15,COMS+16
         OI  COMS+16,X'8Ø'
         MVC COMID,=C'IRXEXCOM'
*
VARLOOP DS ØH
*
***********************************************************************
*        LOCATE THE ACEE
***********************************************************************
*
         EXTRACT (5),FIELDS=(ASID)
*
         L 5,Ø(,5)
         STH 5,MYASID
*
         LOCASCB ASID=MYASID
*
         LTR 15,15
         BNZ ABENDØØ2
         USING ASCB,R1
         L  R5,ASCBASXB
         USING ASXB,R5
         L  R4,ASXBSENV
         USING ACEE,R4
         DROP R5
         L  R5,ACEETOKP
         USING TOKEN,R5
         SHOW TOKSCL,SECURITY_LABEL
         SHOW ACEE,RACEE,LEN=168
         SHOW ACEEUSRI,RUSER
         SHOW ACEEFLG1,RFLAG1
         SHOW ACEEFLG2,RFLAG2
         SHOW ACEEDATE,RDATE,UNPACK=YES
         L    R3,ACEEUNAM
         XR   R5,R5
         ICM  R5,1,Ø(R3)
         LA   R3,1(,R3)      * POINT TO ACTUAL DATA
         USING WORK,R3
         BCTR R5,Ø           * DECREMENT R5 FOR LENGTH
         SHOW  NAME,RNAME,LEN=Ø(5)
*
* NOW CHECK FOR USERDATA
*
         L    R3,ACEEINST    * POINT AT INSTALLATION DATA
         LTR  R3,R3          * IF ZERO, THEN NONE
         BZ   RETURN
         XR   R5,R5
         ICM  R5,1,Ø(R3)     * GET LENGTH OF USERDATA
         LA   R3,1(,R3)      * POINT TO ACTUAL DATA
         USING WORK,R3
         BCTR R5,Ø           * DECREMENT R5 FOR LENGTH
*
         SHOW  USEDATA,RDATA,LEN=Ø(5)
*
         B RETURN
         EJECT
ABENDØØ1 DS    ØH
         ABEND 1
ABENDØØ2 DS    ØH
         ABEND 2
*
***********************************************************************
***      RETURN TO CALLER                                           ***
***      RELEASING ALL STORAGE IN THE PROCESS                       ***
***********************************************************************
*
RETURN   DS ØH
         STORAGE RELEASE,LENGTH=AREALEN,ADDR=(2)
         PR
*
***********************************************************************
***      WORKING STORAGE ETC                                        ***
```

```
***********************************************************************
*
         TITLE  ‘WORKING STORAGE / DSECTS’
         LTORG
*
WORKAREA DSECT
*
*      IRXEXCOM PARAMETER AREA
*
         DS  ØD
COMS     DS  5AL4
COMID    DS  CL8
COMDUMMY DS  AL4                * NOT USED
COMSHVB  DS  (SHVBLEN)X         * IRXEXCOM SHVBLOCK (LENGTH FROM DSECT)
COMRET   DS  AL4                * IRXECOM RC
         DS  ØD
MYASID   DS  H
AREALEN  EQU *-WORKAREA
COMSLEN  EQU *-COMS
         CVT DSECT=YES
         IHALDA
         IHAASCB
         IHAASXB
         IHAACEE
         ICHRUTKN
         IRXEFPL
         IRXARGTB
         IRXEVALB
         IRXENVB
         IRXEXTE
         IRXSHVB
WORK     DSECT
NAME     DS  CL25Ø
         ORG *-25Ø
USEDATA  DS  CL25Ø
         END
```

SAMPLE REXX

```
/* REXX */
CALL ACEETT
SAY SECURITY_LABEL
SAY RUSER
SAY RDATE
IF BITAND(RFLAG2,'8Ø'X) ¬= 'ØØ'X THEN SAY 'DEFAULT ALTER'
IF BITAND(RFLAG2,'4Ø'X) ¬= 'ØØ'X THEN SAY 'DEFAULT CONTROL'
IF BITAND(RFLAG2,'2Ø'X) ¬= 'ØØ'X THEN SAY 'DEFAULT UPDATE'
IF BITAND(RFLAG2,'1Ø'X) ¬= 'ØØ'X THEN SAY 'DEFAULT READ'
IF BITAND(RFLAG2,'Ø1'X) ¬= 'ØØ'X THEN SAY 'DEFAULT NONE'
```

```
SAY C2X(RACEE)
IF BITAND(RFLAG1,'8Ø'X) ¬= 'ØØ'X THEN SAY 'SPECIAL ATTRIBUTE'
IF BITAND(RFLAG1,'4Ø'X) ¬= 'ØØ'X THEN SAY 'AUTOMATIC DATA SECURITY'
IF BITAND(RFLAG1,'2Ø'X) ¬= 'ØØ'X THEN SAY 'OPERATIONS ATTRIBUTE'
IF BITAND(RFLAG1,'1Ø'X) ¬= 'ØØ'X THEN SAY 'AUDITOR ATTRIBUTE'
IF BITAND(RFLAG1,'Ø8'X) ¬= 'ØØ'X THEN SAY 'LOG MOST RACF FUNCTIONS'
IF BITAND(RFLAG1,'Ø1'X) ¬= 'ØØ'X THEN SAY 'RACF DEFINED'
SAY RNAME
SAY RDATA
```

Cross-checking RACF/MVS definitions

Certain MVS and RACF definitions normally have an interdependency and the existence of such resources should be coordinated. Such resources are:

- Catalog alias.
- RACF dataset profile.
- RACF user.
- RACF group.
- RACF TSO segment.
- SYSL.UADs TSO definition.

In a clean and neat installation, the resources will exist in a certain combination. Although individual variations in different MVS systems can apply, the following rules would seem to be a reasonable approach. It is of course assumed that ‘Protect all’ is in effect:

- There should be at least one corresponding RACF dataset profile for every alias.
- There should be either a RACF user or a group definition for every alias.

- There should be either a RACF user or a group definition for datasets without an alias (catalogued in the master catalog).
- There should be a covering dataset profile for datasets without an alias (catalogued in the master catalog).
- Preferably, there should be corresponding datasets to an alias, although not of course in all situations.
- A TSO user should not be defined in both RACF TSO segment and SYSL.UADs.
- A RACF group should not be defined as a TSO user in SYSL.UADS.
- There should be a corresponding catalog alias for every TSO user.
- There should be a corresponding dataset profile for every TSO user.
- Preferably, there should be datasets for every TSO user.
- For every dataset profile, there should be a corresponding alias if the dataset is not allowed to be catalogued in the master catalog.
- There should be a corresponding RACF user or group for every dataset profile.
- Preferably, there should be corresponding datasets for every dataset profile, although not in all situations.
- Preferably, there should be a corresponding catalog alias for every RACF group (in case datasets for the group exist).
- Preferably, there should be a corresponding dataset profile for every RACF group (in case datasets for the group exist).
- Users should not be defined in SYSL.UADs if the installation standard is to define TSO users in the TSO segment (except for emergency purposes).
- RACF users without a corresponding TSO user should not have an alias (when datasets are not expected).
- RACF users without a corresponding TSO user should not have a dataset profile (when datasets are not expected).

- RACF users without a corresponding TSO user should not have datasets (when datasets are not expected).

In my experience, mismatches are common and are easily introduced. This can happen when TSO users are defined or removed using manual routines.

To keep an eye on the situation, and to check for consistency in the definitions concerned, checks can be made based upon the various points of view of the five different resource types:

- Catalog alias
- RACF dataset profile
- RACF user
- RACF group
- TSO user.

For this purpose, a set of routines has been implemented. The routines must run under a user-id that has RACF special and auditor attributes, and preferably operations. All of the routines are designed using the same basic approach with similar coding.

The routines will write a log of possible mismatches of minor importance, as well of those considered to be severe mismatches. In the case of severe mismatches, the routine will send a mail to selected receivers, although the mailing function itself is not described here.

In this installation, only ‘SYSn’ (‘SYS1’ etc) datasets are allowed in the master catalog and the routines perform no special checks on the contents of the master catalog versus RACF definitions.

FROM AN ALIAS POINT OF VIEW

The first routine looks at the definitions from an alias point of view:

```
//STEP1     EXEC PGM=IEBGENER  GENERATE CLIST
//SYSIN     DD   DUMMY
//SYSPRINT  DD   SYSOUT=*
//SYSUT2    DD   DSN=&&TEMP(CLIST),DISP=(,PASS),SPACE=(TRK,(1,1,1)),
// UNIT=VIO
//SYSUT1   DD   *
  /*
  /*  CHECK THAT ALIASES FROM MASTER CATALOG HAS APPROPRIATE
  /*  RACF/UADS/DATASET DEFINITIONS
  /*
  /*  Parameters:
  /*   RC1 - RCn: mail receivers
  /*   GENLLQ   : Generic qualifier; * for non-enhanced generic naming,
  /*              ** for enhanced generic naming.
  /*
PROC Ø ID('ALIAS RACF CHECK') DEBUG(NEBUG) RC1(MAILUS1) RC2() +
RC3() RC4() RC5() RC6() RC7() RC8() GENLLQ(*)
CONTROL   MSG NOFLUSH NOLIST NOCONLIST NOSYMLIST
ATTN DO                     /* attention routine */
 SET &FLUSH = FLUSH         /* NEXT STATEMENT MUST BE NULL LINE     */

END
ERROR DO                    /* error routine */
 SET &RET = &LASTCC
 RETURN
END
IF &FLUSH = FLUSH THEN DO   /* is attention set */
 EXIT CODE(Ø)
END
IF &STR(&DEBUG) = DEBUG THEN DO
 CONTROL MSG NOFLUSH LIST CONLIST SYMLIST
END
OPENFILE OUT OUTPUT
SET OUT =  STATUS FROM AN ALIAS POINT OF VIEW
PUTFILE OUT
IF &SYSPREF = &STR() THEN DO
 PROFILE PREFIX(INSTPREF)           /* Default prefix */
END
ELSE DO
 PROFILE PREFIX(&SYSUID)
END
SET &TSTAMP = +
&STR(T)&SUBSTR(1:2,&SYSTIME)&SUBSTR(4:5,&SYSTIME)&SUBSTR(7:8,&SYSTIME)
DEL '&SYSPREF..&TSTAMP..TEMP.LIST'
SET &CNT = Ø
SET &RET = Ø
  /* allocate using SMS parameters storclas and mgmtclas;
     if not SMS-managed use unit instead
ALLOC FI(LIST) DA('&SYSPREF..&TSTAMP..TEMP.LIST') +
NEW SPACE(1 1) CYLINDERS +
STORCLAS(TEMP) MGMTCLAS(TEMP) RECFM(F B) LRECL(8Ø) BLKSIZE(2792Ø) REUSE
DO WHILE &RET NE Ø AND &CNT < 3ØØ THEN DO
 SET &TSTAMP = +
 &STR(T)&SUBSTR(1:2,&SYSTIME)&SUBSTR(4:5,&SYSTIME)&SUBSTR(7:8,&SYSTIME)
 DEL '&SYSPREF..&TSTAMP..TEMP.LIST'
 SET &RET = Ø
 ALLOC FI(LIST) DA('&SYSPREF..&TSTAMP..TEMP.LIST') +
 NEW SPACE(1 1) CYLINDERS +
 STORCLAS(TEMP) MGMTCLAS(TEMP) RECFM(F B) LRECL(8Ø) BLKSIZE(2792Ø) REUSE
 IF &RET NE Ø THEN DO
  FREE DA('&SYSPREF..&TSTAMP..TEMP.LIST')
 END
 SET &CNT = &CNT + 1
    /* wait utility waiting indicated no of secs */
 SLEEP 15
END
SET &PREFIX = &SYSPREF
OPENFILE LIST OUTPUT
SET LIST =  STATUS FROM AN ALIAS POINT OF VIEW
PUTFILE LIST
SET &RP = &STR())
PROFILE NOPREFIX
SET &SYSOUTTRAP = 999999
LISTC ALIAS
SET &SYSOUTTRAP = Ø
SET &MAXLINE = &SYSOUTLINE
SET &ALIAS = ALIAS
SET &P = Ø
SET &RET = Ø
DO WHILE &P < &MAXLINE
 SET &P = &P + 1
 SET &SYSDVAL = &STR(&&SYSOUTLINE&P)
 SET &SYSDVAL = &STR(&SYSDVAL)
 READDVAL &A1 &A2 &A3 &A4 &A5 &A6 &A7 &A8  /* split into tokens */
 SET &&ALIAS&P = &STR(&A3)      /* double substitition */
END
SET &P = Ø
SET &RET = Ø
DO WHILE &P < &MAXLINE
 SET &P = &P + 1
 SET &A3 = &STR(&&ALIAS&P)
 SET &PREF = &STR()
 IF &LENGTH(&STR(&A3)) = 4 THEN DO
  SET &PREF = &SUBSTR(1:3,&STR(&A3))
 END
 IF &STR(&A3) NE &STR() AND &PREF NE SYS AND +
 &LENGTH(&STR(&A3)) < 9 THEN DO
  WRITE
  WRITE &STR(========>) INFORMATION FOR HLQ &STR("&A3")
  SET &HLQ = NO
  SET &DS = NO
  SET &RACFU = NO
  SET &RACFT = NO
  SET &GROUP = NO
  SET &DSP = NO
  SET &UADS = NO
  SET &SYSOUTTRAP = 99999
  SET &RET = Ø
  LISTC ALL ENT('&A3'&RP
  SET &SYSOUTTRAP = Ø
  IF &RET = Ø THEN DO
   SET &HLQ = YES
   SET &SYSOUTTRAP = 99999
   SET &RET = Ø
   LISTC     LEVEL(&A3&RP
   SET &SYSOUTTRAP = Ø
   IF &RET = Ø THEN DO
    SET &DS = YES
    WRITE SOME DATASETS/ENTRIES DO EXIST UNDER ALIAS &STR("&A3")
   END
   ELSE DO
    WRITE NO DATASETS/ENTRIES EXIST UNDER ALIAS &STR("&A3")
   END
  END
  ELSE DO
   WRITE NO ALIAS EXISTS FOR &STR("&A3") ANY LONGER
  END
  SET &RET = Ø
  LG &A3
  IF &RET = Ø THEN DO
   SET &GROUP = YES
  END
  ELSE DO
   WRITE NO RACF GROUP EXISTS FOR &STR("&A3")
   CONTROL NOMSG
   SET &SYSOUTTRAP = 999999
   SET &RET = Ø
   LU &A3
   SET &SYSOUTTRAP = Ø
   SET &LURET = &RET
   CONTROL MSG         /* must use control msg to list TSO segment */
   IF &LURET < 4 THEN DO
    SET &RACFU = YES
    WRITE "&A3" USER EXISTS IN RACF
    SET &SYSOUTTRAP = 999999
    LU &A3 TSO NORACF
    SET &SYSOUTTRAP = Ø
    SET &MAXTSO = &SYSOUTLINE
    SET &RACFTSO = &STR()
    SET &N = Ø
    SET &RET = Ø
    DO WHILE &N < &MAXTSO
```

```
   SET &N = &N + 1
   SET &SYSDVAL = &STR(&&SYSOUTLINE&N)
   SET &SYSDVAL = &STR(&SYSDVAL)
   READDVAL &B1 &B2 &B3 &B4 &B5 &B6 &B7 &B8
   IF &STR(&B1) = NO AND &STR(&B2) = TSO AND &STR(&B3) = INFORMATION +
   THEN DO
    WRITE "&A3" IS NOT DEFINED AS TSO USER IN RACF
    SET &RACFTSO = NO
   END
  END
  IF &RACFTSO NE NO AND &RACFTSO NE &STR() THEN DO
   SET &RACFT = YES
   WRITE "&A3" IS DEFINED AS TSO USER IN RACF
  END
 END
 ELSE DO
  WRITE "&A3" DOES NOT EXIST AS USER IN RACF
 END
END
SET &RET = Ø
SET &MEMBER = &A3
SET &MEMBER = &MEMBER.&STR(.)
SEARCH ALL CLASS(DATASET) MASK(&MEMBER) LIST
IF &RET > Ø THEN DO
 WRITE NO RACF DATASET PROFILES EXIST FOR &STR("&A3")
END
ELSE DO
 SET &DSP = YES
 SET &RET = Ø
 SET &SYSOUTTRAP = 999999
 LD DATASET('&A3...&GENLLQ')  /* use 3 dots due to double subst. */
 SET &SYSOUTTRAP = Ø
 IF &RET > Ø THEN DO
  /* SHOW THAT ALIAS LACKS THE BASE DATASET PROFILE */
  WRITE ERROR: NO BASE PROFILE &A3...&GENLLQ PROVIDED FOR HLQ &A3
  SET &OUT = +
  &STR(ERROR: NO BASE PROFILE &A3...&GENLLQ PROVIDED FOR HLQ &A3)
  PUTFILE OUT
  SET &LIST = &STR(&OUT)
  PUTFILE LIST
    /* Write to programmer program */
  WTP &STR(&OUT)
  SET &ERROR = YES
 END
END
SET &MEMBER = &A3
SET &MEMBER = &MEMBER.Ø
IF &LENGTH(&STR(&A3)) < 8 THEN DO
 SET &UADS = &STR(&SYSDSN('SYS1.UADS(&MEMBER)'))
 WRITE TSO DEFINITION IN SYS1.UADS = &UADS FOR &STR("&A3")
 IF &STR(&UADS) = OK THEN DO
  SET &UADS = YES
 END
END
ELSE DO
 WRITE TSO DEFINITION IN SYS1.UADS = MEMBER NOT FOUND FOR &STR("&A3")
END
IF &HLQ NE YES THEN DO
 WRITE ===> LOGICAL ERROR: HLQ &STR("&A3") MISSING
 SET &OUT = &STR(LOGICAL ERROR: HLQ &STR("&A3") MISSING)
 PUTFILE OUT
 SET &LIST = &STR(&OUT)
 PUTFILE LIST
 WTP &STR(&OUT)
 SET &ERROR = YES
END
IF &HLQ = YES THEN DO
 IF &DS NE YES AND &GROUP NE YES THEN DO
  WRITE ===> POSSIBLE ERROR: NO DATASET FOR USER HLQ &STR("&A3")
  SET &OUT = &STR(POSSIBLE ERROR: NO DATASET FOR USER HLQ &STR("&A3"))
  PUTFILE OUT
 END
 IF &DS NE YES AND &GROUP = YES THEN DO
/*allowed prefixes for quasi temporary datasets in this installation*/
  IF &STR(&A3) NE XMIT AND &STR(&A3) NE TEMP AND &STR(&A3) NE WORK +
  THEN DO
   WRITE ===> ERROR: NO DATASET FOR GROUP HLQ &STR("&A3")
   SET &OUT = &STR(ERROR: NO DATASET FOR GROUP HLQ &STR("&A3"))
   PUTFILE OUT
   SET &LIST = &STR(&OUT)
   PUTFILE LIST
   WTP &STR(&OUT)
   SET &ERROR = YES
  END
 END
 IF &DSP NE YES THEN DO
  WRITE ===> ERROR: NO DATASET RACF PROFILE FOR HLQ &STR("&A3")
  SET &OUT = &STR(ERROR: NO DATASET RACF PROFILE FOR HLQ &STR("&A3"))
  PUTFILE OUT
  SET &LIST = &STR(&OUT)
  PUTFILE LIST
  WTP &STR(&OUT)
  SET &ERROR = YES
  IF &DS = YES THEN DO
   WRITE ===>        BUT DATASETS EXIST.
   SET &OUT = &STR(        BUT DATASETS EXIST.)
   PUTFILE OUT
   SET &LIST = &STR(&OUT)
   PUTFILE LIST
```

```
     WTP &STR(&OUT)
    END
   END
   IF &RACFT = YES AND &UADS = YES THEN DO
    WRITE ===> ERROR: &STR("&A3") TSO USER BOTH IN RACF AND UADS
    SET &OUT = &STR(ERROR: &STR("&A3") TSO USER BOTH IN RACF AND UADS)
    PUTFILE OUT
    SET &LIST = &STR(&OUT)
    PUTFILE LIST
    WTP &STR(&OUT)
    SET &ERROR = YES
   END
   IF &RACFU NE YES AND &GROUP NE YES THEN DO
    WRITE ===> ERROR: &STR("&A3") NEITHER RACF USER NOR GROUP
    SET &OUT = &STR(ERROR: &STR("&A3") NEITHER RACF USER NOR GROUP)
    PUTFILE OUT
    SET &LIST = &STR(&OUT)
    PUTFILE LIST
    WTP &STR(&OUT)
    SET &ERROR = YES
   END
   IF &UADS = YES AND &GROUP = YES THEN DO
    WRITE ===> ERROR: &STR("&A3") BOTH IN UADS AND RACF GROUP
    SET &OUT = &STR(ERROR: &STR("&A3") BOTH IN UADS AND RACF GROUP)
    PUTFILE OUT
    SET &LIST = &STR(&OUT)
    PUTFILE LIST
    WTP &STR(&OUT)
    SET &ERROR = YES
   END
   IF &UADS = YES AND &RACFU NE YES THEN DO
    WRITE ===> ERROR: &STR("&A3") IN UADS BUT NOT A RACF NONTSO USER
    SET &OUT = &STR(ERROR: &STR("&A3") IN UADS BUT NOT A RACF +
    NONTSO USER.)
    PUTFILE OUT
    SET &LIST = &STR(&OUT)
    PUTFILE LIST
    WTP &STR(&OUT)
    SET &ERROR = YES
   END
  END
 END
END
CLOSFILE OUT
CLOSFILE LIST
IF &ERROR = YES THEN DO
 SET &RC = RC
 SET &N = 1
 SET &RC = &&RC&N
 DO WHILE &RC NE &STR() AND &N < 9
   /* mail routine not described here */
  %MAILSENS ID('&ID') RC('&RC') DS(&PREFIX..&TSTAMP..TEMP.LIST) +
  DEBUG(&DEBUG) RCDOMAIN(YES) /* INFORM ERROR STATUS VIA MAIL */
  SET &N = &N + 1
  SET &RC = &&RC&N
 END
END
FREE DA('&PREFIX..&TSTAMP..TEMP.LIST')
DEL '&PREFIX..&TSTAMP..TEMP.LIST'
  /*
//STEP2    EXEC PGM=IKJEFTØ1,DYNAMNBR=128,  EXECUTE CLIST
// PARM='%CLIST DEBUG(NEBUG)'
//SYSTSPRT DD   SYSOUT=*
//SYSPRINT DD   SYSOUT=*
//OUT      DD   SYSOUT=*
//SYSPROC  DD   DSN=&TEMP,DISP=(OLD,PASS)
//         DD   DSN=SYSTEM.CLIST,DISP=SHR
//SYSTSIN  DD   DUMMY
//SYSIN    DD   DUMMY
//*
```

FROM A RACF DATASET PROFILE POINT OF VIEW

This routine looks at the definitions from the point of view of the RACF dataset profile:

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

END
ERROR DO
 SET &RET = &LASTCC
 RETURN
END
IF &FLUSH = FLUSH THEN DO
 EXIT CODE(0)
END
IF &STR(&DEBUG) = DEBUG THEN DO
 CONTROL MSG NOFLUSH LIST CONLIST SYMLIST
END
OPENFILE OUT OUTPUT
SET OUT =  STATUS FROM AN RACF DATASET PROFILE POINT OF VIEW
PUTFILE OUT
IF &SYSPREF.. = . THEN DO
 PROFILE PREFIX(INSTPREF)           /* Default prefix */
END
ELSE DO
 PROFILE PREFIX(&SYSUID)
END
SET &TSTAMP = +
&STR(T)&SUBSTR(1:2,&SYSTIME)&SUBSTR(4:5,&SYSTIME)&SUBSTR(7:8,&SYSTIME)
DEL '&SYSPREF..&TSTAMP..TEMP.LIST'
SET &CNT = 0
SET &RET = 0
ALLOC FI(LIST) DA('&SYSPREF..&TSTAMP..TEMP.LIST') +
NEW SPACE(1 1) CYLINDERS +
STORCLAS(TEMP) MGMTCLAS(TEMP) RECFM(F B) LRECL(80) BLKSIZE(27920) REUSE
DO WHILE &RET NE 0 AND &CNT < 300 THEN DO
 SET &TSTAMP = +
 &STR(T)&SUBSTR(1:2,&SYSTIME)&SUBSTR(4:5,&SYSTIME)&SUBSTR(7:8,&SYSTIME)
 DEL '&SYSPREF..&TSTAMP..TEMP.LIST'
 SET &RET = 0
 ALLOC FI(LIST) DA('&SYSPREF..&TSTAMP..TEMP.LIST') +
 NEW SPACE(1 1) CYLINDERS +
 STORCLAS(TEMP) MGMTCLAS(TEMP) RECFM(F B) LRECL(80) BLKSIZE(27920) REUSE
 IF &RET NE 0 THEN DO
  FREE DA('&SYSPREF..&TSTAMP..TEMP.LIST')
 END
 SET &CNT = &CNT + 1
 SLEEP 15
END
SET &SPREFIX = &SYSPREF
OPENFILE LIST OUTPUT
SET LIST =  STATUS FROM AN RACF DATASET PROFILE POINT OF VIEW
PUTFILE LIST
SET &RP = &STR())
PROFILE NOPREFIX
SET &SYSOUTTRAP = 999999
SEARCH ALL CLASS(DATASET) NOMASK LIST
SET &SYSOUTTRAP = 0
SET &MAXLINE = &SYSOUTLINE
SET &ALIAS = ALIAS
SET &DSN = DSN
SET &P = 0
SET &RET = 0
DO WHILE &P < &MAXLINE
 SET &P = &P + 1
 SET &SYSDVAL = &STR(&&SYSOUTLINE&P)
 SET &SYSDVAL = &STR(&SYSDVAL)
 READDVAL &A1 &A2 &A3 &A4 &A5 &A6 &A7 &A8
 SET &DSNAME = &STR(&A1)
  /*
  /*  SCAN FOR QUALIFIERS
  /*
 SET &PREFIX = &STR(&DSNAME)
 SET &SUFFIX = &STR(&DSNAME)
 SET &PRESEC = &STR(&DSNAME)
 SET &PRESUF = &STR(&DSNAME)
 SET &SECQUAL = &STR()
 SET PERIOD = 0
 SET &LENDSN = &LENGTH(&STR(&DSNAME))
 IF &LENDSN > 0 THEN DO
  SET &NQ = 0
  SET &QUAL = QUAL
  DO WHILE (&NQ < &EVAL((44/2)+1))
   SET &NQ = &NQ + 1
   SET &&QUAL&NQ = &STR()
  END
  SET &SRCHCHAR = &STR(.)
  SET &S = &LENDSN
  SET &STARTLOC = 1
  SET &QUAL = QUAL
  DO WHILE &S > 0
   SET &LOC = &SYSINDEX(&STR(&SRCHCHAR),&STR(&DSNAME),&STARTLOC)
   IF &LOC = 0 THEN DO
    SET &QUALIFIER = +
    &SUBSTR(&STARTLOC:&LENGTH(&STR(&DSNAME)),&STR(&DSNAME))
    SET &S = 0
   END
   ELSE DO
    SET &PERIOD = &PERIOD + 1
    SET &QUALLEN = &LOC - &STARTLOC
    SET &ENDLOC = &LOC - &LENGTH(&STR(&SRCHCHAR))
    SET &QUALIFIER = &SUBSTR(&STARTLOC:&ENDLOC,&STR(&DSNAME))
    SET &PREVQUAL = &STR(&QUALIFIER)
    SET &STARTLOC = &LOC + &LENGTH(&STR(&SRCHCHAR))
    SET &S = &S - &QUALLEN - &LENGTH(&STR(&SRCHCHAR))
    SET &&QUAL&PERIOD = &STR(&QUALIFIER)
    IF &PERIOD = 1 THEN DO
     SET &PREFIX = &STR(&QUALIFIER)
    END
    IF &PERIOD = 2 THEN DO
     SET &PRESEC = &STR(&PREFIX&STR(&SRCHCHAR)&QUALIFIER)
     SET &SECQUAL = &STR(&QUALIFIER)
    END
   END
  END
  SET &SUFFIX = &STR(&QUALIFIER)
  IF &PERIOD > Ø THEN DO
   SET &PRESUF = &STR(&PREVQUAL&STR(&SRCHCHAR)&QUALIFIER)
  END
  IF &PERIOD = 1 THEN DO
   SET &SECQUAL = &STR(&QUALIFIER)
  END
  SET &NOQUAL = &PERIOD + 1
  SET &&QUAL&NOQUAL = &STR(&QUALIFIER)
 END
  /*
 SET &&ALIAS&P = &STR(&PREFIX)
 SET &&DSN&P = &STR(&DSNAME)
END
SET &P = Ø
SET &RET = Ø
DO WHILE &P < &MAXLINE
 SET &P = &P + 1
 SET &A1 = &STR(&&ALIAS&P)
 SET &PREF = &STR()
 IF &LENGTH(&STR(&A1)) = 4 THEN DO
  SET &PREF = &SUBSTR(1:3,&STR(&A1))
 END
 IF &STR(&A1) NE &STR() AND &PREF NE SYS AND +
 &STR(&A1) NE &STR(&LASTPF) AND +
 &LENGTH(&STR(&A1)) < 9 THEN DO
  SET &LASTPF = &STR(&A1)
  SET &DSNAME = &STR(&&DSN&P)
  WRITE
  WRITE &STR(========>) INFORMATION FOR DSP &STR("&DSNAME")
  LD DATASET('&DSNAME') DSNS NORACF GENERIC  /* SHOW COVERED DATASETS */
  WRITE
  SET &HLQ = NO
  SET &DS = NO
  SET &RACFU = NO
  SET &RACFT = NO
  SET &GROUP = NO
  SET &DSP = NO
  SET &UADS = NO
  SET &SYSOUTTRAP = 99999
  SET &RET = Ø
  LISTC ALL ENT('&A1'&RP
  SET &SYSOUTTRAP = Ø
```

```
IF &RET = Ø THEN DO
 SET &HLQ = YES
 SET &SYSOUTTRAP = 99999
 SET &RET = Ø
 LISTC     LEVEL(&A1&RP
 SET &SYSOUTTRAP = Ø
 IF &RET = Ø THEN DO
  SET &DS = YES
  WRITE SOME DATASETS/ENTRIES DO EXIST UNDER ALIAS &STR("&A1")
 END
 ELSE DO
  WRITE NO DATASETS/ENTRIES EXIST UNDER ALIAS &STR("&A1")
 END
END
ELSE DO
 WRITE NO ALIAS EXISTS FOR &STR("&A1")
 SET &SYSOUTTRAP = 99999
 SET &RET = Ø
 LISTC     LEVEL(&A1&RP
 SET &SYSOUTTRAP = Ø
 IF &RET = Ø THEN DO
  SET &HLQ = YES
  SET &DS = YES
  WRITE SOME DATASETS DO EXIST IN MCAT FOR HLQ &STR("&A1")
 END
 ELSE DO
  WRITE NO DATASET WITHOUT ALIAS EXISTS IN MCAT FOR HLQ &STR("&A1")
 END
END
SET &RET = Ø
LG &A1
IF &RET = Ø THEN DO
 SET &GROUP = YES
END
ELSE DO
 WRITE NO RACF GROUP EXISTS FOR &STR("&A1")
 CONTROL NOMSG
 SET &SYSOUTTRAP = 999999
 SET &RET = Ø
 LU &A1
 SET &SYSOUTTRAP = Ø
 SET &LURET = &RET
 CONTROL MSG
 IF &LURET < 4 THEN DO
  SET &RACFU = YES
  WRITE "&A1" USER EXISTS IN RACF
  SET &SYSOUTTRAP = 999999
  LU &A1 TSO NORACF
  SET &SYSOUTTRAP = Ø
  SET &MAXTSO = &SYSOUTLINE
  SET &RACFTSO = &STR()
  SET &N = Ø
  SET &RET = Ø
  DO WHILE &N < &MAXTSO
   SET &N = &N + 1
   SET &SYSDVAL = &STR(&&SYSOUTLINE&N)
   SET &SYSDVAL = &STR(&SYSDVAL)
   READDVAL &B1 &B2 &B3 &B4 &B5 &B6 &B7 &B8
   IF &STR(&B1) = NO AND &STR(&B2) = TSO AND &STR(&B3) = INFORMATION +
   THEN DO
    WRITE "&A1" IS NOT DEFINED AS TSO USER IN RACF
    SET &RACFTSO = NO
   END
  END
  IF &RACFTSO NE NO AND &RACFTSO NE &STR() THEN DO
   SET &RACFT = YES
   WRITE "&A1" IS DEFINED AS TSO USER IN RACF
  END
 END
 ELSE DO
  WRITE "&A1" DOES NOT EXIST AS USER IN RACF
 END
END
SET &RET = Ø
SET &MEMBER = &A1
SET &MEMBER = &MEMBER.&STR(.)
SET &SYSOUTTRAP = 999999
SEARCH ALL CLASS(DATASET) MASK(&MEMBER) LIST
SET &SYSOUTTRAP = Ø
IF &RET > Ø THEN DO
 WRITE NO RACF DATASET PROFILES EXIST FOR &STR("&A1") ANY LONGER
END
ELSE DO
 SET &DSP = YES
END
SET &MEMBER = &A1
SET &MEMBER = &MEMBER.Ø
IF &LENGTH(&STR(&A1)) < 8 THEN DO
 SET &UADS = &STR(&SYSDSN('SYS1.UADS(&MEMBER)'))
 WRITE TSO DEFINITION IN SYS1.UADS = &UADS FOR &STR("&A1")
 IF &STR(&UADS) = OK THEN DO
  SET &UADS = YES
 END
END
ELSE DO
 WRITE TSO DEFINITION IN SYS1.UADS = MEMBER NOT FOUND FOR &STR("&A1")
END
IF &DSP NE YES THEN DO
 WRITE ===> LOGICAL ERROR: RACF DS PROFILE &STR("&A1") MISSING
 SET &OUT = &STR(LOGICAL ERROR: RACF DS-PROFILE &STR("&A1") MISSING)
 PUTFILE OUT
 SET &LIST = &STR(&OUT)
 PUTFILE LIST
 WTP &STR(&OUT)
 SET &ERROR = YES
END
IF &DSP = YES THEN DO
 IF &DS NE YES THEN DO
 /* Catindex is implied user catalog HLQ in certain DFP releases */
 /* Format4 is implied HLQ when zapping VTOC */
 /* SINGLEV is defined as RACF single level dataset name in SETROPTS*/
  IF &STR(&A1) NE CATINDEX AND &STR(&A1) NE FORMAT4 AND +
  &STR(&A1) NE SINGLEV THEN DO
   WRITE ===> POSSIBLE ERROR: NO DATASET FOR DSP &STR("&A1")
   SET &OUT = &STR(POSSIBLE ERROR: NO DATASET FOR DSP &STR("&A1"))
   PUTFILE OUT
  END
 END
 IF &HLQ NE YES THEN DO
  IF &STR(&A1) NE CATINDEX AND &STR(&A1) NE FORMAT4 AND +
  &STR(&A1) NE SINGLEV THEN DO
   WRITE ===> ERROR: NO ALIAS FOR RACF DSP &STR("&A1")
   SET &OUT = &STR(ERROR: NO ALIAS FOR RACF DSP &STR("&A1"))
   PUTFILE OUT
   SET &LIST = &STR(&OUT)
   PUTFILE LIST
   WTP &STR(&OUT)
   SET &ERROR = YES
  END
 END
 IF &RACFT = YES AND &UADS = YES THEN DO
  WRITE ===> ERROR: &STR("&A1") TSO USER BOTH IN RACF AND UADS
  SET &OUT = &STR(ERROR: &STR("&A1") TSO USER BOTH IN RACF AND UADS)
  PUTFILE OUT
  SET &LIST = &STR(&OUT)
  PUTFILE LIST
  WTP &STR(&OUT)
  SET &ERROR = YES
 END
 IF &RACFU NE YES AND &GROUP NE YES THEN DO
  WRITE ===> ERROR: &STR("&A1") NEITHER RACF USER NOR GROUP
  SET &OUT = &STR(ERROR: &STR("&A1") NEITHER RACF USER NOR GROUP)
  PUTFILE OUT
  SET &LIST = &STR(&OUT)
  PUTFILE LIST
  WTP &STR(&OUT)
  SET &ERROR = YES
 END
 IF &UADS = YES AND &GROUP = YES THEN DO
```

```
    WRITE ===> ERROR: &STR("&A1") BOTH IN UADS AND RACF GROUP
    SET &OUT = &STR(ERROR: &STR("&A1") BOTH IN UADS AND RACF GROUP)
    PUTFILE OUT
    SET &LIST = &STR(&OUT)
    PUTFILE LIST
    WTP &STR(&OUT)
    SET &ERROR = YES
   END
   IF &UADS = YES AND &RACFU NE YES THEN DO
    WRITE ===> ERROR: &STR("&A1") IN UADS BUT NOT A RACF NON-TSO USER
    SET &OUT = &STR(ERROR: &STR("&A1") IN UADS BUT NOT A RACF +
    NONTSO USER.)
    PUTFILE OUT
    SET &LIST = &STR(&OUT)
    PUTFILE LIST
    WTP &STR(&OUT)
    SET &ERROR = YES
   END
  END
 END
END
CLOSFILE OUT
CLOSFILE LIST
IF &ERROR = YES THEN DO
 SET &RC = RC
 SET &N = 1
 SET &RC = &&RC&N
 DO WHILE &RC NE &STR() AND &N < 9
  %MAILSENS ID('&ID') RC('&RC') DS(&SPREFIX..&TSTAMP..TEMP.LIST) +
  DEBUG(&DEBUG) RCDOMAIN(YES) /* INFORM ERROR STATUS VIA MAIL */
  SET &N = &N + 1
  SET &RC = &&RC&N
 END
END
FREE DA('&SPREFIX..&TSTAMP..TEMP.LIST')
DEL '&SPREFIX..&TSTAMP..TEMP.LIST'
  /*
//STEP2    EXEC PGM=IKJEFTØ1,DYNAMNBR=128,  EXECUTE CLIST
// PARM='%CLIST DEBUG(NEBUG)'
//SYSTSPRT DD   SYSOUT=*
//SYSPRINT DD   SYSOUT=*
//OUT      DD   SYSOUT=*
//SYSPROC  DD   DSN=&TEMP,DISP=(OLD,PASS)
//         DD   DSN=SYSTEM.CLIST,DISP=SHR
//SYSTSIN  DD   DUMMY
//SYSIN    DD   DUMMY
//*
```

FROM A RACF GROUP POINT OF VIEW

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

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

END
ERROR DO
 SET &RET = &LASTCC
 RETURN
END
IF &FLUSH = FLUSH THEN DO
 EXIT CODE(Ø)
END
IF &STR(&DEBUG) = DEBUG THEN DO
 CONTROL MSG NOFLUSH LIST CONLIST SYMLIST
END
OPENFILE OUT OUTPUT
SET OUT =  STATUS FROM AN RACF GROUP POINT OF VIEW
PUTFILE OUT
IF &SYSPREF.. = . THEN DO
 PROFILE PREFIX(INSTPREF)           /* Default prefix */
END
ELSE DO
 PROFILE PREFIX(&SYSUID)
END
SET &TSTAMP = +
&STR(T)&SUBSTR(1:2,&SYSTIME)&SUBSTR(4:5,&SYSTIME)&SUBSTR(7:8,&SYSTIME)
DEL '&SYSPREF..&TSTAMP..TEMP.LIST'
SET &CNT = Ø
SET &RET = Ø
ALLOC FI(LIST) DA('&SYSPREF..&TSTAMP..TEMP.LIST') +
NEW SPACE(1 1) CYLINDERS +
STORCLAS(TEMP) MGMTCLAS(TEMP) RECFM(F B) LRECL(8Ø) BLKSIZE(2792Ø) REUSE
DO WHILE &RET NE Ø AND &CNT < 3ØØ THEN DO
 SET &TSTAMP = +
 &STR(T)&SUBSTR(1:2,&SYSTIME)&SUBSTR(4:5,&SYSTIME)&SUBSTR(7:8,&SYSTIME)
 DEL '&SYSPREF..&TSTAMP..TEMP.LIST'
 SET &RET = Ø
 ALLOC FI(LIST) DA('&SYSPREF..&TSTAMP..TEMP.LIST') +
 NEW SPACE(1 1) CYLINDERS +
 STORCLAS(TEMP) MGMTCLAS(TEMP) RECFM(F B) LRECL(8Ø) BLKSIZE(2792Ø) REUSE
 IF &RET NE Ø THEN DO
  FREE DA('&SYSPREF..&TSTAMP..TEMP.LIST')
 END
 SET &CNT = &CNT + 1
 SLEEP 15
END
SET &PREFIX = &SYSPREF
OPENFILE LIST OUTPUT
SET LIST =  STATUS FROM AN RACF GROUP POINT OF VIEW
PUTFILE LIST
SET &RP = &STR())
PROFILE NOPREFIX
SET &SYSOUTTRAP = 999999
SEARCH ALL CLASS(GROUP) NOMASK LIST
SET &SYSOUTTRAP = Ø
SET &MAXLINE = &SYSOUTLINE
SET &ALIAS = ALIAS
SET &P = Ø
SET &RET = Ø
DO WHILE &P < &MAXLINE
 SET &P = &P + 1
 SET &SYSDVAL = &STR(&&SYSOUTLINE&P)
 SET &SYSDVAL = &STR(&SYSDVAL)
 READDVAL &A1 &A2 &A3 &A4 &A5 &A6 &A7 &A8
 SET &&ALIAS&P = &STR(&A1)
END
SET &P = Ø
SET &RET = Ø
DO WHILE &P < &MAXLINE
 SET &P = &P + 1
 SET &A1 = &STR(&&ALIAS&P)
 SET &PREF = &STR()
 IF &LENGTH(&STR(&A1)) = 4 THEN DO
  SET &PREF = &SUBSTR(1:3,&STR(&A1))
 END
 IF &STR(&A1) NE &STR() AND &PREF NE SYS AND +
 &LENGTH(&STR(&A1)) < 9 THEN DO
  WRITE
  WRITE &STR(========>) INFORMATION FOR DSP &STR("&A1")
  SET &HLQ = NO
  SET &DS = NO
  SET &RACFU = NO
  SET &RACFT = NO
  SET &GROUP = NO
  SET &DSP = NO
  SET &UADS = NO
  SET &SYSOUTTRAP = 99999
  SET &RET = Ø
  LISTC ALL ENT('&A1'&RP
  SET &SYSOUTTRAP = Ø
  IF &RET = Ø THEN DO
   SET &HLQ = YES
   SET &SYSOUTTRAP = 99999
   SET &RET = Ø
   LISTC     LEVEL(&A1&RP
   SET &SYSOUTTRAP = Ø
   IF &RET = Ø THEN DO
    SET &DS = YES
    WRITE SOME DATASETS/ENTRIES DO EXIST UNDER ALIAS &STR("&A1")
   END
   ELSE DO
    WRITE NO DATASETS/ENTRIES EXIST UNDER ALIAS &STR("&A1")
   END
  END
  ELSE DO
   WRITE NO ALIAS EXISTS FOR &STR("&A1")
   SET &SYSOUTTRAP = 99999
   SET &RET = Ø
   LISTC     LEVEL(&A1&RP
   SET &SYSOUTTRAP = Ø
   IF &RET = Ø THEN DO
    SET &HLQ = YES
    SET &DS = YES
    WRITE SOME DATASETS DO EXIST IN MCAT FOR HLQ &STR("&A1")
   END
   ELSE DO
    WRITE NO DATASET WITHOUT ALIAS EXISTS IN MCAT FOR HLQ &STR("&A1")
   END
  END
  SET &GROUP = YES
  SET &RET = Ø
  SET &MEMBER = &A1
  SET &MEMBER = &MEMBER.&STR(.)
  SET &SYSOUTTRAP = 999999
  SEARCH ALL CLASS(DATASET) MASK(&MEMBER) LIST
  SET &SYSOUTTRAP = Ø
  IF &RET > Ø THEN DO
   WRITE NO RACF DATASET PROFILES EXIST FOR &STR("&A1") ANY LONGER
  END
  ELSE DO
   SET &DSP = YES
  END
  SET &MEMBER = &A1
  SET &MEMBER = &MEMBER.Ø
  IF &LENGTH(&STR(&A1)) < 8 THEN DO
   SET &UADS = &STR(&SYSDSN('SYS1.UADS(&MEMBER)'))
   WRITE TSO DEFINITION IN SYS1.UADS = &UADS FOR &STR("&A1")
   IF &STR(&UADS) = OK THEN DO
    SET &UADS = YES
   END
  END
  ELSE DO
   WRITE TSO DEFINITION IN SYS1.UADS = MEMBER NOT FOUND FOR &STR("&A1")
  END
  IF &GROUP NE YES THEN DO
   WRITE ===> LOGICAL ERROR: RACF GROUP &STR("&A1") MISSING
   SET &OUT = &STR(LOGICAL ERROR: RACF GROUP &STR("&A1") MISSING)
   PUTFILE OUT
   SET &LIST = &STR(&OUT)
   PUTFILE LIST
   WTP &STR(&OUT)
   SET &ERROR = YES
  END
  IF &GROUP = YES THEN DO
   IF &DS NE YES THEN DO
    WRITE ===> POSSIBLE ERROR: NO DATASET FOR GROUP &STR("&A1")
    SET &OUT = &STR(POSSIBLE ERROR: NO DATASET FOR GROUP &STR("&A1"))
    PUTFILE OUT
   END
   IF &HLQ NE YES THEN DO
    WRITE ===> POSSIBLE ERROR: NO ALIAS FOR RACF GROUP &STR("&A1")
    SET &OUT = &STR(POSSIBLE ERROR: NO ALIAS FOR RACF GROUP &STR("&A1"))
    PUTFILE OUT
   END
   IF &DSP = YES AND &HLQ NE YES THEN DO
    IF &STR(&A1) NE CATINDEX AND &STR(&A1) NE FORMAT4 AND +
    &STR(&A1) NE SINGLEV THEN DO
     WRITE ===> ERROR: RACF DATASET PROFILE EXISTS BUT +
     NO ALIAS FOR RACF GROUP &STR("&A1")
     SET &OUT = &STR(ERROR: RACF DATASET PROFILE EXISTS BUT +
     NO ALIAS FOR RACF GROUP &STR("&A1"))
     PUTFILE OUT
     SET &LIST = &STR(&OUT)
     PUTFILE LIST
     WTP &STR(&OUT)
     SET &ERROR = YES
    END
   END
   IF &DSP NE YES THEN DO
    WRITE ===> POSSIBLE ERROR: NO DATASET RACF PROFILE FOR +
    HLQ &STR("&A1")
    SET &OUT = &STR(POSSIBLE ERROR: NO DATASET RACF PROFILE FOR +
    HLQ &STR("&A1"))
    PUTFILE OUT
```

```
    IF &DS = YES THEN DO
     WRITE ===>        BUT DATASETS EXIST.
     SET &OUT = &STR(        BUT DATASETS EXIST.)
     PUTFILE OUT
    END
   END
   IF &UADS = YES THEN DO
    WRITE ===> ERROR: &STR("&A1") BOTH IN UADS AND RACF GROUP
    SET &OUT = &STR(ERROR: &STR("&A1") BOTH IN UADS AND RACF GROUP)
    PUTFILE OUT
    SET &LIST = &STR(&OUT)
    PUTFILE LIST
    WTP &STR(&OUT)
    SET &ERROR = YES
   END
  END
 END
END
CLOSFILE OUT
CLOSFILE LIST
IF &ERROR = YES THEN DO
 SET &RC = RC
 SET &N = 1
 SET &RC = &&RC&N
 DO WHILE &RC NE &STR() AND &N < 9
  %MAILSENS ID('&ID') RC('&RC') DS(&PREFIX..&TSTAMP..TEMP.LIST) +
  DEBUG(&DEBUG) RCDOMAIN(YES) /* INFORM ERROR STATUS VIA MAIL */
  SET &N = &N + 1
  SET &RC = &&RC&N
 END
END
FREE DA('&PREFIX..&TSTAMP..TEMP.LIST')
DEL '&PREFIX..&TSTAMP..TEMP.LIST'
  /*
//STEP2    EXEC PGM=IKJEFTØ1,DYNAMNBR=128,  EXECUTE CLIST
// PARM='%CLIST DEBUG(NEBUG)'
//SYSTSPRT DD   SYSOUT=*
//SYSPRINT DD   SYSOUT=*
//OUT      DD   SYSOUT=*
//SYSPROC  DD   DSN=&TEMP,DISP=(OLD,PASS)
//         DD   DSN=SYSTEM.CLIST,DISP=SHR
//SYSTSIN  DD   DUMMY
//SYSIN    DD   DUMMY
//*
```

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

Nils Plum (Denmark) © Xephon 1999

When RACF won’t tell you enough – VRA

THE PROBLEM

If you don’t have RACF access to everything, it can be difficult to get the additional access you need as your job, or the applications you support, change over time. Although there are other obstacles, one of the biggest is RACF’s search facility. This only displays the names of profiles that you already have access to, not the ones you need to know about so that you can request access to them.

For high levels with subsidiary RACF dataset profiles, this represents a major problem. If you request Alter access to the high level, and the security administrator takes you literally, you will get access only to datasets not protected by subsidiary RACF profiles – even though you have no way of knowing the names of those profiles so that you can request access to them.

It may take a lot of management intervention before a security administrator will accept a request for a high-level profile, and all profiles under it, to be modified to give you access to all datasets in that high level.

A TOOL THAT HELPS

For this, and a number of other reasons, I was given access to Vanguard RACF Administrator (VRA). This provides read-only access to RACF information that I would not otherwise have, but still prevents me from doing anything that RACF would not permit. With it, I have been able to answer all of my RACF questions, plus some others for those around me.

Although VRA is an on-line tool, it can also generate batch jobs to perform some of its functions. The following JCL was captured for a very common request – to determine what access a specific user-id has to a specific dataset.

```
//*  TELLS WHAT ACCESS A GIVEN USER HAS TO A GIVEN DATASET
//*****************************************************************
```

```
//*                                                                *
//*      THE RACF ADMINISTRATOR                                    *
//*                                                                *
//*      MEMBER: VRADSNAJ - THE DATASET AUTHORITY REPORT           *
//*                                                                *
//*      FORMAT OF INPUT CARDS IS AS FOLLOWS:                      *
//*      COL 1-8 - USER-ID                                         *
//*      COL 9     BLANK                                           *
//*      COL 10    DATASET NAME                                    *
//*****************************************************************
//VRADSNAJ PROC RSIZE=6M
//STEP01  EXEC PGM=VRADSNA,REGION=&RSIZE
//VIPOPTS   DD DISP=SHR,DSN=SYS3.VRA.VANOPTS
//REPORT   DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//        PEND
//GO   EXEC  VRADSNAJ
//SYSIN    DD  *      PLACE INPUT COMMANDS HERE
#JRP     LPWHP001.PRODO.COPY
#60T     LPLSMEDM.ARCHIVE.TEST
#JRR     LPLSMEDM.ARCHIVE.TEST
#JRP     LPLSMEDM.ARCHIVE.TEST
//
```

SOLVING THE PROBLEM

Returning to the original problem – how do we determine all of the RACF dataset profiles within a high level?

VRA requires ISPF to run on-line. At my customer's site the procedure is as follows:

1 Type $VRA in ISPF Option 6.

2 Hit ENTER to clear the banner screen, then option 2 ('Audit Services') on the VRA main menu.

3 Select option 1 for 'On-line RACF Reports'. If you are like me, this will become your starting point for all the on-line functions that you will ever use within VRA.

4 Select option 3 for 'Data Set Reports', since it is dataset profiles that are of interest, and the 'On-line Dataset Reports' panel will be displayed. The top third of the screen lists menu options, while the rest of the screen is labelled 'Masking Fields'. Except for the last field ('Show Errors Only:'), all masking fields should default to a single asterisk ('*').

TO QUOTE OR NOT TO QUOTE?

Change the ‘Dataset:’ field to the relevant high level followed by a period (full stop) and an asterisk:

```
    Dataset:  SYS1.*
```

Unlike RACF, VRA does not require surrounding single quotes. However, they do have a purpose if specified and it is important not to get confused. Without quotes, the asterisk is a wildcard that will match all profile names that begin with the characters preceding it. With quotes, an asterisk is like any other character specified.

In the example above, all RACF profiles are matched if they begin with ‘SYS1’ followed by a full stop (period). On the other hand,

```
    Dataset:  'SYS1.*'
```

only matches the single ‘SYS1.*’ profile, which is the default profile defined for the SYS1 high level.

Several options on this panel will list the specified high level’s subsidiary profiles; however, the kind of information you need will influence your decision. I find that I am usually interested in ‘Access Lists’, so I choose option 4, by typing 4 on the command line and hitting ENTER.

The normal PF7/PF8 up/down ISPF scrolling will display additional profiles if there are more than will fit on the screen – 10 profiles are displayed on a standard 24x80 non-split screen. Typing ‘S’ alongside any of the profiles listed will provide another scrollable list of users and groups with the type of access that each is allowed. An ‘S’ specified alongside any group will display yet another scrollable list (of user-ids within the group).

OTHER APPROACHES AND INFORMATION

The ‘On-line Dataset Reports’ panel (accessed by =2;1;3 from anywhere in VRA) also has 13 other mask fields that can be used to search for RACF dataset profiles on their own, or in conjunction with the ‘Dataset:’ field to narrow down the search to give a shorter resultant list.

Beyond dataset profiles, there are 18 other menu options on the online RACF Reports menu (=2;1). Option 1 covers user profiles, option 2 covers group profiles, and option 4 covers general resource profiles.

Yesterday, for example, final testing of an archive procedure failed when the generated JCL was run under the all-powerful scheduling-id. The RACF message read:

```
ICH408I USER(SCHED) GROUP(ASGROUP) NAME(SHIFT.SCHEDULER)
   STGADMIN.IGG.ALTER.SMS CL(FACILITY)
   INSUFFICIENT ACCESS AUTHORITY
   ACCESS INTENT(READ) ACCESS ALLOWED(NONE)
```

Although the scheduling-id has access to most datasets, it had not been given RACF authority to use the IDCAMS ALTER command. VRA option 4 on the ‘Online RACF Reports’ menu was used to verify this theory by confirming that the user-id where the archive procedure had been developed did have access to that RACF facility, but the scheduling-id did not. The data centre outsourcer admitted it was an oversight and resolved it immediately.

IN CONCLUSION

The data in VRA is not real-time. It is obtained from RACF on a regular schedule and stored in a VSAM file. At my customer’s site, VRA is unavailable during the lunch hour each day while the VRA VSAM file is being updated from RACF.

The intent of this article was not to describe all of VRA’s capabilities – there are many more than have been touched on here – but to indicate that VRA can be a very efficient way to get RACF information if your own RACF access is limited.

But one word of caution: capturing JCL from VRA, as illustrated above, will result in a batch job that will only work as long as changes are not made to the way that VRA is installed. For example, a recent ‘License Expired’ message from one such job resulted from the RACF technical support person installing a new version of VRA and changing the name of the dataset referenced by the VIPOPTS DD name.

Jon Pearkins
Adiant Corporation (Canada)

RACF news

Consul has announced Consul/RACF Administrator for Windows (C/RA-Win), for administering security on RACF mainframe computer systems via a Microsoft Windows graphical application. The new software allows security administrators and Help Desk staff to perform basic RACF user management tasks through a point-and-click interface.

C/RA-Win supports both sophisticated display (list) and control (update) options. The list options support filtering, sorting, printing, and the display of various user fields. Only those users within the authorized scope of the administrator are displayed.

C/RA-Win can be configured with RACF profiles to allow the administrator any or all of the following functions: resume, set password, set password to previous, and set password to default. The administrator may also simply display information.

No TSO session is required by the administrator to run C/RA-Win, which is an MS-Windows application that communicates to a started task on OS/390 via TCP/IP. C/RA-Win displays and updates the live RACF database.

C/RA-Win is fully compatible with the other Consul/RACF security products. When used in conjunction with Consul's password and account synchronization product, Consul/OnePass, C/RA-Win also controls the security administration functions supported by Consul/OnePass on NT, Unix, and NetWare platforms.

For further information contact:
Consul Risk Management, 900 Ridgefield Drive, Suite 140, Raleigh, NC 27609, USA.
Tel: (919) 790 8220.
URL: http://www.consul.com.

* * *

RACF users can benefit from Tivoli Security Management 3 for OS/390 which enables OS/390 to participate in Tivoli framework-based security management. The new release extends Security Profile support to the RACF component of the OS/390 Security Server.

For further information contact your local IBM representative.

* * *