August 1999

In this issue

3  Automatic password change by RRSF
8  Replacement for the RACF Report Writer – part 6
31 An ISPF dialog to manage catalog aliases
41 A RACFPROF ISPF interface – part 2
49 TSO command to maintain RACF user-data
67 RACF news
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/racfupdate.html; you will need the user-id shown on your address label.

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.
Automatic password change by RRSF

I work on TSO in both Lyon and in Paris and I have the same user-id in these two towns. For some time, I have dreamed of being able to change my password for one user-id and have this password change automatically for my other user-id.

This can now be achieved using RRSF (RACF Remote Sharing Facility).

Multisystem RRSF node support is an enhancement to RACF 2.2 shipped via PTF UW90235 for APAR OW13567. It extends the function provided by RRSF for MVS system images that share a RACF database.

The RACF remote sharing facility provided by RACF 2.2 allows you to configure MVS system images connected by APPC/MVS into a network of RRSF nodes capable of communicating RACF commands and password changes to each other.

We use RRSF in two ways:

• Firstly at Lyon, we synchronize the password between two TSO user-ids on the same OS/390 node on the same system (see job PASSYNC).

• Secondly, via the network, the same TSO user-id on two OS/390 systems is synchronized.

To customize RRSF at Lyon the SYS1.PARM.LIB has been updated. The following members have been created:

• APPCXMLY for APPC.

• IRROPTLY for RACF RRSF for the node situated in Lyon.

• APPCCMC1 for APPC.

• IRROPTC1 for RACF RRSF for the node situated in Paris.

In addition, a VTAM major node should be added in VTAM (A691RRSF in Lyon and ACP1RRSF in Paris).

RACF should be defined as a started class (see job RACFAS).
The logmod RRSFLOGM is necessary because the VTAM node A691RRSF uses it. The job RRSFDATA has to be submitted to define RRSF to RACF.

Finally, the following new commands are available:

- RACF can be stopped using ‘$close’.
- RACF can be restarted by the command ‘S RACF,SUB=MSTR’.

**IRROPTLY**

```
TARGET NODE(LYO1)  PREFIX(RRSF) -
   WORKSPACE(STORCLAS(SPECIAL) MGMTCLAS(NOMIG)) -
   PROTOCOL(APPC(LUNAME(A691RRSF) TPNAME(IERRRACF) )) -
   DESCRIPTION('RACF SYSTEM LYO1') -
   LOCAL -
   OPERATIVE
SET PWSYNC (NOTIFY(ALWAYS(LYO1.DUNAND)))
```

**PASSYNC**

```
//PASSYNC JOB SYS,DUNAND,CLASS=W,MSGCLASS=3,MSGLEVEL=(1,1),
//       NOTIFY=&SYSUID
//STEP1 EXEC PGM=IKJEFT01,DYNAMNBR=20,TIME=1440,REGION=4096K
//SYSTSPRT DD SYSOUT=* 
//SYSTSIN  DD *
RACLINK ID(PSY1) LIST(*.*) 
RACLINK LIST(*.*) 
RACLINK ID(PHILG) DEFINE(LYO1.PSY1 LY01.YSVB0SPH) PEER(PWSYNC)
```

**RACFAS**

```
LOGON
RDEFINE STARTED  RACF.* UACC(READ) +
   STDATA(USER(RACFAS) GROUP(SYS1) PRIVILEGED=YES))
AU RACFAS DFLTGRP(SYS1) OWNER(SYS1) NAME('STC')
```

**SRVCLASS**

```
//SRVCLASS JOB SYS,DUNAND,CLASS=W,MSGCLASS=Y,MSGLEVEL=(1,1),
//       NOTIFY=&SYSUID
//STEP1 EXEC PGM=IKJEFT01,DYNAMNBR=20,TIME=1440,REGION=4096K
//SYSTSPRT DD SYSOUT=* 
//SYSTSIN  DD *
RDEFINE SDSF ISFATTR.JOB.SRVCLASS UACC(READ) OWNER(SYS1)
```
PERMIT ISFATTR.JOB.SRVCLASS CLASS(SDSF) ID(SYS1) ACCESS(UPDATE)
PERMIT ISFATTR.JOB.SRVCLASS CLASS(SDSF) ID(DUNAND) ACCESS(UPDATE)
PERMIT ISFATTR.JOB.SRVCLASS CLASS(SDSF) ID(psy1) ACCESS(UPDATE)
PERMIT ISFATTR.JOB.SRVCLASS CLASS(SDSF) ID(PHILG) ACCESS(UPDATE)
SETR CLASSACT(SDSF) RACLST(SDSF) REFRESH

STRATAR
SETR RACLST(STARTED) REFRESH

APPCCMLY
LUADD ACBNAME(A691AS0) TPDATA(RDVUSER.APPCTP) TLEVEL(SYSTEM)
LUADD ACBNAME(A691APP) BASE TPDATA(RDVUSER.APPCTP) TLEVEL(USER)
LUADD ACBNAME(SASSESS) BASE TPDATA(RDVUSER.APPCTP) TLEVEL(USER)
LUADD ACBNAME(A691RRSF) BASE TPDATA(RDVUSER.APPCTP) NOSCHED
LMADD ACBNAME(*) LUNAME(*) LOGMODE(APPCHOST)
SIDEINFO DATASET(RDVUSER.APPCSI)

ARRSF691
ARRSF691 VBUILD TYPE=APPL APPLICATION MAJOR NODE
A691RRSF APPL ACBNAME=A691RRSF,
       APPC=YES,
       AUTOSES=Ø,
       DDRAINL=NALLOW,
       DLOGMOD=IRRMODE,
       DMINWNL=5,
       DMINWNR=5,
       DRESPL=NALLOW,
       DSESLIM=10,
       LMDENT=19,
       MODETAB=APPCMODE,
       PARSESS=YES,
       SECACPT=AVPV,
       SRBEXIT=YES

AAPPCCP1
AAPPCCP1 VBUILD TYPE=APPL APPLICATION MAJOR NODE
ACPIASCH APPL ACBNAME=ACPIASCH,
       APPC=YES,
       AUTOSES=Ø,
       DDRAINL=NALLOW,
       DLOGMOD=APPCHOST,
       DMINWNL=5,
       DMINWNR=5,
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRESPL</td>
<td>NALLOW.</td>
</tr>
<tr>
<td>DSESLIM</td>
<td>10</td>
</tr>
<tr>
<td>LMDENT</td>
<td>19</td>
</tr>
<tr>
<td>MODETAB</td>
<td>APPCMODE</td>
</tr>
<tr>
<td>PARSESS</td>
<td>YES</td>
</tr>
<tr>
<td>SECACPT</td>
<td>CONV</td>
</tr>
<tr>
<td>SRBEXIT</td>
<td>YES</td>
</tr>
<tr>
<td>ACBNAME</td>
<td>ACP1APPC</td>
</tr>
<tr>
<td>APPC</td>
<td>YES</td>
</tr>
<tr>
<td>AUTOSES</td>
<td>Ø</td>
</tr>
<tr>
<td>DDRAINL</td>
<td>NALLOW</td>
</tr>
<tr>
<td>DLOGMOD</td>
<td>APPCHOST</td>
</tr>
<tr>
<td>DMINWNL</td>
<td>5</td>
</tr>
<tr>
<td>DMINWNR</td>
<td>5</td>
</tr>
<tr>
<td>DRESPL</td>
<td>NALLOW</td>
</tr>
<tr>
<td>DSESLIM</td>
<td>10</td>
</tr>
<tr>
<td>LMDENT</td>
<td>19</td>
</tr>
<tr>
<td>MODETAB</td>
<td>APPCMODE</td>
</tr>
<tr>
<td>PARSESS</td>
<td>YES</td>
</tr>
<tr>
<td>SECACPT</td>
<td>CONV</td>
</tr>
<tr>
<td>SRBEXIT</td>
<td>YES</td>
</tr>
<tr>
<td>ACBNAME</td>
<td>ACP1ISPF</td>
</tr>
<tr>
<td>APPC</td>
<td>YES</td>
</tr>
<tr>
<td>AUTOSES</td>
<td>Ø</td>
</tr>
<tr>
<td>DDRAINL</td>
<td>NALLOW</td>
</tr>
<tr>
<td>DLOGMOD</td>
<td>APPCHOST</td>
</tr>
<tr>
<td>DMINWNL</td>
<td>5</td>
</tr>
<tr>
<td>DMINWNR</td>
<td>5</td>
</tr>
<tr>
<td>DRESPL</td>
<td>NALLOW</td>
</tr>
<tr>
<td>DSESLIM</td>
<td>10</td>
</tr>
<tr>
<td>LMDENT</td>
<td>19</td>
</tr>
<tr>
<td>MODETAB</td>
<td>APPCMODE</td>
</tr>
<tr>
<td>PARSESS</td>
<td>YES</td>
</tr>
<tr>
<td>SECACPT</td>
<td>CONV</td>
</tr>
<tr>
<td>SRBEXIT</td>
<td>YES</td>
</tr>
<tr>
<td>ACBNAME</td>
<td>ACP1RRSF</td>
</tr>
<tr>
<td>APPC</td>
<td>YES</td>
</tr>
<tr>
<td>AUTOSES</td>
<td>Ø</td>
</tr>
<tr>
<td>DDRAINL</td>
<td>NALLOW</td>
</tr>
<tr>
<td>DLOGMOD</td>
<td>IRRMODE</td>
</tr>
<tr>
<td>DMINWNL</td>
<td>5</td>
</tr>
</tbody>
</table>

**ARRSFCP1**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBUILD TYPE</td>
<td>APPL</td>
</tr>
<tr>
<td>ACBNAME</td>
<td>ACP1RRSF</td>
</tr>
<tr>
<td>APPC</td>
<td>YES</td>
</tr>
<tr>
<td>AUTOSES</td>
<td>Ø</td>
</tr>
<tr>
<td>DDRAINL</td>
<td>NALLOW</td>
</tr>
<tr>
<td>DLOGMOD</td>
<td>IRRMODE</td>
</tr>
<tr>
<td>DMINWNL</td>
<td>5</td>
</tr>
</tbody>
</table>

* STATOPT = 'APPLID APPC ' * STATOPT = 'APP ISPF '
IRROPTC1

TARGET NODE(CRETEIL) SYSNAME(ESA1) PREFIX(RRSF) -
   WORKSPACE(STORCLAS(SPECIAL) MGMTCLAS(NOMIG)) -
   PROTOCOL(APPC(LUNAME(ACP1RRSF) TPNAME(IRRRACF)) ) -
   DESCRIPTION('RACF SYSTEM ESA1') -
   MAIN -
   LOCAL -
   OPERATIVE

TARGET NODE(CRETEIL) SYSNAME(ESA2) PREFIX(RRSF) -
   WORKSPACE(STORCLAS(SPECIAL) MGMTCLAS(NOMIG)) -
   PROTOCOL(APPC(LUNAME(ACP2RRSF) TPNAME(IRRRACF)) ) -
   DESCRIPTION('RACF SYSTEM ESA2') -
   OPERATIVE

TARGET NODE(C3PO) PREFIX(RRSF) -
   WORKSPACE(STORCLAS(SPECIAL) MGMTCLAS(NOMIG)) -
   PROTOCOL(APPC(LUNAME(C3PORRSF) TPNAME(IRRRACF)) ) -
   DESCRIPTION('RACF SYSTEM C3PO') -
   OPERATIVE

TARGET NODE(LYO1) PREFIX(RRSF) -
   WORKSPACE(STORCLAS(SPECIAL) MGMTCLAS(NOMIG)) -
   PROTOCOL(APPC(LUNAME(A691RRSF) TPNAME(IRRRACF)) ) -
   DESCRIPTION('LYON SYSTEM ') -
   OPERATIVE

SET PWSYNC (NOTIFY(ALWAYS(ESA2.GIL)))

RRSFLOGM

IRRMODE   MODEENT LOGMODE=IRRMODE,           *
   RUSIZES=X'8989',                *
   SRCVPAC=X'00',                 *
   SSNDPAC=X'01'
TITLE 'IRRMODE'
MODEEND
END

RRSFDATA

LOGON
SETOPTS CLASSACT(RRSFDATA)
Replacement for the RACF Report Writer – part 6

This month we conclude 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.

To enable the article to be published in a manageable form, considerable editing of the original code has been necessary. The original, unedited code is available on our Web site (www.xephon.com) and can be downloaded in the usual way. This service is free to subscribers.

)`
%END;
%IF &REQ = EXTRACT %THEN
  %DO;
    %PUT Including datadefinition for IPCCTL extension;
    WHEN('IPCCTL') DO;
    INPUT %SMFB0HDR(REQ=EXTRACT) ICTCLASS $ 282-289
<table>
<thead>
<tr>
<th>Variable</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICTUSERNAME</td>
<td>291-310</td>
</tr>
<tr>
<td>ICTUSERNAME</td>
<td>312-315</td>
</tr>
<tr>
<td>ICTUPRE</td>
<td>317-320</td>
</tr>
<tr>
<td>ICTUVF</td>
<td>322-325</td>
</tr>
<tr>
<td>ICTUNJE</td>
<td>327-330</td>
</tr>
<tr>
<td>ICTUAAUD</td>
<td>332-335</td>
</tr>
<tr>
<td>ICTUSESPEC</td>
<td>337-340</td>
</tr>
<tr>
<td>ICTUDFLT</td>
<td>342-345</td>
</tr>
<tr>
<td>ICTUUNDF</td>
<td>347-350</td>
</tr>
<tr>
<td>ICTUERR</td>
<td>352-355</td>
</tr>
<tr>
<td>ICTUTRST</td>
<td>357-360</td>
</tr>
<tr>
<td>ICTUSEST</td>
<td>362-369</td>
</tr>
<tr>
<td>ICTUSURO</td>
<td>371-374</td>
</tr>
<tr>
<td>ICTURMT</td>
<td>376-379</td>
</tr>
<tr>
<td>ICTUPRVL</td>
<td>381-384</td>
</tr>
<tr>
<td>ICTUSECL</td>
<td>386-393</td>
</tr>
<tr>
<td>ICTUENX</td>
<td>395-402</td>
</tr>
<tr>
<td>ICTUSUSR</td>
<td>404-411</td>
</tr>
<tr>
<td>ICTUSNOD</td>
<td>413-420</td>
</tr>
<tr>
<td>ICTUSGRP</td>
<td>422-429</td>
</tr>
<tr>
<td>ICTUSPOE</td>
<td>431-438</td>
</tr>
<tr>
<td>ICTUSPCL</td>
<td>440-447</td>
</tr>
<tr>
<td>ICTUTUSR</td>
<td>449-456</td>
</tr>
<tr>
<td>ICTUTGRP</td>
<td>458-465</td>
</tr>
<tr>
<td>ICTUTDFT</td>
<td>467-470</td>
</tr>
<tr>
<td>ICTUTSEC</td>
<td>472-475</td>
</tr>
<tr>
<td>ICTAPPC</td>
<td>477-492</td>
</tr>
<tr>
<td>ICTAUDIT</td>
<td>494-504</td>
</tr>
<tr>
<td>ICTORUID</td>
<td>506-515</td>
</tr>
<tr>
<td>ICTOEUID</td>
<td>517-526</td>
</tr>
<tr>
<td>ICTOSUID</td>
<td>528-537</td>
</tr>
<tr>
<td>ICTORGID</td>
<td>539-548</td>
</tr>
<tr>
<td>ICTOEGRID</td>
<td>550-559</td>
</tr>
<tr>
<td>ICTOSGID</td>
<td>561-570</td>
</tr>
<tr>
<td>ICTKOID</td>
<td>572-581</td>
</tr>
<tr>
<td>ICTKOGID</td>
<td>583-592</td>
</tr>
<tr>
<td>ICTUID</td>
<td>594-603</td>
</tr>
<tr>
<td>ICTGID</td>
<td>605-614</td>
</tr>
<tr>
<td>ICTRSV1</td>
<td>616-619</td>
</tr>
<tr>
<td>ICTRSV2</td>
<td>621-624</td>
</tr>
<tr>
<td>ICTRSV3</td>
<td>626-629</td>
</tr>
<tr>
<td>ICTOLORD</td>
<td>631-634</td>
</tr>
<tr>
<td>ICTOLOWR</td>
<td>636-639</td>
</tr>
<tr>
<td>ICTOLOEX</td>
<td>641-644</td>
</tr>
<tr>
<td>ICTOLGRD</td>
<td>646-649</td>
</tr>
<tr>
<td>ICTOLGWR</td>
<td>651-654</td>
</tr>
<tr>
<td>ICTOLGEX</td>
<td>656-659</td>
</tr>
<tr>
<td>ICTOLWRD</td>
<td>661-664</td>
</tr>
<tr>
<td>ICTOLWWR</td>
<td>666-669</td>
</tr>
<tr>
<td>ICTOLWEX</td>
<td>671-674</td>
</tr>
<tr>
<td>ICTRSV4</td>
<td>676-679</td>
</tr>
</tbody>
</table>
ICTRSV5 $ 681-684
ICTRSV6 $ 686-689
ICTNWORD $ 691-694
ICTNWWWR $ 696-699
ICTNWOEX $ 701-704
ICTNWGRD $ 706-709
ICTNWGWR $ 711-714
ICTNWGEX $ 716-719
ICTNWWRD $ 721-724
ICTNWWWR $ 726-729
ICTNWWEX $ 731-734
ICTSRVCD $ 736-746
ICTRSV7 $ 748-751
ICTRSV8 $ 753-756
ICTRSV9 $ 758-761
ICTRQORD $ 763-766
ICTRQOWR $ 768-771
ICTRQOEX $ 773-776
ICTRQGRD $ 778-781
ICTRQGWR $ 783-786
ICTRQGEX $ 788-791
ICTRQWRD $ 793-796
ICTRQWWR $ 798-801
ICTRQWEX $ 803-806
ICTKEY $ 808-815
ICTID 817-826
ICTCRUID 828-837
ICTCRGID 839-848

; LABEL ICTCLASS = 'Class name'
ICTUSERN = 'User name'
ICTUTKNE = 'Utoken encr.?'
ICTUPRE = 'Pre-1.9?'
ICTUVFYX = 'VERIFYX propagation?'
ICTUNJEU = 'Undefined NJE user?'
ICTUAUD = 'UAUDIT?'
ICTUSPEC = 'RACF special?'
ICTUDFLT = 'Default token?'
ICTUUNDF = 'Undefined user?'
ICTUERR = 'Token in error?'
ICTUSTRT = 'User trusted?'
ICTUSEST = 'Session type'
ICTUSURO = 'Surrogate user?'
ICTURMT = 'Remote job?'
ICTUPRVL = 'Privileged user?'
ICTUSECL = 'User SECLABEL'
ICTUEXND = 'Execution node'
ICTUSUSR = 'Submitting user'
ICTUSNOD = 'Submitting node'
ICTUSGRP = 'Submitting group'
ICTUSPOE = 'Port of entry'
ICTUSPCL = 'Class of POE'
ICTUTUSR = 'Userid'
ICTUTGRP = 'Groupid'
ICTUTDFT = 'Default group?'
ICTUTSEC = 'Default SECLABEL?'
ICTAPPC = 'APPC key link'
ICTAUDIT = 'Audit code'
ICTORUID = 'Old real UID'
ICTOEUID = 'Old effective UID'
ICTOSUID = 'Old saved UID'
ICTORGID = 'Old real GID'
ICTOEGID = 'Old effective GID'
ICTOSGID = 'Old saved GID'
ICTKOUID = 'Key owner UID'
ICTKOGID = 'Key owner GID'
ICTRSV1 = 'Reserved'
ICTRSV2 = 'Reserved'
ICTRSV3 = 'Reserved'
ICTOLORD = 'Old Owner read ?'
ICTOLLOWR = 'Old Owner write ?'
ICTOLOEX = 'Old Owner exec ?'
ICTOLGRD = 'Old Group read ?'
ICTOLGWR = 'Old Group write ?'
ICTOLGEX = 'Old Group exec ?'
ICTOLWRD = 'Old Other read ?'
ICTOLWWR = 'Old Other write ?'
ICTOLWEX = 'Old Other exec ?'
ICTRSV4 = 'Reserved'
ICTRSV5 = 'Reserved'
ICTRSV6 = 'Reserved'
ICTNEWRD = 'New Owner read ?'
ICTNEWWR = 'New Owner write ?'
ICTNEWEX = 'New Owner exec ?'
ICTNWGRD = 'New Group read ?'
ICTNWGWR = 'New Group write ?'
ICTNWGEX = 'New Group exec ?'
ICTNWWRD = 'New Other read ?'
ICTNWWWR = 'New Other write ?'
ICTNWWEX = 'New Other exec ?'
ICTSRVCD = 'Service code'
ICTRSV7 = 'Reserved'
ICTRSV8 = 'Reserved'
ICTRSV9 = 'Reserved'
ICTRQORD = 'Req Owner read ?'
ICTRQOWR = 'Req Owner write ?'
ICTRQOEX = 'Req Owner exec ?'
ICTRQGRD = 'Req Group read ?'
ICTRQGWR = 'Req Group write ?'
ICTRQGEX = 'Req Group exec ?'
ICTRQWRD = 'Req Other read ?'
ICTRQWWR = 'Req Other write ?'
IGTROWEX = 'Req Other exec ?'
ICTKEY = 'IPC key'
ICTID = 'IPC unique id'
ICTCRUID = 'Creator UID'
ICTCRGID = 'Creator GID'

: OUTPUT RACF.IPCCTL;
END;
%MEND IPCCTL;
./ ADD LIST=ALL,NAME=SETGROUP
%MACRO SETGROUP(REQ=);
%LET REQ = %UPCASE(&REQ);
%IF &REQ = DEFINE %THEN
%DO;
%PUT Including variables from SETGROUP extension;
RACF.SETGROUP (KEEP=%SMFHDR
  %SMF8HDR(REQ=DEFINE)
  SGRCLASS
  SGRUSERN
  SGRUTKNE
  SGRUPRE
  SGRUVFYX
  SGRUJEU
  SGRUAUD
  SGRUSPEC
  SGRUDFLT
  SGRUNDF
  SGRUERR
  SGRUTRST
  SGRUSEST
  SGRUSURO
  SGRURMT
  SGRUPRVL
  SGRUSECL
  SGRUXND
  SGRUSUSR
  SGRUSNOD
  SGRUSGRP
  SGRUSPOE
  SGRUSPCL
  SGRUTUSR
  SGRUTGRP
  SGRUTDFT
  SGRUTSEC
  SGRAPPC
  SGRAUDIT
  SGRORUID
  SGROEUID
  SGROSUID
  SGRORGID
SGROEGID
SGROSGID
SGRDCELK
SGRAUTYP

%END;
%IF &REQ = EXTRACT %THEN
%DO;
  %PUT Including datadefinition for SETGROUP extension;
  WHEN('SETGROUP') DO;
  INPUT %SMFB0HDR(REQ=EXTRACT)
   SGRCLASS $  282-289
   SGRUSERN $  291-310
   SGRUTKNE $  312-315
   SGRUPRE $   317-320
   SGRUVFYX $  322-325
   SGRUNJEU $  327-330
   SGRUUAUD $  332-335
   SGRUSPEC $  337-340
   SGRUDFLT $  342-345
   SGRUUNDF $  347-350
   SGRUERR $   352-355
   SGRUTRST $  357-360
   SGRUSEST $  362-369
   SGRUSURO $  371-374
   SGRURMT $   376-379
   SGRUPRVL $  381-384
   SGRUSECL $  386-393
   SGRUEXND $  395-402
   SGRUSUSR $  404-411
   SGRUSNOD $  413-420
   SGRUSGRP $  422-429
   SGRUSPOE $  431-438
   SGRUSPCL $  440-447
   SGRUTUSR $  449-456
   SGRUTGRP $  458-465
   SGRUTDFT $  467-470
   SGRUTSEC $  472-475
   SGRAPPCC $  477-492
   SGRAUDIT $  494-504
   SGRORUID $  506-515
   SGROEUID $  517-526
   SGROSUID $  528-537
   SGROGID $   539-548
   SGROEGID $  550-559
   SGROSGID $  561-570
   SGRDCELK $  572-587
   SGRAUTYP $  589-601
;  
LABEL SGRCLASS = 'Class name'
   SGRUSERN = 'User name'

SGRUTKNE = 'Utoken encr.?'  
SGRUPRE = 'Pre-1.9?' 
SGRUVFYX = 'VERIFYX propagation?' 
SGRUNJEU = 'Undefined NJE user?' 
SGRUAUD = 'UAUDIT?' 
SGRUSPEC = 'RACF special?' 
SGRUDFLT = 'Default token?' 
SGRUUNDF = 'Undefined user?' 
SGRUERR = 'Token in error?' 
SGRUTRST = 'User trusted?' 
SGRUSEST = 'Session type' 
SGRUSURO = 'Surrogate user?' 
SGRURMT = 'Remote job?' 
SGRUPRVL = 'Privileged user?' 
SGRUSECL = 'User SECLABEL' 
SGRUEXND = 'Execution node' 
SGRUSUSR = 'Submitting user' 
SGRUSNOD = 'Submitting node' 
SGRUSGRP = 'Submitting group' 
SGRUSPOE = 'Port of entry' 
SGRUSPCL = 'Class of POE' 
SGRUTUSR = 'Userid' 
SGRUTGRP = 'Groupid' 
SGRUTDFT = 'Default group?' 
SGRUTSEC = 'Default SECLABEL?' 
SGRAPPC = 'APPC key link' 
SGRAUDIT = 'Audit code' 
SGORUID = 'Old real UID' 
SGOEUID = 'Old effective UID' 
SGOSUID = 'Old saved UID' 
SGORGID = 'Old real GID' 
SGOEGID = 'Old effective GID' 
SGOSGID = 'Old saved GID' 
SGROCELK = 'DCE link' 
SGRAUTYP = 'Request type' 
;
  OUTPUT RACF.SETGROUP; 
END; 
%MEND SETGROUP; 
./        ADD   LIST=ALL,NAME=CKOWN2 
%MACRO CKOWN2(REQ=); 
%LET REQ = %UPCASE(&REQ); 
%IF &REQ = DEFINE %THEN 
  %DO; 
    %PUT Including variables from CKOWN2 extension; 
    RACF.CKOWN2 (KEEP=%SMFHDR 
                  %SMF8HDR(REQ=DEFINE) 
                  CO2CLASS 
                  CO2USERN 
                  CO2UTKNE 
                  CO2UTGR 
                  CO2UTDFT 
                  CO2UTSEC 
                  CO2APPC 
                  CO2AUDIT 
                  CO2ORUID 
                  CO2OEUID 
                  CO2OSUID 
                  CO2ORCID 
                  CO2OEGID 
                  CO2OSGID 
                  CO2ROCELK 
                  CO2AUTYP); 
  %END; 
%ELSE 
  %DO; 
    %PUT Including variables from CKOWN2 extension; 
    RACF.CKOWN2 (KEEP=%SMFHDR 
                  %SMF8HDR(REQ=DEFINE) 
                  CO2CLASS 
                  CO2USERN 
                  CO2UTKNE 
                  CO2UTGR 
                  CO2UTDFT 
                  CO2UTSEC 
                  CO2APPC 
                  CO2AUDIT 
                  CO2ORUID 
                  CO2OEUID 
                  CO2OSUID 
                  CO2ORCID 
                  CO2OEGID 
                  CO2OSGID 
                  CO2ROCELK 
                  CO2AUTYP); 
  %END; 
%END; 

© 1999. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.
%END;
%IF &REQ = EXTRACT %THEN
%DO;
  %PUT Including datadefinition for CKOWN2 extension;
  WHEN('CKOWN2') DO;
    INPUT %SMF8HDR(REQ=EXTRACT)
    CO2CLASS $ 282-289
    CO2USERN $ 291-310
    CO2UTKNE $ 312-315
LABEL CO2CLASS = 'Class name'
CO2USERN = 'User name'
CO2UTKNE = 'Utoken encr.?'
CO2UPRE = 'Pre-1.9?'
CO2UVFYX = 'VERIFYX propagation?'
CO2UNJEU = 'Undefined NJE user?'
CO2UUAUD = 'UAUDIT?'
CO2USPEC = 'RACF special?'
CO2UDFLT = 'Default token?'

CO2UUndef = 'Undefined user?'
CO2UErr = 'Token in error?'
CO2UTRST = 'User trusted?'
CO2USEST = 'Session type'
CO2USURO = 'Surrogate user?'
CO2URMT = 'Remote job?'
CO2UPRVL = 'Privileged user?'
CO2USECL = 'User SECLABEL'
CO2UEXND = 'Execution node'
CO2USUSR = 'Submitting user'
CO2USNOD = 'Submitting node'
CO2USGRP = 'Submitting group'
CO2USPOE = 'Port of entry'
CO2USPCL = 'Class of POE'
CO2UTUSR = 'Userid'
CO2UTGRP = 'Groupid'
CO2UTDFT = 'Default group?'
CO2UTSEC = 'Default SECLABEL?'
CO2APPC = 'APPC key link'
CO2AUDIT = 'Audit code'
CO2ORUID = 'Old real UID'
CO2OEUID = 'Old effective UID'
CO2OSUID = 'Old saved UID'
CO2ORGID = 'Old real GID'
CO2OEGID = 'Old effective GID'
CO2OSGID = 'Old saved GID'
CO2PATHN = 'Path name'
CO2F1ID = '1st file id'
CO2F1UID = '1st owner UID'
CO2F1GID = '1st owner GID'
CO2F2ID = '2nd file id'
CO2F2UID = '2nd owner UID'
CO2F2GID = '2nd owner GID'
CO2DCELK = 'DCE link'
CO2AUTYP = 'Request type'

; OUTPUT RACF.CKOWN2;
END;
%END;
%MEND CKOWN2;
./ ADD LIST=ALL,NAME=RAUDIT
%MACRO RAUDIT(REQ=);
%LET REQ = %UPCASE(&REQ);
%IF &REQ = DEFINE %THEN
%DO;
%PUT Including variables from RAUDIT extension;
RACF.RAUDIT (KEEP=%SMFHDR
 %SMF8HDR(REQ=DEFINE)
 RAUCLASS
 RAUUSERN
 RAUUTKNE

%END;
%IF &REQ = EXTRACT %THEN
%DO:
  %PUT Including datadefinition for RAUDIT extension;
  WHEN('RAUDIT') DO;
  INPUT %SMF80HDR(REQ=EXTRACT)
  RAUCLASS $  282-289
  RAUUSERN $  291-310
  RAUUKNE $  312-315
  RAUUPRE $  317-320
  RAUVFYX $  322-325
  RAUUNJEU $  327-330
  RAUUUAUD $  332-335
  RAUUSPEC $  337-340
  RAUUDFLT $  342-345
  RAUUNDF $  347-350
LABEL RAUCLASS = 'Class name'
RAUUSERN = 'User name'
RAUUTKNE = 'Utoken encr.?'
RAUUPRE = 'Pre-1.9?'
RAUUVFYX = 'VERIFYX propagation?'
RAUUNJEU = 'Undefined NJE user?'
RAUUAUD = 'UAUDIT?'
RAUUSPEC = 'RACF special?'
RAUUDFLT = 'Default token?'
RAUUUNDF = 'Undefined user?'
RAUERRR = 'Token in error?'
RAUUTRST = 'User trusted?'
RAUUSEST = 'Session type'
RAUUSURO = 'Surrogate user?'
RAUURMT = 'Remote job?'
RAUUPRVL = 'Privileged user?'
RAUUSECL = 'User SECLABEL'
RAUUEXND = 'Execution node'
RAUUSUSR = 'Submitting user'
RAUUSNOD = 'Submitting node'
RAUUSGRP = 'Submitting group'
RAUUSPOE = 'Port of entry'
RAUUSPCL = 'Class of POE'
RAUUTUSR = 'Userid'
RAUUTGRP = 'Groupid'
RAUUTDFT = 'Default group?'
RAUUTSEC = 'Default SECLABEL?'
RAUAPPC = 'APPC key link'
RAUAUDIT = 'Audit code'
RAUORUID = 'Old real UID'
RAUOEUID = 'Old effective UID'
RAUOSUID = 'Old saved UID'
RAUORGID = 'Old real GID'
RAUOEGID = 'Old effective GID'
RAUOSGID = 'Old saved GID'
RAUPATHN = 'Path name'
RAUFILID = 'File id'

; OUTPUT RACF.RAUDIT;
END;
%MEND RAUDIT;
./        ADD   LIST=ALL,NAME=RACFINIT
%MACRO RACFINIT(REQ=);
%LET REQ = %UPCASE(&REQ);
%IF &REQ = DEFINE %THEN
%DO:
  %PUT Including variables from RACFINIT extension;
  RACF.RACFINIT (KEEP=%SMFHDR
  RINRVS1
  RINTIME
  RINDATE
  RINSMFID
  RINDSNAM
  RINDSVOL
  RINDSUNT
  RINUADSN
  RINUADSV
  RINRACST
  RINDSNST
  RINRINPR
  RINRACPR
  RINRDEPR
  RINRINPO
  RINRACPO
  RINPWDEX
  RINTAPST
  RINDASST
  RINTRMST
  RINCMDEX
  RINDLCEX
  RINADSP
  RINENCEX

%END;
%MEND RACFINIT;
%END;
%IF &REQ = EXTRACT %THEN
%DO;
%PUT Including datadefinition for RACFINIT extension;
    WHEN('RACFINIT') DO;
        INPUT RINRSV1 $ 10-17
            @19
                    RINTIME TIME8.
            @28
                    RINDATE YYMMDD10.
                    RINSMFID $ 39-42
                    RINDSNAM $ 44-87
                    RINDSVOL $ 89-94
                    RINDSUNT $ 96-98
                    RINUADSN $ 100-143
                    RINUADSV $ 145-150
                    RINRACST $ 152-155
                    RINDSNST $ 157-160
                    RINRINPR $ 162-165
                    RINRACPR $ 167-170
                    RINRDEPR $ 172-175
                    RINRINPO $ 177-180
                    RINRACPO $ 182-185
                    RINPWDEX $ 187-190
                    RINTAPST $ 192-195
                    RINDASST $ 197-200
                    RINTRMST $ 202-205
                    RINCMDEX $ 207-210
                    RINDLCEX $ 212-215
                    RINADSP $ 217-220
                    RINENCEX $ 222-225
                    RINNMCNV $ 227-230
                    RINTPVOL $ 232-235
                    RINDUPDS $ 237-240
                    RINDASD $ 242-245
                    RINFRAPR $ 247-250
                    RINRLIPR $ 252-255
                    RINRLISL $ 257-260
                    RINRDEPO $ 262-265
                    RINAUUSR $ 267-270
                    RINAUGRP $ 272-275

© 1999. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.
FORMAT RINTIME TIME8.
  RINDATE YYYYMMDD8.

; LABEL RINRSV1 = 'Reserved'
RINTIME = 'Time written'
RINDATE = 'Date written'
RINSMFD = 'SMF id'
RINDSNAM = 'RACF dataset'
RINDSVOL = 'RACF volume'
RINDSUNT = 'RACF unitname'
RINUADSN = 'UADS dataset'
RINUADSV = 'UADS volume'
RINRACST = 'RACINIT stats?'
RINDSNST = 'Dataset stats?'
RINRINPR = 'Pre-RACINIT exit?'
RINRACPR = 'Pre-RACHECK exit?'
RINRDEPR = 'Pre-RACDEF exit?'
RINRINPO = 'Post-RACINIT exit?'
RINRACPO = 'Post-RACHECK exit?'
RINPWDEX = 'New password exit?'
RINTAPST = 'TAPEVOL stats?'
RINDASST = 'DASD stats?'
RINTRMST = 'Terminal stats?'
RINCMDEX = 'Command exit?'
RINDLCEX = 'Del command exit?'
RINADSP = 'ADSP active?'
RINENCEX = 'Encryption exit?'
RINNMNCNV = 'Naming conv. present?'
RINTPVOL = 'TAPEVOL active?'
RINDUPDS = 'Duplicate datasetname allowed?'
RINDASD = 'DASD active?'
RINFRAPR = 'Pre-FRACHECK exit?'
RINRLIPR = 'Pre-RACLST exit?'
RINRLISL = 'RACLST select exit?'
RINRDEPO = 'Post-RACDEF exit?'
RINAUUSR = 'User class audit?'
RINAUGRP = 'Group class audit?'
RINAUDSN = 'Dataset class audit?'
RINAUTVL = 'TAPEVOL class audit?'
RINAUDV = 'DASD class audit?'
RINAUTRM = 'Terminal class audit?'
RINAUCCMD = 'Command violation audit?'
RINAPUSPE = 'SPECIAL user audit?'
RINAUOPR = 'OPRATNS user audit?'
RINADELEV = 'Security level audit?'
RINACECM = 'ACEE compress exit?'
RINFSTPR = 'Pre-FASTAUTH exit?'
RINFSTPO = 'Post-FASTAUTH exit?'
RINTERM = 'Term. auth. check active?'
RINTRMNO = 'Undef. term. UACC=NONE?'
RINRLDSN = 'REALDSN active?'
RINXMBAL = 'XBMBATCHALL active?'
RINRLYVF = 'EARLYVERIFY active?'
RINBTCAL = 'BATCHALLRACF active?'
RINFRAPM = 'Post-FRACHECK exit?'
RINPWDIN = 'Max password interval'
RINSLDS = 'Single level datasetname'
RINTPDSN = 'TAPEDSN active?'
RINPRTLAL = 'PROTECTALL active?'
RINPRTAW = 'PROTECTALL warning active?'
RINERASE = 'ERASE-ON-SCRATCH active?'
RINERSLV = 'ERASE-ON-SCR. by sec. level active?'
RINERSAL = 'ERASE-ON-SCR. for all datasets active?'
RINEGN = 'EGN active?'
RINWHNPR = 'When-program ACTIVE?'
RINRETPD = 'System rentention period'
RINLVERS = 'Sec. level for ERASE-ON-SCRATCH'
RINLVAUD = 'Sec. level for audit'
RINSCCLCT = 'SECLABELCONTROL active?'
RINCATDS = 'CATDSNS active?'
RINMLOQ = 'MLQUIET active?'
RINMLSTB = 'MLSTABLE active?'
RINMLTS = 'MLS active?'
RINMLACT = 'MLACTIVE active?'
RINGENOW = 'GENERICOWNER active?'
RINSCLAU = 'SECLABELAUDIT active?'
RINSESIN = 'LU-verification interval'
RINNUNAM = 'NJE userid'
RINNUND = 'Undefineduser userid'
RINCMPAT = 'COMPATMODE active?'
RINCDSFA = 'CATDSN fail. active?'
RINMLSFA = 'MLS fail. active?'
RINMACFA = 'MLACTIVE fail. active?'
RINAPLAU = 'APPLAUDIT active?'
RINPRMLG = 'Def. prim. lang.'
RINNSDLG = 'Def. sec. lang.'

OUTPUT RACF.RACFINIT;

END;

%MEND RACFINIT:

MACRO CLASNAME(REQ=);
%LET REQ = %UPCASE(&REQ);
%IF &REQ = DEFINE %THEN
%DO;
   %PUT Including variables from CLASNAME extension;
   RACF.CLASNAME (KEEP=%SMFHDR
      RICRSV1
      RICTIME

%END;
%IF &REQ = EXTRACT %THEN
  %DO;
    %PUT Including datadefinition for CLASNAME extension;
    WHEN('CLASNAME') DO;
      INPUT RICRSV1 $ 10-17
        @19
       RICTIME TIME8.
        @28
      RICDATE YYMMDD10.
      RICSMFID $ 39-42
      RICCLASN $ 44-51
      RICSTATS $ 53-56
      RICAUDIT $ 58-61
      RICACTIV $ 63-66
      RICGNRIC $ 68-71
      RICGNCMD $ 73-76
      RICGLBAL $ 78-81
      RICRCLST $ 83-86
      RICGNLST $ 88-91
      RICLOGOP $ 93-100
    %END;
    FORMAT RICTIME TIME8.
    RICDATE YYMMDD10.
    :
    LABEL RICRSV1 = 'Reserved'
    RICTIME = 'Time written'
    RICDATE = 'Date written'
    RICSMFID = 'SMF id'
    RICCLASN = 'Class name'
    RICSTATS = 'Stats collected?'
    RICAUDIT = 'Audited?'
    RICACTIV = 'Active?'
    RICGNRIC = 'Generic profiles?'
    RICGNCMD = 'Generic command?'
    RICGLBAL = 'Global access?'
    RICRCLST = 'May be RACLSTed?'
  %END;
RICGNLST = 'May be GENLISTed?'
RICLOGOP = 'LOGOPTIONS'
;
   OUTPUT RACF.CLASNAME;
END;
%END;
%MEND CLASNAME:
./        ADD   LIST=ALL,NAME=DSA
%MACRO DSAF(REQ=);
   %LET REQ = %UPCASE(&REQ);
   %IF &REQ = DEFINE %THEN
   %DO:
         %PUT Including variables from DSAF extension;
         RACF.DSAF (KEEP=%SMFHDR
                   DSARSV1
                   DSATIME
                   DSADATE
                   DSASMFID
                   DSASECCLL
                   DSAVIOL
                   DSAUNDEF
                   DSAWARN
                   DSAUSRID
                   DSAGRPID
                   DSAANORM
                   DSAAASPEC
                   DSAAOPER
                   DSAAAUDT
                   DSAAEXIT
                   DSAAFSFT
                   DSAABYPS
                   DSAATRST
                   DSALCLAS
                   DSALUSER
                   DSALSPEC
                   DSALACCS
                   DSALRACI
                   DSALALWS
                   DSALCMDV
                   DSALGLBL
                   DSATRMLV
                   DSABKOFI
                   DSAPRFSM
                   DSADETERM
                   DSAJOBNM
                   DSARDTME
                   DSARDTTDE
                   DSAUSRID
                   DSALGLVL
                   DSALGOPT
DSALGSCL
DSALGCMP
DSAUSECL
DSADSNAM

}%END;
%IF &REQ = EXTRACT %THEN
%DO;
%PUT Including datadefinition for DSAF extension;
WHEN('DSAF') DO;
 INPUT DSARSVI $ 10-17
@19
DSATIME  TIME8.
@28
 DSADATE  YYMMDD10.
 DSASMFID $ 39-42
 DSASECLL $ 44-59
 DSAVIOL $ 61-64
 DSAUUNDEF $ 66-69
 DSAWARN $ 71-74
 DSASUSRID $ 76-83
 DSAGRPID $ 85-92
 DSAANORM $ 94-97
 DSAASPEC $ 99-102
 DSAAOOPER $ 104-107
 DSAAAAUDT $ 109-112
 DSAAEEXIT $ 114-117
 DSAAFSFT $ 119-122
 DSAABYPS $ 124-127
 DSAATRST $ 129-132
 DSALCLAS $ 134-137
 DSALUSER $ 139-142
 DSALSPEC $ 144-147
 DSALACCS $ 149-152
 DSALRACI $ 154-157
 DSALALWS $ 159-162
 DSALCMDV $ 164-167
 DSALGLBL $ 169-172
 DSATRMLV  174-176
 DSABKOF $ 178-181
 DSAPRFSM $ 183-186
 DSATERM $ 188-195
 DSAJOBNM $ 197-204
 @206
 DSARDME TIME8.
 @215
 DSARDDTE YYMMDD10.
 DSASUSRID $ 226-233
 DSALGLVL $ 235-238
 DSALGOPT $ 240-243

© 1999. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.
DSALGSCL  $  245-248
DSALGCM$  $  250-253
DSAUSECL $  255-262
DSADSNAM $  264-307

;  
FORMAT DSATIME  TIME8.
DSADATE  YYMMDD10.
DSARDTME TIME8.
DSARDDTE YYMMDD10.

;  
LABEL DSARSV1  = 'Reserved'
DSATIME  = 'Time written'
DSADATE  = 'Date written'
DSASMFID = 'SMF id'
DSASECLL = 'SECLABEL link'
DSAVIOL = 'Violation?'
DSAUNDEF = 'Undefined user?'
DSAWARN = 'Warning?'
DSAUSRID = 'Userid'
DSAGRPID = 'Groupid'
DSAANORM = 'Auth. normal?'
DAAASPEC = 'Auth. SPECIAL?'
DAAOPER = 'Auth. OPERATNS?'
DSAAAUDT = 'Auth. AUDITOR?'
DAAEXIT = 'Auth. exit?'
DAAFSFT = 'Auth. failsoft?'
DAAABYPS = 'Auth. BYPASS?'
DAAATRST = 'Auth. trusted?'
DSALCLAS = 'Class audit?'
DSALUSER = 'User audit?'
DALSPEC = 'SPECIAL audit?'
DSALACCS = 'Profile audit?'
DASLACCI = 'RACINIT fail?'
DSALALWS = 'Always audit?'
DSALCMDV = 'CMDVIOL audit?'
DSALGLBL = 'GLOBALAUDIT?'
DSATRMLV = 'Terminal level'
DSABKOFCL = 'Backout failure?'
DSAPRFSM = 'Same profile?'
DSATERM = 'Terminal?'
DASAEXIT = 'SMF exit userid'
DSALGCM$ = 'SECLABEL audit?'
DALSPEC = 'COMPATMODE?'
DSAGRPID = 'SECLABEL'
DSADSNAM = 'Dataset affected'
; OUTPUT RACF.DSAF;
END;
%END;
%MEND DSAF;
./     ENDUP
/*
//

© Reserved 1999
An ISPF dialog to manage catalog aliases

Maintaining catalog aliases is an important responsibility. When a new alias is requested, it needs to be defined along with the related RACF profiles. For example, new TSO users require a RACF user-id and dataset profile. New software products to be installed would require a RACF group and dataset profile.

Developers who need application aliases, however, require special handling. At our site, application aliases need a project hierarchy (similar to SCLM) for the application’s datasets.

The hierarchy consists of a RACF group and dataset profiles that follow our naming convention for an application (see Figure 1). Using this convention, RACF dataset profiles with the names alias.PROD.**, alias.TEST.**, alias.BKUP.**, and alias.VALT.** would be created. In addition, the list of application aliases needs to be maintained for promotion verification. Two edit macros (EDMAC01 and EDMAC02) update this list, but the dataset(member) must be created and initialized with one record.

![Figure 1: Application development hierarchy](image)
Each level of the hierarchy (TEST, PROD, BKUP, VALT) requires a different access list. To facilitate this process, I have written the following ISPF dialog.

SYSTEM REQUIREMENTS AND RESTRICTIONS
This dialog was developed using ISPF Version 3, and is currently running under ISPF Version 4.2. The environment this is currently running on is OS/390, TSO/E, and RACF 2.2. In the CLISTs, the user creates a dataset and executes it. The TSO PROFILE PREFIX is set to the user’s ID, and then removed. If the standard is a prefix of the user’s ID, the PROFILE command can be removed. There are no other restrictions for this dialog.

The dollar sign ($) command was taken from the article in MVS Update, May 1992, written by Adrian Gallagher. Although it is not required to run this dialog, it was a nice way to display the TSO command output. You should remove the dollar sign to run without this code, or replace it with your site’s command. Editor’s note: the code from the article is available from the Xephon Web site at http://www.xephon.com.

OTHER INTERFACES
This dialog uses the RACF Panel Driver Interface (PDI). The purpose of this interface is to bypass unnecessary RACF panels. The PDI uses program ICHSPF03. Before calling the module, three variables can be set. For more information on this interface, refer to the RACF Macros and Interfaces Guide.

THE ALIAS DELETE/DEFINE RACF DIALOG
A brief description follows of the components of this dialog. Some sections may include ways to improve on the design or ways to customize the design to meet your company’s requirements. The components are divided into CLISTs and panels.

CLISTS
CLIST01 controls the applications. To invoke this application, set up
a command option on any primary menu panel as follows:

```
ARF,'CMD(%CLIST01) NEWAPPL(ARF)'
```

The NEWAPPL option will create members in your ISPF profile dataset that will store information from session to session. As supplied, LIBDEF statements direct the application to specific panel and message libraries. For testing purposes, place all the members in one library and concatenate the library to the ISPCLIB DD. Modify the LIBDEFs to point to this library.

CLIST02 defines the alias and the RACF profiles. This is where the dialog can be customized for any special processing at your company. Figure 2 shows the process that is in place. Any changes made to the process must also be made in CLIST03 and CLIST04. CLIST03 deletes the alias and the RACF profiles. CLIST04 displays the alias and the RACF profiles. EDMAC01 is for application aliases – it will insert the alias name into a dataset. EDMAC02 removes the alias name from a dataset (application alias only). MESSG00 is the panel messages member.

PANELS

PANEL01 is the main panel input panel. STDERR is the error panel. HLPNL01-06 are the help panels.

CLIST01

```
/* START OF CLIST01 */
/*****************************************************************************/
/* LIB: DATASET.ISPCLIB(CLIST01) */
/* GDE: THIS IS THE CONTROLLING CLIST. THIS CLIST WILL INVOKE */
/*      THE OTHER CLISTS. */
/* DOC: */
/* */
/* TO INVOKE THIS CLIST FROM A PANEL SET UP THE OPTION AS FOLLOWS: */
/* */
/* ARF,'CMD(%ALSRC01) NEWAPPL(ARF)' */
/* */
/* THE NEWAPPL OPTION WILL CREATE MEMBERS IN YOUR ISPF PROFILE */
/* DATASET THAT WILL STORE INFORMATION FROM SESSION TO SESSION. */
/* */
/*****************************************************************************/
PROC Ø DEBUG
IF &DEBUG = DEBUG THEN +
```
Figure 2: CLIST02 process
CONTROL MAIN LIST CONLIST SYMLIST
ELSE +
  CONTROL MAIN NOLIST NOCONLIST NOSYMLIST NOMSG

ISPEXEC LIBDEF ISPPLIB DATASET ID('DATASET.ISPCLIB')
ISPEXEC LIBDEF ISPLIB DATASET ID('DATASET.ISPCLIB')
SET ERRRC = Ø
DO WHILE(&ERRRC < 8)
  SET SYSOUTTRAP = 1000
  SET ZCMD = &STR( )
  ISPEXEC DISPLAY PANEL(PANELØ1) MSG(MESSGØØØ)
  SET ERRRC = &LASTCC
  IF &ERRRC = 8 THEN +
    DO
      ISPEXEC VERASE (ZCMD,ALSRVØ1,TYPE,ALSCREL)
      EXIT CODE(Ø)
    END
  IF &ERRRC ≠ Ø THEN +
    DO
      SET ERRSERV = &STR(DISPLAY)
      SET ERRFUNC = &STR(ATTEMPT TO DISPLAY PANEL(PANELØ1))
      ISPEXEC DISPLAY PANEL(STDERR)
      RETURN
    END
  ISPEXEC VPUT (ALSRVØ1,ALSCREL,ALSMCAT1,ALSMCAT2,ALSMCAT3, +
               ALSMCAT4,TYPE,ALSJCLCK)
  SET SYSOUTLINE = Ø
  SELECT &ZCMD
    WHEN (&STR( )) ISPEXEC SELECT CMD(CLISTØ4)
    WHEN (A) ISPEXEC SELECT CMD(CLISTØ2)
    WHEN (D) ISPEXEC SELECT CMD(CLISTØ3)
    OTHERWISE ISPEXEC SETMSG MSG(MESSGØØ8)
  END
END
/* END OF CLISTØ1 */

CLIST02
/* START OF CLISTØ2 */
*******************************************************************************/
/* LIB: DATASET.ISPCLIB(CLISTØ2) */
/* GDE: DEFINE ALIAS AND RACF INFO */
/* DOC: DEFINE THE ALIAS TO THE CATALOG WITH PROPER RACF PROFILE */
/* INFORMATION. */
/****
*******************************************************************************/
PROC Ø DEBUG
IF &DEBUG = DEBUG THEN +
  CONTROL MAIN LIST CONLIST SYMLIST
ELSE +
  CONTROL MAIN NOLIST NOCONLIST NOSYMLIST NOMSG
SET SYSOUTTRAP = 1000
ISPEXEC VGET (ALSRV01,ALSMCAT1,ALSMCAT2,ALSMCAT3,ALSMCAT4, +
   TYPE,ALSCREL,ALSJCLCK)
DO J = 1 TO 4
   SET MCAT = &&ALSMCAT&J
   IF &MCAT ¬& STR( ) THEN +
      DO
         SET SYSOUTLINE = Ø
         LISTCAT ENT(&ALSRV01) CATALOG(&MCAT)
         SET RC = &LASTCC
         IF &RC ¬Ø THEN +
            DEFINE ALIAS ( +
               NAME(&ALSRV01) +
               RELATE(&ALSCREL) +
            ) +
            CATALOG(&MCAT)
         ELSE +
            DO
               ISPEXEC SETMSG MSG(MESSG001)
            END
         END
      ELSE +
         DO
            ISPEXEC SETMSG MSG(MESSG001)
            $ LISTCAT ENT(&ALSRV01) CATALOG(&MCAT) ALL
         END
      END
   END
IF &STR(&TYPE) = &STR(U) THEN +
   DO
      SET SYSOUTLINE = Ø
      LISTUSER &ALSRV01
      SET RC = &LASTCC
      IF &RC ¬Ø THEN +
         SET ICHFUNCT = ADD
         SET ICHRESCL = USER
         SET ICHRESNM = &ALSRV01
         ISPEXEC VPUT (ICHFUNCT ICHRESCL ICHRESNM)
         ISPEXEC SELECT PGM(ICHSPF03)
         SET RC = &LASTCC
         IF &RC ¬Ø THEN +
            EXIT CODE(&RC)
         END
      ELSE +
         DO
            ISPEXEC SETMSG MSG(MESSG001)
            $ LISTUSER &ALSRV01
         END
   END
IF &TYPE = &STR(S) OR &TYPE = &STR(A) THEN +
    DO
        SET SYSOUTLINE = Ø
        LISTGRP &ALSRVØ1
        SET RC = &LASTCC
        IF &RC ¬= Ø THEN +
            DO
                SET ICHFUNCT = ADD
                SET ICHRESCL = GROUP
                SET ICHRESNM = &ALSRVØ1
                ISPEXEC VPUT (ICHFUNCT ICHRESCL ICHRESNM)
                ISPEXEC SELECT PGM(ICHSPFØ3)
                SET RC = &LASTCC
                IF &RC ¬= Ø THEN +
                    EXIT CODE(&RC)
            END
        ELSE +
            DO
                ISPEXEC SETMSG MSG(MESSGØØ1)
            END
        END
    ELSE +
        DO
            ISPEXEC SETMSG MSG(MESSGØØ1)
        END
    END
    IF &STR(&TYPE) = &STR(A) THEN +
        DO
            /**************************************************************/
            /* THE FOLLOWING IS AN EXAMPLE OF HOW YOU CAN BUILD A NAMING  */
            /* CONVENTION INTO THE ALIAS DEFINITION. THE NAMING CONVEN- */
            /* TION CAN APPLY TO APPLICATION DATASETS (TEST & PROD, ETC) */
            /* OR SCLM PROJECT DATASETS (SOURCE, LOAD, LIST, ETC). THE */
            /* RELATED SECURITY FOR THESE DATASETS IS IMPLEMENTED DURING */
            /* THE CREATION OF THE RACF PROFILE. THE ACCESS LISTS CAN */
            /* BE UPDATED LATER, BUT CAN EASILY BE COPIED FROM EXISTING */
            /* RACF PROFILES NOW. */
            /**************************************************************/
            /* SET THE NAMING CONVENTION VARIABLES */
            SET APPLNC1 = BKUP
            SET APPLNC2 = PROD
            SET APPLNC3 = TEST
            SET APPLNC4 = VALT
            /* SET THE LOOP COUNT BASED ON THE VAR */
            DO I = 1 TO 4
                SET SYSOUTLINE = Ø
                SET SLQ = &&APPLNC&I
                LISTDSD DATASET(&STR(&STR(&ALSRVØ1).&STR(&SLQ).**))
                SET RC = &LASTCC
                IF &RC ¬= Ø THEN +
                    DO
                        SET ICHFUNCT = ADD
                        SET ICHRESCL = DATASET
                        SET ICHRESNM = &STR(&STR(&ALSRVØ1).&STR(&SLQ).**)
                        ISPEXEC VPUT (ICHFUNCT ICHRESCL ICHRESNM)
ISPEXEC SELECT PGM(ICHSPFØ3)
SET RC = &LASTCC
IF &RC ≠ Ø THEN +
    EXIT CODE(&RC)
END
ELSE +
    DO
        ISPEXEC SETMSG MSG(MESSGØØ1)
        $ LISTDSD DATASET(&STR(&STR(&ALSRVØ1).&STR(&SLQ).**)) ALL DSNS
        END
        ISPEXEC VPUT ALSRVØ1
        ISPEXEC EDIT DATASET(&STR(&ALSJCLCK)) MACRO(EDMACØ1)
        ISPEXEC VPUT ALSRVØ1
        ISPEXEC SETMSG MSG(MESSGØØ9)
        ISPEXEC BROWSE DATASET(&STR(&ALSJCLCK))
    END
    SET SYSOUTLINE = Ø
    LISTDSD DATASET(&STR(&STR(&ALSRVØ1).**))
    SET RC = &LASTCC
    IF &RC ≠ Ø THEN +
        DO
            SET ICHFUNCT = ADD
            SET ICHRRESCL = DATASET
            SET ICHRRESNM = &STR(&STR(&ALSRVØ1).**)
            ISPEXEC VPUT (ICHFUNCT ICHRRESCL ICHRRESNM)
            ISPEXEC SELECT PGM(ICHSPFØ3)
            SET RC = &LASTCC
            IF &RC ≠ Ø THEN +
                EXIT CODE(&RC)
            END
        END
        ELSE +
            DO
                ISPEXEC SETMSG MSG(MESSGØØ1)
                $ LISTDSD DATASET(&STR(&STR(&ALSRVØ1).**)) ALL DSNS
                END
                SET ANAME = &ALSRVØ1&STR(        )
                SET ANAME = &SUBSTR(1:8,&ANAME)
                /* DISPLAY THE FOLLOWING MESSAGE TO REMIND THE USER OF ANY OTHER */
                /* RELATED ACTIVITIES.                                           */
                WRITE
                WRITE
                WRITE
                WRITE
                WRITE *************************************************
                WRITE **                                             **
                WRITE **    R E M I N D E R :                        **
                WRITE **                                             **
                WRITE **  SEE YOUR SMS ADMINISTRATOR TO ADD THIS     **
                WRITE **  NEW ALIAS (&ANAME) TO THE ACS ROUTINE.     **
                WRITE **                                             **
CLIST03

/* START OF CLIST03 */
*******************************************************************************/
/* LIB: DATASET.ISPCLIB(CLSTØ3) */
/* GDE: DELETE ALIAS AND RACF INFO */
/* DOC: DELETE THE ALIAS FROM THE CATALOG WITH PROPER RACF PROFILE */
/* INFORMATION DELETED ALSO. */
/* */
/*******************************************************************************/
PROC Ø DEBUG
IF &DEBUG = DEBUG THEN +
  CONTROL MAIN LIST CONLIST SYMLIST
ELSE +
  CONTROL MAIN NOLIST NOCONLIST NOSYMLIST NOMSG
SET SYSOUTTRAP = 1000
ISPEXEC VGET (ALSRVØ1,ALSMCAT1,ALSMCAT2,ALSMCAT3,ALSMCAT4, +
   TYPE,ALSCREL,ALSJCLCK)
SET SYSOUTLINE = Ø
LISTCAT LVL(&ALSRVØ1)
SET RC = &LASTCC
IF &RC = Ø THEN +
  DO
    ISPEXEC SETMSG MSG(MESSGØØ3)
    $ LISTCAT LVL(&ALSRVØ1)
    EXIT CODE(&RC)
  END
DO J = 1 TO 4
SET MCAT = &&ALSMCAT&J
IF &MCAT ¬= &STR( ) THEN +
  DO
    SET SYSOUTLINE = Ø
    LISTCAT ENT(&ALSRVØ1) CATALOG(&MCAT)
    SET RC = &LASTCC
    IF &RC = Ø THEN +
      DELETE +
      (&ALSRVØ1) +
      ALIAS +
      CATALOG(&MCAT)
    END
  END
SET SYSOUTLINE = Ø
SEARCH CLASS(DATASET) MASK(&ALSRVØ1..)
SET RC = &LASTCC
IF &RC = Ø THEN +
  $ SEARCH CLASS(DATASET) MASK(&ALSrv01..)
ELSE +
  DO
    PROFILE PREFIX(&SYSUID)
    SET SYSOUTLINE = Ø
    SEARCH CLASS(DATASET) MASK(&ALSrv01..) CLIST('DD ')
    EXEC EXEC.RACF.CLIST
    PROFILE NOPREFIX
  END
IF &STR(&TYPE) = &STR(U) THEN +
  DO
    SET SYSOUTLINE = Ø
    LISTUSER &ALSrv01
    SET RC = &LASTCC
    IF &RC = Ø THEN +
      DO
        SET ICHFUNCT = DEL
        SET ICHRESCL = USER
        SET ICHRESNM = &ALSrv01
        ISPEXEC VPUT (ICHFUNCT ICHRESCL ICHRESNM)
        SET RC = &LASTCC
        IF &RC = Ø THEN +
          EXIT CODE(&RC)
          ISPEXEC SELECT PGM(ICHSPF03)
          SET RC = &LASTCC
          IF &RC = Ø THEN +
            EXIT CODE(&RC)
            END
      END
    END
ELSE +
  DO
    SET SYSOUTLINE = Ø
    LISTGRP &ALSrv01 NORACF DFP
    SET RC = &LASTCC
    IF &RC = Ø THEN +
      DO
        SET ICHFUNCT = DEL
        SET ICHRESCL = GROUP
        SET ICHRESNM = &ALSrv01
        ISPEXEC VPUT (ICHFUNCT ICHRESCL ICHRESNM)
        SET RC = &LASTCC
        IF &RC = Ø THEN +
          EXIT CODE(&RC)
          ISPEXEC SELECT PGM(ICHSPF03)
          SET RC = &LASTCC
          IF &RC = Ø THEN +
            EXIT CODE(&RC)
            END
      END
      IF &STR(&TYPE) = &STR(A) THEN +
        DO
    END

A RACFPROF ISPF interface – part 2

This month we conclude the program designed to encourage user accountability and reduce the RACF administration of dataset profiles/groups by the RACF administrator.

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

---

Steven Windhausen
Technical Specialist
County of Onondaga (USA)  © Xephon 1999
+ UserId.:#Userid +
+
+ Access.:#z+Alter (Allocate & Delete)
+    #z+Update (Non-VSAM Update)
+    #z+Control (VSAM Update)
+    #z+Read
+    #z+Execute
+    #z+None
+
+ &RacfMsg
)INIT
    &DATEX = '&ZMONTH/&ZDAY/&ZYEAR  &ZTIME PST'
 .CURSOR = UserId
 .ZVARS = '(Alter Update Control Read Execute None)'
)REINIT
)PROC
)END

RACFPOP2 PANEL
)
)ATTR
    _ TYPE(input) INTENS(HIGH) CAPS(ON) just(left) pad(' ')
    # TYPE(input) INTENS(HIGH) CAPS(ON) just(left) pad(' _')
    + TYPE(TEXT) INTENS(LOW) SKIP(ON) CAPS(ON)
    % TYPE(TEXT) INTENS(HIGH) SKIP(ON) CAPS(ON)
)BODY WIDTH(60) EXPAND(/) window(60,9)
+ /-/%RACF Dataset Profile Update   &DATEX +/-/
+
+ Modify the Dataset Profile, then press the ENTER key.
+
+ Profile:#PermDsn +
+
+ Model Like Profile:#ModelDsn +
+
+ &RacfMsg
)INIT
    &DATEX = '&ZMONTH/&ZDAY/&ZYEAR  &ZTIME PST'
 .CURSOR = PermDsn
)REINIT
)PROC
)END

RACFPOP3 PANEL
)
)ATTR
    _ TYPE(input) INTENS(HIGH) CAPS(ON) just(left) pad(' ')
    # TYPE(input) INTENS(HIGH) CAPS(ON) just(left) pad(' _')
    + TYPE(TEXT) INTENS(LOW) SKIP(ON) CAPS(ON)
RACFPPOP4 PANEL

)ATTR
   _ TYPE(input) INTENS(HIGH) CAPS(ON) just(left) pad(' ')
   # TYPE(input) INTENS(HIGH) CAPS(ON) just(left) pad('_')
   + TYPE(TEXT) INTENS(LOW) SKIP(ON) CAPS(ON)
   % TYPE(TEXT) INTENS(HIGH) SKIP(ON) CAPS(ON)
)BODY WIDTH(60) EXPAND(/) window(60,10)
+ /-/RACF Connect Userid +/-/
  + Update the User-id, then press the ENTER key.
  + Group..: &ConGroup
  + UserId.:#ConUser +
  +
  + &RacfMsg
)INIT
.CURSOR = ConUser
)REINIT
)PROC
)END

RACFPPOP5 PANEL

)ATTR
   _ TYPE(input) INTENS(HIGH) CAPS(ON) just(left) pad(' ')
   # TYPE(input) INTENS(HIGH) CAPS(ON) just(left) pad('_')
   + TYPE(TEXT) INTENS(LOW) SKIP(ON) CAPS(ON)
   % TYPE(TEXT) INTENS(HIGH) SKIP(ON) CAPS(ON)
+ Update the User-id field, then press the ENTER key.
+ % Pressing ENTER will remove access from the profile.
+ + Profile:%&PermDsn
+ + UserId.:#Userid +
+ + &RacfMsg
)INIT
   &DATEX = '&ZMONTH/&ZDAY/&ZYEAR  &ZTIME PST'
   .CURSOR = UserId
)REINIT
)PROC
)END

RACFPOP6 PANEL

)ATTR
   _ TYPE(input) INTENS(HIGH) CAPS(ON) just(left) pad( ' ' )
   # TYPE(input) INTENS(HIGH) CAPS(ON) just(left) pad( '_' )
   + TYPE(TEXT) INTENS(LOW) SKIP(ON) CAPS(ON)
   % TYPE(TEXT) INTENS(HIGH) SKIP(ON) CAPS(ON)
)BODY WIDTH(6Ø) EXPAND(///) window(6Ø,11)
+ /-/ RACF Access Update &DATEX +-/+
+ + Update the User-id field, then press the ENTER key.
+ + % Pressing ENTER will remove access from the profile.
+ + Profile:%&PermDsn
+ + UserId.:#Userid +
+ + &RacfMsg
)INIT
   &DATEX = '&ZMONTH/&ZDAY/&ZYEAR  &ZTIME PST'
   .CURSOR = UserId
)REINIT
)PROC
)END

RACFPOP7 PANEL

)ATTR
   _ TYPE(input) INTENS(HIGH) CAPS(ON) just(left) pad( ' ' )
   # TYPE(input) INTENS(HIGH) CAPS(ON) just(left) pad( '_' )
   + TYPE(TEXT) INTENS(LOW) SKIP(ON) CAPS(ON)
   % TYPE(TEXT) INTENS(HIGH) SKIP(ON) CAPS(ON)
   $ TYPE(TEXT) INTENS(HIGH) COLOR(TURQ)
Enter the Group name, then press the ENTER key.

Group: #Group

SupGroup: #Supgroup

$'You may leave blank...'

&RacfMsg

&DATEX = '&ZMONTH/&ZDAY/&ZYEAR &ZTIME PST'

.CURSOR = Group

%INFO+List Profile Creation Dates
%ADDG+Add RACF groups
%AUTH+List Userid(s) and Access for Profile
%DSNS+List Catalogued Dataset Names Affected by Profile
%ADD +Add Dataset Profile
%DEL +Delete Dataset Profile
%PE +Permit Access
%REM +Remove Access from Dataset Profile

+ Dataset Profile(s)

+Dataset Profile Mask:_Mask

%COMMAND ===> _ZCMD

+Dataset Profile Mask:_Mask

+ %INFO+List Profile Creation Dates
%ADDG+Add RACF groups
%AUTH+List Userid(s) and Access for Profile
%DSNS+List Catalogued Dataset Names Affected by Profile
%ADD +Add Dataset Profile
%DEL +Delete Dataset Profile
%PE +Permit Access
%REM +Remove Access from Dataset Profile

+ Dataset Profile(s)

+Dataset Profile Mask:_Mask

+ %INFO+List Profile Creation Dates
%ADDG+Add RACF groups
%AUTH+List Userid(s) and Access for Profile
%DSNS+List Catalogued Dataset Names Affected by Profile
%ADD +Add Dataset Profile
%DEL +Delete Dataset Profile
%PE +Permit Access
%REM +Remove Access from Dataset Profile

+ Dataset Profile(s)

+Dataset Profile Mask:_Mask

+ %INFO+List Profile Creation Dates
%ADDG+Add RACF groups
%AUTH+List Userid(s) and Access for Profile
%DSNS+List Catalogued Dataset Names Affected by Profile
%ADD +Add Dataset Profile
%DEL +Delete Dataset Profile
%PE +Permit Access
%REM +Remove Access from Dataset Profile

+ Dataset Profile(s)
CURSOR = ZCMD
PROC
VER (&LCMD, LIST, AUTH, DSNS, ADD, DEL, PE, REM, INFO, ADDG)
VPUT (AMT) PROFILE
END

RACFLSTU PANEL

)ATTR
+ TYPE(TEXT) INTENS(LOW) SKIP(ON) CAPS(ON)
% TYPE(TEXT) INTENS(HIGH) SKIP(ON) CAPS(ON)
_ TYPE(input) INTENS(HIGH) CAPS(ON) just(left) pad(' ')
# TYPE(input) INTENS(HIGH) CAPS(ON) just(left) pad('_')
@ TYPE(output) INTENS(low)
)BODY WIDTH(&ZSCREENW) EXPAND(/)
%-/%RACF User List %/-/
%COMMAND ===> _ZCMD %SCROLL
====>_AMT +
+UserId . . . . .:%&LstuUser
+Name . . . . .:%&LstuName
+Default-Group. .:%&LstuGrp
+
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
)MODEL
#z @lstuline
)INIT
.CURSOR = ZCMD
.ZVARS = '(LCMD)'
&ZCMD = ' '
)REINIT
.CURSOR = ZCMD
PROC
VPUT (AMT) PROFILE
)END

RACFLSTG PANEL

)ATTR
+ TYPE(TEXT) INTENS(LOW) SKIP(ON) CAPS(ON)
% TYPE(TEXT) INTENS(HIGH) SKIP(ON) CAPS(ON)
_ TYPE(input) INTENS(HIGH) CAPS(ON) just(left) pad(' ')
# TYPE(input) INTENS(HIGH) CAPS(ON) just(left) pad('_')
@ TYPE(output) INTENS(low)
)BODY WIDTH(&ZSCREENW) EXPAND(/)
%-/%RACF Group List %/-/
%COMMAND ===> _ZCMD %SCROLL
====>_AMT +
+
\begin{verbatim}
+Group. . . . :%&LstgGrp
+Superior Group :%&LstgSgrp
+
+%CON +Connect Userid TO Group
+%REM +Remove Userid FROM Group
+
+    User(S)       Access       Access Count       Universal Access
+    - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

)MODEL

#z  @lstgline

)INIT

.CURSOR = ZCMD
.ZVARS = '(LCMD)'
&ZCMD = '

)REINIT

.CURSOR = ZCMD

)PROC

VER (&LCMD,LIST,CON,REM)
VPUT (AMT) PROFILE

)END

\end{verbatim}

\textbf{RACFLIST PANEL}

\begin{verbatim}
)

)ATTR

+  TYPE(TEXT) INTENS(LOW) SKIP(ON) CAPS(ON)
%  TYPE(TEXT) INTENS(HIGH) SKIP(ON) CAPS(ON)
  _  TYPE(input) INTENS(HIGH) CAPS(ON) just(left) pad(' ')
  #  TYPE(input) INTENS(HIGH) CAPS(ON) just(left) pad(' _ ')
    @  TYPE(output) INTENS(low)

)BODY WIDTH(&ZSCREENW) EXPAND(/)

%/-/%RACF User List %/-/

%COMMAND ===>_ZCMD                        %SCROLL

====>_AMT +

+

%PE  +Permit Access
+

+    Catalogued Dataset(s) Affected by Profile
+

)MODEL

#z  @listline

)INIT

.CURSOR = ZCMD
.ZVARS = '(LCMD)'
&ZCMD = '

)REINIT

.CURSOR = ZCMD

)PROC

VER (&LCMD,LIST,PE)
VPUT (AMT) PROFILE

)END

\end{verbatim}
RACFGUID PANEL

)ATTR
  + TYPE(TEXT) INTENS(LOW) SKIP(ON) CAPS(ON)
  % TYPE(TEXT) INTENS(HIGH) SKIP(ON) CAPS(ON)
  _ TYPE(input) INTENS(HIGH) CAPS(ON) just(left) pad(' ')  
  # TYPE(input) INTENS(HIGH) CAPS(ON) just(left) pad(' _')
  @ TYPE(output) INTENS(low)
)BODY WIDTH(&ZSCREENW) EXPAND(/)
%/-/%RACF UID List %/-/ 
%COMMAND ===>>_ZCMD %SCROLL
 ===>>_AMT +
 +
 +Group UID . . . .: _Mask
 +Superior Group. .: &GuidGrp
 +
 %LG +List Group Information
 %LU +List User Information
 %CON +Connect User-id TO Group
 %REM +Remove User-id FROM Group
 +
 +   Sub Group(s)
 +
)MODEL
#z  @guidline
)INIT
 .CURSOR = Mask
 .ZVARS = '((LCMD)'
 &ZCMD = '
)REINIT
 .CURSOR = Mask
)PROC
  VER (&LCMD,LIST, LG, LU, CON, REM)
  VPUT (AMT) PROFILE
)END

RACFINFO PANEL

)ATTR
  + TYPE(TEXT) INTENS(LOW) SKIP(ON) CAPS(ON)
  % TYPE(TEXT) INTENS(HIGH) SKIP(ON) CAPS(ON)
  _ TYPE(input) INTENS(HIGH) CAPS(ON) just(left) pad(' ')  
  # TYPE(input) INTENS(HIGH) CAPS(ON) just(left) pad(' _')
  @ TYPE(output) INTENS(low)
)BODY WIDTH(&ZSCREENW) EXPAND(/)
%/-/%RACF Profile Creation Info%/-/ 
%COMMAND ===>>_ZCMD %SCROLL
 ===>>_AMT +
 +
TSO command to maintain RACF user-data

RACF user-data is the term given to fields present in most RACF profiles that are provided for installation purposes. These should not be confused with the ‘installation data’ field. The user-data fields are repeat groups, that is multiple instances can exist and each has a ‘member’ name to distinguish it from others.

Although IBM has helpfully provided these fields in the database templates, they have not supplied any means to access them from the usual LU or ALU commands. The comprehensive TSO command processor shown below does this and will allow the installation to store any number of additional fields in RACF user profiles.

The uses for these are many-fold, from purely administrative such as staff number, to data used by other programs and extracted by those programs (using similar code to that shown below). The possibilities really are limitless because RACF is the best place to store user-related security information instead of creating other non-standard security databases.
The syntax and format of the command processor are explained in its comments, but essentially three types of call are provided: GET to retrieve fields, PUT to update fields, and DEL to delete them.

To aid the development of ISPF dialogs, the list of existing members can be retrieved with a GET call of LISTMEMS. Each field has a key value (or member name) and a data area up to 255 bytes.

Normal RACF command authority applies in that RACF specials or group specials can update users within their scope. Because this feature is so useful, it may be desired to extend this and therefore a FACILITY class profile is also tested so that additional access rights can determine who may update certain field names.

One example of use might be to store the name of the user’s session manager profile. The session manager would read this value (using Assembler code) when the user logs on. The network support team might be granted access to maintain this field on behalf of any user, as well as the usual RACF administrators.

Because no practical limit exists to the number of user data fields, RACF can be used freely to control and store information like this without the inevitable conflict that arises if the single, fixed ‘installation data’ field is used. INSTDATA can still be used because it is a completely separate field from user data.

The command shown below is best used behind an ISPF interface and is designed to be integrated into REXX or CLIST code. For that reason, the data retrieval is into a TSO variable field.

ZRACUSR

```
TITLE 'ZRACUSR - MAINTAIN RACF USER DATA'
*****************************************************************
* ZRACUSR - AUTHORIZED TSO COMMAND PROCESSOR
* ____________
* NOT REENTRANT, AC=1, ENTRY POINT ZRACUSR
* *
* MUST RUN AS A CP (NOT BE CALLED) AND MUST RUN AUTHORIZED SO
* BE FROM AN AUTHORIZED LIBRARY, LINKED AC=1, AND IN IKJTSO00.
* *
* UPDATES RACF USER-DATA FIELDS, ANY NUMBER OF USER-DATA MEMBERS
* CAN EXIST IN A USER PROFILE AND EACH HAS A NAME (KEY) AND A
* DATA FIELD (VARIABLE LENGTH TO 255 MAX)
```
TSO COMMAND PROCESSOR SYNTAX, OPERANDS IN ANY SEQUENCE, EG

EXAMPLES:

ZRACUSR USR(RID) KEY(LISTMEMS) ACT(GET) DAT(VARNAM)
ZRACUSR USR(RID) KEY(PRNTINFO) ACT(GET) DAT(VARNAM)
ZRACUSR USR(RID) KEY(PRNTINFO) ACT(PUT) LEN(33) DAT('XXXXX....')
ZRACUSR USR(RID) KEY(PRNTINFO) ACT(DEL)

QUOTES ARE ALLOWED TO DELIMIT ANY VALUE WITHIN BRACKETS AND MUST BE
USED IF THE VALUE ITSELF CONTAINS BRACKETS, OTHERWISE ARE OPTIONAL.

THIS PROGRAM USES A FACILITY CLASS RULE TO DETERMINE WHO CAN
UPDATE VARIOUS MEMBER NAMES, IE INST.USERDATA.KEYNAME.USERID
(WHERE KEYNAME AND USERID ARE VARIABLE)

ALSO SPECIALS AND GROUP SPECIALS WILL BE ALLOWED TO UPDATE USERS.
USERS CAN ALWAYS READ THEIR USRDATA AND ALWAYS UPDATE PRNTINFO.

KEYWORD VALUES:

USR IS THE USER TO BE ACCESSED, THIS IS NORMALLY THE SAME AS THE
CURRENT USER-ID (EG &SYSUID IN A CLIST), HOWEVER RACF SPECIAL
USERS ARE ALLOWED TO UPDATE ANY ENTRIES FOR ANY USER.

KEY IS THE NAME OF THE USRDATA MEMBER, USE EIGHT CHARACTERS

NOTE SPECIAL KEYNAMES ARE 'LISTMEMS' AND 'LISTALLD' WHICH
HAVE THE EFFECT OF RETURNING A LIST OF MEMBERS IN THE
USERDATA OR A LIST INCLUDING THE USERDATA AS WELL. THESE
ARE USED FOR BUILDING A DISPLAYABLE INDEX OF MEMBERS.

ACT IS THE ACTION WHICH MAY BE PUT, GET, OR DEL. PUT MEANS UPDATE
AND WRITES TO THE RACF DATABASE. GET MEANS RETRIEVE AND READS
THE RACF DATABASE. DEL MEANS DELETE AND ERASES ONLY THIS
USER-DATA ENTRY IN THE USERS PROFILE, NOT THE USER PROFILE.

DAT IS THE DATA TO BE STORED IN THE MEMBER OR VARIABLE NAME TO SET

THE VARIABLE NAME (TSO REXX/CLIST) MUST BE SIX CHARLS LONG.

LEN IS THE LENGTH OF DATA FOR THIS TYPE OF KEY (1-255 FOR PUT)

NOTES:

FOR GET ACTION THE LENGTH IS SET FROM THE DATA RETRIEVED AND THE
THE DAT OPERAND IS THE TSO CLIST/REXX VARIABLE NAME TO BE SET
FROM THE CURRENT CONTENTS FROM THE RACF DATABASE AND MUST BE
A SIX CHARACTER VARIABLE NAME (LENGTH OF NAME IS FIXED).
* FOR PUT ACTION THE DAT OPERAND IS THE ACTUAL DATA TO BE STORED
* AND QUOTES ARE NOT USED. LEN MUST BE SPECIFIED AND IF IT IS
* MORE THAN THE DATA PROVIDED THEN BLANK PADDING WILL OCCUR.
* 
* FOR DEL ACTION THE OPERANDS DAT AND LEN ARE NOT REQUIRED.
* 
* RACF PERFORMS THE ACCESS AND THE USER DOES NOT REQUIRE DATASET
* LEVEL ACCESS TO THE RACF DATABASE FOR ANY OF THESE FUNCTIONS.
* 
* RETURN CODES:
* 
* ALL EXCEPT ØØ AND Ø1 RESULT IN ERROR MESSAGES TO THE USER
* 
* ØØ - OPERATION SUCCESSFUL
* Ø1 - GET REQUEST FOUND NO PREVIOUS DATA
* Ø5 - USERID NOT DEFINED TO RACF
* Ø6 - TSO VARIABLE UPDATE FAILED
* Ø7 - PARAMETER WAS MISSING
* Ø8 - DELIMITER IS MISSING
* Ø9 - NOT APF AUTHORIZED
* 10 - NO RACF ACEE FOUND
* 11 - RACF ICHEINTY MACRO FAILED
* 12 - RACF ICHEACTN PARMS WRONG
* 13 - INVALID KEY NAME GIVEN
* 14 - NO PARM ENTERED
* 15 - INVALID LENGTH
* 16 - RACF DENIED ACCESS TO USERDATA
* 17 - INVALID ACTION
* 
* COMMENTS:
* 
* THIS COMMAND PROCESSOR ALLOWS TSO-BASED MAINTAINANCE OF THE
* USER-DATA FIELDS IN THE RACF DATABASE. BE WARNED RACF USER-DATA
* FIELDS ARE NOT SIMPLE BECAUSE THEY ARE PART OF A REPEAT GROUP.
* MULTIPLE RACF USER-DATA MEMBERS CAN EXIST AND ARE NAMED BY KEY.
* YOU CANNOT UPDATE A MEMBER OF A REPEAT GROUP BUT ONLY DELETE
* AND ADD THEM. THIS CODE LOCATES THE EXISTING ENTRY IF ANY AND
* THEN ALLOWS REPLACEMENT (DEL/ADD) OF THE CONTENTS. THE NORMAL
* USER-ID IS THE CURRENT RACF USER-ID BUT IF THIS IS A SPECIAL THEN
* THEY MAY MAINTAIN OTHER USERS' DATA. THE PROGRAM ITSELF CAN
* UPDATE ANYONE SINCE ICHEINTY ONLY CHECKS FOR APF AUTHORIZATION.
* 
* SEE THE RACF TEMPLATES FOR INFORMATION ON USER FIELDS. NOTE THAT
* USER-DATA IS A COMBINED FIELD CONTAINING A REPEAT GROUP MEMBER,
* WHEREAS USRDATA IS ONE OF THE MEMBER FIELDS ALONG WITH USRNAME
* AND USRFLG, USRCNT IS THE NUMBER OF REPEAT GROUP MEMBERS IN THE
* USER PROFILE AND IS MAINTAINED BY RACF AUTOMATICALLY. TO DELETE
* A MEMBER OF THE REPEAT GROUP SIMPLY DELETE ONE OF THE FIELDS AND
* THE ENTIRE MEMBER IS DELETED. TO DELETE ALL THE MEMBERS AT ONCE
* YOU CAN DELETE THE USRCNT FIELD WITH GROUP=YES SPECIFIED.

*****************************************************************
ZRACUSR  CSECT
SLEVEL SET=2
STM R14,R12,12(R13) SAVE REGISTERS
LR R11,R15 ADDRESSABILITY
LR R2,R1 SAVE PARMS
LA R12,2048(R11) BASE REG 2
LA R12,2048(R12) UP 2048 TO 4096 ON R12
USING ZRACUSR,R11,R12 BASE REGS
L R0,SIZDATD R0 = SIZE OF DYNAMIC AREA
GETMAIN R,LV=(0) GETMAIN DYNAMIC AREA
LR R10,R1 R10 -> DYNAMIC AREA
USING DATD,R10 ADDRESS DYNAMIC AREA
ST R13,SAVEAREA+4 SAVE CALLERS SAVEAREA ADDRESS
ST R10,8(R13) SAVE SAVEAREA ADDRESS
LR R13,R10 SAVE AREA PTR

* TEST APF AUTHORITY OF MODULE *

TESTAUTH FCTN=1 TEST APF
LTR R15,R15 RESULT
BZ CONTØ OK
TPUT =CL40'ZRACUSR09 - NOT APF AUTHORIZED',40
MVC RC,=F'9' RC 9
B EXIT2 EXIT

* TSO PARM RETRIEVAL VIA CPPL *

CONTØ EQU *
L R2,0(R2) CPPL PARM OF COMMAND BUFFER
LA R5,PARM POINT TO VARIABLE
LH R3,0(R2) LOAD LEN OF BUFFER
LH R4,2(R2) LOAD REL OFF PARM
LA R4,4(R4) LOAD OFFSET OF PARM
CR R4,R3 OFF = LEN THEN NO PARM
BNE SETPARM CONTINUE
TPUT =CL40'ZRACUSR14 - NO PARM ENTERED',40
MVC RC,=F'14' RC 14
B EXIT2 LEAVE

SETPARM EQU *
LR R6,R2 COMMAND BUFFER
AR R6,R4 PLUS OFFSET
MVC 0(1,R5),0(R6) MOVE ONE BYTE
LA R5,1(R5) UP OFFSET
LA R4,1(R4) UP OFFSET
CR R4,R3 OFF = LEN THEN END
BL SETPARM NOT YET
OC PARM,BLANKS UPPER CASE WITH BLANKS

* PARSE PARM TO EXTRACT KEY, ACT, USR, DAT, LEN, VALUES *

MVC FIELD,=CL3'KEY' SET FIELD NAME
BAL R2,PARSE CALL PARSE ROUTINE
MVC KEY,VALUE           STORE VALUE
MVC FIELD,=CL3'ACT'     SET FIELD NAME
BAL R2,PARSE            CALL PARSE ROUTINE
MVC ACT,VALUE           STORE VALUE
MVC FIELD,=CL3'USR'     SET FIELD NAME
BAL R2,PARSE            CALL PARSE ROUTINE
MVC USR,VALUE           STORE VALUE

* IF ACTION IS DEL THEN SKIP DAT AND LEN PARSING *

CLC ACT,=CL3'DEL'       IS IT DEL ACTION
BE SKIP                 SKIP DAT AND LEN CHECKS
MVC FIELD,=CL3'DAT'     SET FIELD NAME
BAL R2,PARSE            CALL PARSE ROUTINE
MVC DAT,VALUE           STORE VALUE

* IF ACTION IS GET THEN SKIP LEN PARSING *

CLC ACT,=CL3'GET'       IS IT GET ACTION
BE SKIP                 SKIP LEN CHECK
MVC FIELD,=CL3'LEN'     SET FIELD NAME
BAL R2,PARSE            CALL PARSE ROUTINE
MVC LEN,VALUE           STORE VALUE

WHEN: EQU *

CLC ACT,=CL3'PUT'       IS IT PUT ACTION
BNE LENOK               NO, SKIP LEN VALIDATE

* CONVERT 1-3 DIGIT CHARACTER NUMBER TO BINARY *

MVC PICTURE,=CL3'ØØØ'   INITIALISE FIELD WITH ZEROS
LA R1,PICTURE+2         ADDRESS END PICTURE FIELD
LA R2,LEN+2             ADDRESS END LEN FIELD
LA R3,3                  LENGTH OF FIELD
RLOOP1 EQU *

CLI Ø(R2),C' '          IS THERE A CHAR HERE
BE RNEXT                NO, SKIP TO NEXT
CLI Ø(R2),C'Ø'          LESS THAN Ø
BL LENBAD                BAD
CLI Ø(R2),C'9'          MORE THAN 9
BH LENBAD                BAD
MVC Ø(1,R1),Ø(R2)       MOVE IT TO PICTURE FIELD
BCTR R1,0                BACK ONE BYTE PICTURE FIELD
RNEXT EQU *

BCTR R2,Ø                BACK ONE BYTE PERIOD FIELD
BCT R3,RLOOP1            AND NEXT

* CONVERT LENGTH VALUE INTO BINARY FIELD *

PACK DECNUM,PICTURE     PACK IT
CVB R8,DECNUM           CONVERT TO BIN
ST R8,BINNUM            SAVE IT
CLC BINNUM,=F'Ø' IS IT ZERO
BE LENBAD BAD
CLC BINNUM,=F'255' IS IT MORE THAN MAX
BH LENBAD BAD
B LENOK GOOD
LENBAD EQU *
TPUT =CL4Ø'ZRACUSR15 - INVALID LENGTH',4Ø
MVC RC,=F'15' RC 15
B EXIT2 LEAVE
LENOK EQU *
*
* PERMIT ACCESS IF CALLER IS RACF SPECIAL OR ACCESS IF GRANTED BY
* TESTING RACF PROFILE CLASS FACILITY RESOURCE INST.USERDATA.KEYNAME
*
NORMUSR EQU *
L R8,PSATOLD-PSA CURRENT TCB
L R9,TCBSENV-TCB(R8) ACEE
LTR R9,R9 CHECK ACEE
BNZ ACEEOK THERE OK
L R9,PSAAOLD-PSA CURRENT ASCB
L R9,ASCBASXB-ASCB(R9) ASXB
L R9,ASXBSENV-ASXB(R9) ACEE
LTR R9,R9 CHECK ACEE
BNZ ACEEOK THERE
TPUT =CL4Ø'ZRACUSR1Ø - NO RACF ACEE FOUND',4Ø
MVC RC,=F'10' RC 10
B EXIT2 LEAVE
*
* ALLOW ACCESS IF RACF SPECIAL, OR USERS OWN DATA FOR READ
*
ACEEOK EQU *
USING ACEE,R9 ADDR IT
ST R9,ADDRACEE SAVE ADDR
MVC ISSUER,ACEEUSRI SAVE ISSUER
TM ACEEFLG1,ACEESPEC RACF SPECIAL ?
BO USROK OK, ALLOW
CLC ACT,=CL3'GET' READ REQUEST ?
BE TESTU YES, TEST USERS OWN DATA
*
* ALLOW THE USER TO UPDATE THEIR OWN DATA FOR CERTAIN KEYS ONLY
*
TESTK EQU *
CLC KEY,=CL8'PRNTINFO' IS IT KEY PRNTINFO
BE TESTU TEST USERS OWN DATA
B DORACG ASK PERMISSION FROM RACF
TESTU EQU *
CLC USR,ACEEUSRI IS IT THE CURRENT USERS DATA
BNE DORACG NO, ASK PERMISSION FROM RACF
B USROK YES, ALLOWED FOR THIS KEY
DROP R9
*
* NON-GLOBAL SPECIAL OBTAIN OWNING GROUP OF USER TO BE MODIFIED
* DORACG EQU *
  LA R8,USR ADDR USERID FIELD
  LA R6,RFLDIA ADDR FIELDS TO BE OBTAINED
  LA R9,RACWORK RACROUTE WORK AREA
  MVC RX4(RX4L),RX3 INITIALIZE REENTRANT AREA
  RACROUTE REQUEST=EXTRACT,WORKA=(9),RELEASE=1.8.1,
  TYPE=EXTRACT,FIELDS=(6),ENTITY=(8),MF=(E,RX4)
  L R8,RX4 LOAD RACXTRT RETURN CODE
  L R7,RX4+4 LOAD RACXTRT REASON CODE
  LTR R15,R15 TEST RACROUTE RETURN CODE
  BNZ BADICH RACXTRT FAILED
  LR R7,R1 RETURN AREA ADDR

* PROCESS THE RETURNED INFORMATION, SEG AREA IS LEN(4), DATA(8) *

XR R4,R4 CLEAR
IC R4,Ø(R7) SUBPOOL OF GETMAINED AREA
XR R5,R5 CLEAR
ICM R5,B'Ø111',1(R7) LENGTH OF GETMAINED AREA
LH R6,4(R7) OFFSET TO SEGMENT AREA
AR R6,R7 ADD BASE ADDR FOR SEGMENT AREA
MVC GROUPN,4(R6) COPY USER PROFILE OWNER (8)
FREEMAIN R,LV=(5),A=(7),SP=(4) FREE RACXTRT AREA

* CHECK ISSUER IS THE OWNER *

CLC ISSUER,GROUPN ISSUER IS THE PROFILE OWNER
BE USROK YES, OK

* CHECK ISSUER HAS GROUP SPECIAL FOR THE USERS OWNING GROUP *

CHECKG EQU *
  L R4,ADDRACEE LOAD ACEE ADDR
  USING ACEE,R4
  L R5,ACEE_CGRP ADDR OF CONNECT TABLE, 1
  LTR R5,R5 TEST
  BNZ OKCG OK, THERE
  L R5,ACEE_FCGP ADDR OF CONNECT TABLE, 2
  LTR R5,R5 TEST
  BZ DORACH NOT THERE

OKCG EQU *
  USING CGRP,R5 ADDR CONNECT TABLE
  LA R6,CGRPENT ADDR FIRST ENTRY
  LH R7,CGRPNUM NUMBER OF ENTRIES
  USING CGRPENTD,R6

LOOPG EQU *
  CLC GROUPN,CGRPNAME IS THIS THE GROUP
  BE MATCHG YES
  LA R6,L'CGRPENT(R6) INCREMENT ENTRY
  BCT R7,LOOPG CHECK NEXT
B     NEXTG                    NOT AUTHORISED

MATCHG EQU *
TM   CGRPAUTH,CGRPSPEC         GROUP SPECIAL FLAG
BO   USROK                     ON, AUTH
NEXTG EQU *
CLC GROUPN,=CL8'SYS1'         IS IT THE FINAL GROUP
BE   DORACH                    YES, NOT AUTHORIZED HERE

* GET SUPERIOR GROUP TO LAST GROUP AND CHECK AGAIN UNTIL SYS1 *

LA   R8,GR Nguyen               ADDR GROUP FIELD
LA   R6,RFLD2A                  ADDR FIELDS TO BE OBTAINED
LA   R9,RACWORK                 RACROUTE WORK AREA
MVC RX6(RX6L),RX5               INITIALISE REENTRANT AREA
RACROUTE REQUEST=EXTRACT,WORKA=(9),RELEASE=1.8.1,
TYPE=EXTRACT,FIELDS=(6),ENTITY=(8),MF=(E,RX6)
L    R8,RX6                     LOAD RACXTRT RETURN CODE
L    R7,RX6+4                   LOAD RACXTRT REASON CODE
LTR  R15,R15                    TEST RACROUTE RETURN CODE
BNZ  BADICH                     RACXTRT FAILED
LR   R7,R1                      RETURN AREA ADDR

* PROCESS THE RETURNED INFORMATION, SEG AREA IS LEN(4), DATA(8) *

XR   R4,R4                      CLEAR
IC    R4,Ø(R7)                  SUBPOOL OF GETMAINED AREA
XR   R5,R5                      CLEAR
ICM   R5,B'Ø111',1(R7)          LENGTH OF GETMAINED AREA
LH    R6,4(R7)                  OFFSET TO SEGMENT AREA
AR    R6,R7                     ADD BASE ADDR FOR SEGMENT AREA
MVC GROUPN,4(R6)                COPY GROUP SUPERIOR GROUP
FREEMAIN R,LV=(5),A=(7),SP=(4)   FREE RACXTRT AREA
L    R1,COUNTG                  LOAD COUNT OF LOOPS
LA    R1,1(R1)                  INCREMENT IT
ST    R1,COUNTG                 SAVE COUNT
CLC   COUNTG,=F'2ØØ'           2ØØ LOOPS
BH    DORACH                     STOP LOOPING IF NO SYS1 FOUND
B    CHECKG                     TRY AGAIN FOR THIS GROUP
DROP R4,R5,R6

* PERFORM RACHECK ON FACILITY CLASS RESOURCE, FOR UPDATE TO THE *
* NON-PRNTINFO USERDATA OR ANY ACCESS TO OTHER USERS USERDATA. *

DORACH EQU *
MVC RESO,BLANKS                  CLEAR RESOURCE NAME
MVC RESO(14),=CL14'INST.USERDATA.'
MVC RESO+14(8),KEY                ADD KEY NAME

* ADD USER-ID TO END OF RESOURCE NAME *

LA   R1,RESO+14                  START OF SCAN
LA   R15,L'RESO-14            MAX LOOP
LOOPRES EQU *
CLI   Ø(R1),C' '             BLANK YET
BE   ENDRES                 YES
LA   R1,1(R1)                INCREMENT
BCT  R15,LOOPRES             CONT
ENDRES EQU *
MVI   Ø(R1),C'.'             ADD DOT
MVC   1(B,R1),USR           ADD USER-ID
CLC   ACT,=CL3'GET'          READ REQUEST ?
BE   TRYR                   YES
B    TRYU                   NO, UPDATE
TRYR EQU *
CLC   KEY(4),=CL4'LIST'      LIST REQUEST (NO LOGGING)
BE   TRYRNL                 YES
RACHECK CLASS='FACILITY',ATTR=READ,ENTITY=RESO
B    TESTR                  CHECK RESULT
TRYRNL EQU *
RACHECK CLASS='FACILITY',ATTR=READ,ENTITY=RESO,LOG=None
B    TESTR                  CHECK RESULT
TRYU EQU *
RACHECK CLASS='FACILITY',ATTR=UPDATE,ENTITY=RESO
B    TESTR                  CHECK RESULT
TESTR EQU *
LTR   R15,R15                TEST RC
BZ   USROK                   AUTHORIZED TO KEY
FAILR EQU *
TPUT =CL5Ø'ZRACUSR16 - RACF DENIED ACCESS TO USERDATA'.5Ø
MVC   RC,=F'16'              RC 16
B    EXIT2                   LEAVE
USROK EQU *
*
* SET LENGTH OF USERID
*
LA   R4,USR                  ADDR USER-ID
LA   R5,8                    MAX LENGTH
LA   R6,Ø                    COUNTER
LOOPU EQU *
CLI   Ø(R4),C' '             END YET
BE   ENDU                    YES
LA   R4,1(R4)                UP PTR
LA   R6,1(R6)                UP COUNTER
BCT  R5,LOOPU                LOOP
ENDU EQU *
STC   R6,ENAME                SET LEN
MVC   EUSER,USR               SET NAME AS RACF ENTITY
*
* SELECT REQUIRED ACTION
*
CLC   ACT,=CL3'GET'          IS IT GET (RETRIEVE)
BE   GETUSRD                 YES
CLC ACT,=CL3'PUT' IS IT PUT (UPDATE)  
BE PUTUSRD YES  
CLC ACT,=CL3'DEL' IS IT DEL (DELETE)  
BE DELUSRD YES  
TPUT =CL40'ZRACUSR17 - INVALID ACTION',40  
MVC RC,=F'17' RC 17  
B EXIT2 LEAVE  

* RETRIEVE DATA FROM RACF FOR GIVEN KEY  
*  
GETUSRD EQU *  
MVC VARNAME,DAT SET CLIST/REXX VARIABLE NAME  
MVC DAT,BLANKS CLEAR DATA AREA  
CLC KEY,=CL8'LISTMEMS' DOES HE WANT MEMBER LIST?  
BE MEMLIST YES  
CLC KEY,=CL8'LISTALLD' DOES HE WANT FULL LIST?  
BE MEMLIST YES  
B NOTLIST NOT LIST  

* GET ALL USERDATA IF KEY = LISTMEMS. STORE MEMBER NAMES.  
*  
MEMLIST EQU *  
LA R9,RACWORK RACROUTE WORK AREA  
RACROUTE REQUEST=EXTRACT,WORKA=(9),TYPE=EXTRACT, X  
FIELDS=RFLD3A,ENTITY=USR,RELEASE=1.8.1  
C R15,=F'4' RC 4  
BE BADUSR USER NOT FOUND  
LTR R15,R15 TEST RC  
BNZ EXIT2 BAD  

* PROCESS THE RETURNED INFOMATION, LEN(4), DATA(4), LEN(4), DATA(NN)  
*  
LR R7,R1 RETURN AREA ADDR  
XR R4,R4 CLEAR  
IC R4,Ø(R7) SUBPOOL OF GETMAINED AREA  
XR R5,R5 CLEAR  
ICM R5,B'Ø111',1(R7) LENGTH OF GETMAINED AREA  
LH R6,4(R7) OFFSET TO SEGMENT AREA  
AR R6,R7 ADD BASE ADDR FOR SEGMENT AREA  
CLC KEY,=CL8'LISTALLD' DOES HE WANT FULL LIST?  
BE SETALLD YES  

* SCAN USERDATA (L-NAME, NAME, L-DATA, DATA, L-FLAG, FLAG)  
*  
MVC DAT(4),4(R6) COPY DATABASE USRCNT (MEMBER COUNT)  
MVC BINNUM,=F'4' ASSUME LENGTH FOUR FOR NOW  
L R9,4(R6) LOAD NUMBER OF MEMBERS  
LTR R9,R9 TEST IF ANY  
BZ CONT2 NONE  
LR R1,R9 COPY COUNT  
MH R1,=H'8' TIMES EIGHT  

LA R1,4(R1) ADD 4
ST R1,BINNUM SAVE AS TSO VAR LENGTH
LA R3,DAT+4 START OF OUTPUT AREA + 4
LA R2,12(R6) START OF USERDATA AREA

LOOPUD EQU *
L R1,Ø(R2) LOAD LEN OF MEMBER NAME
LA R2,4(R2) ADDR START OF MEMBER NAME
MVC Ø(8,R3),BLANKS CLEAR OUTPUT MEMBER NAME
BCTR R1,Ø DECREASE FOR EX
EX R1,MOVEM MOVE MEMBER NAME
AR R2,R1 ADD LEN TO OFFSET
LA R2,1(R2) ALLOW FOR BCTR
L R1,Ø(R2) LOAD LEN OF DATA
LA R2,4(R2) ADDR START OF DATA
AR R2,R1 ADD LEN TO OFFSET
L R1,Ø(R2) LOAD LEN OF FLAG
LA R2,4(R2) ADDR START OF FLAG
AR R2,R1 ADD LEN TO OFFSET
LA R3,8(R3) INCREMENT OUTPUT AREA
BCT R9,LOOPUD REPEAT FOR EACH MEMBER

FREEUD EQU *
FREEMAIN R,LV=(5),A=(7),SP=(4) FREE RACXTRT AREA
MVC RC,=F'Ø' SAVE RETURN CODE
B CONT2 CONTINUE

MOVEM MVC Ø(Ø,R3),Ø(R2) MOVE MEMBER NAME

SETALLD EQU *
ST R5,BINNUM LENGTH OF AREA
ST R7,DATPTR AREA ADDR
L R15,CVTPTRX GET TSOE SERVICE ROUTINE
L R15,CVTTVTX(.R15) ADDRESS
L R15,TSVTACC-TSVT(.R15) FROM CVT
CALL (15),(ECODE,NAMEPTR,L6,DATPTR,BINNUM,TOKEN),VL SET
LTR R15,R15 TEST RC
BZ ENDALLD COMPLETED OK
TPUT =CL4Ø'ZRACUSRØ6 - TSO VARIABLE UPDATE FAILURE',5Ø
MVC RC,=F'6' RC 6
B EXIT2 CONTINUE

ENDALLD EQU *
FREEMAIN R,LV=(5),A=(7),SP=(4) FREE RACXTRT AREA
MVC RC,=F'Ø' SAVE RETURN CODE
B EXIT CONTINUE

NOTLIST EQU *
XC WR,WR CLEAR RESERVED AREA
XC WR2,WR2 CLEAR RESERVED AREA
ICHEINTY LOCATE,TYPE='USR',ENTRY=ENAME,WKAREA=WST,
ACTIONS=(ACTNI)
LR R8,R15 SAVE RETURN CODE
LR R7,RØ SAVE REASON CODE
C     R8,=F'12'               RC 12
BNE  CONT1                   NOT USER NOT FOUND
BADUSR EQU * 
TPUT  =CL4Ø'ZRACUSRØ5 - USERID NOT DEFINED TO RACF',4Ø
MVC  RC,=F'5'                RC 5
B    EXIT2                   LEAVE
CONT1 EQU * 
LTR  R8,R8                   TEST RESULT
BNZ  BADICH                   NOT GOOD
* 
* WST+X'1C'+2 IS THE USERDATA, NAME(8) LEN(1), DATA(1-255), FLAG(1)
* 
CLC  WST+X'1C'+2(8),KEY      COMPARE KEY NAME
BE   GOTKEY                   IT IS THERE
***** TPUT  =CL4Ø'ZRACUSRØ1 - GET REQUEST NO PREVIOUS DATA',4Ø
MVC  RC,=F'1'                RC 1
B    EXIT2                   LEAVE
GOTKEY EQU * 
XR   R8,R8                   CLEAR R8
IC   R8,WST+X'1C'+2+8        COPY LENGTH FROM RACF
ST   R8,BINNUM               SAVE LEN
BCTR  R8,Ø                    DOWN FOR EX
EX   R8,GETDAT               COPY IT
B    CONT2                   CONTINUE
* 
GETDAT MVC  DAT(Ø),WST+X'1C'+2+9    COPY DATA FROM RACF
* 
* STORE IN TSO VARIABLE VIA TSO SERVICE ROUTINE
*
CONT2 EQU * 
L   R15,CVTPTRX            GET TSOE SERVICE ROUTINE
L   R15,CVTTVTX(,R15)       ADDRESS
L   R15,TSVTACC-TSVT(,R15)   FROM CVT
CALL (15),(ECODE,NAMEPTR,L6,DATPTR,BINNUM,TOKEN),VL SET
LTR  R15,R15                   TEST RC
BZ    EXIT                    COMPLETED OK
TPUT  =CL4Ø'ZRACUSRØ6 - TSO VARIABLE UPDATE FAILED',4Ø
MVC  RC,=F'6'                RC 6
B    EXIT2                   LEAVE
*
* UPDATE THE RACF DATA FOR GIVEN KEY MEMBER
*
PUTUSRD EQU * 
CLC  KEY,=CL8'LISTMEMS'      DONT ALLOW THIS NAME
BE   FAILI                   FAIL ATTEMPT
MVC  UKEY,KEY                 KEY NAME TO RACF FIELD
L   R8,BINNUM                 LOAD BINARY LENGTH
STC  R8,UDATALEN              INTO RACF FIELD
LA   R1,UDATAV                DATA VALUE
AR   R1,R8                    PLUS DATA LEN
BCTR  R8,Ø                    DOWN FOR EX
EX R8,PUTDAT  COPY DATA VALUE TO RACF FIELD
MVI Ø(R1),X'ØØ'  SET FLAG FIELD AT END
L R2,BINNUM  LOAD BINARY LENGTH
LA R2,1Ø(R2)  ADD 1Ø FOR OTHER FIELDS LEN
CLC ACTN3+X'1Ø'(4),=AL4(13)  IS IT DEFAULT LEN IN ACTN3
BE CONT3  CONTINUE AND UPDATE IT
TPUT =CL4Ø'ZRACUSR12 - RACF ICHEACTN PARMS WRONG',4Ø
MVC RC,=F'12'  RC 12
B EXIT2  LEAVE

* PUTDAT MVC UDATAV(Ø),DAT  SET UP DATA
*
CONT3 EQU *
ST R2,ACTN3+X'1Ø'  UPDATE PARAMETER LENGTH
XC WR,WR  CLEAR RESERVED AREA
XC WR2,WR2  CLEAR RESERVED AREA
ICHEINTY ALTER,TYPE='USR',ENTRY=ENAME, ACTIONS=(ACTN2,ACTN3)  X
LR R8,R15  SAVE RETURN CODE
LR R7,RØ  SAVE REASON CODE
LTR R8,R8  TEST RESULT
BNZ BADICH  NOT GOOD
B EXIT  LEAVE OK

* * DELETE THE RACF DATA FOR GIVEN KEY MEMBER
*
DELUSRD EQU *
XC WR,WR  CLEAR RESERVED AREA
XC WR2,WR2  CLEAR RESERVED AREA
ICHEINTY ALTER,TYPE='USR',ENTRY=ENAME, ACTIONS=(ACTN2)
LR R8,R15  SAVE RETURN CODE
LR R7,RØ  SAVE REASON CODE
LTR R8,R8  TEST RESULT
BNZ BADICH  NOT GOOD
B EXIT  LEAVE OK

* * SHOW ICH RETURN/REASON CODE
*
BADICH EQU *
MVC RMSG,BLANKS  CLEAR MSG
CVD R8,WORK1  CONVERT TO DEC
CVD R7,WORK2  CONVERT TO DEC
UNPK WORK3,WORK1  UNPACK
MVZ WORK3+7(1),=X'FØ'  SET ZONE
MVC RMSG(9),=CL9'RETURN = ' INTO MSG
MVC RMSG+9(8),WORK3  INTO MSG
UNPK WORK4,WORK2  UNPACK
MVZ WORK4+7(1),=X'FØ'  SET ZONE
MVC RMSG+9+9(9),=CL9'REASON = ' INTO MSG
MVC RMSG+9+9+9(8),WORK4  INTO MSG
FAILI EQU *

TPUT =CL4Ø'ZRACUSR11 - RACF ICHEINTY MACRO FAILED',4Ø
MVC RC,=F'11' RC 11
TPUT RMSG,L'RMSG INFORMATION MESSAGE
B EXIT2 LEAVE

EXIT EQU *

* LEAVE PROGRAM *

EXIT2 EQU *

L R13,SAVEAREA+4 RESTORE R13
L R3,RC RETURN CODE
L R0,SIZDATD R0 = DYNAMIC AREA SIZE
LR R1,R10 R1 -> DYNAMIC AREA ADDRESS
FREEMAIN R,LV=(Ø),A=(1) FREE DYNAMIC AREA
LR R15,R3 RETURN CODE
L R14,12(R13) RESTORE R14
LM R0,R12,2Ø(R13) RESTORE R0 TO R12
BR R14 RETURN

* SUBROUTINE R2, PARSE PARM FOR FIELDS REQUIRED *

PARSE EQU *

LA R3,PARM ADDR PARM
LA R4,24Ø MAX SCAN LEN

SCANP EQU *

CLC Ø(4,R3),FIELD IS IT FIELD NAME PLUS ( 
BE ENDS YES
LA R3,1(R3) UP PTR IN PARM
BCT R4,SCANP SCAN AGAIN
TPUT =CL4Ø'ZRACUSRØ7 - PARAMETER WAS MISSING',4Ø
MVC RC,=F'7' RC 7
TPUT FIELD,3 SAY WHICH
B EXIT OUT

ENDS EQU *

MVC VALUE,BLANKS INITIALIZE
LA R5,VALUE VALUE FIELD
LA R4,255 MAX VALUE LEN
LA R3,4(R3) SKIP TO DATA IN PARM
MVI QUOTE,C'N' SET FLAG DEFAULT
CLI Ø(R3),C"" IS FIRST CHAR QUOTE
BNE MOVEV NO
MVI QUOTE,C'Y' SET FLAG
LA R3,1(R3) JUMP OVER QUOTE

MOVEV EQU *

CLI QUOTE,C'Y' QUOTE FLAG
BNE NORMP NO
CLI Ø(R3),C"" IS IT QUOTE
BE ENDV YES THEN END
B QUOP SKIP USUAL TEST

NORMP EQU *

CLC Ø(2,R3),=CL2') ' END OF PARM FIELD
BE ENDV YES
QUOP EQU *
MVC Ø(1,R5),Ø(R3) MOVE ONE CHAR
LA R5,1(R5) NEXT CHAR IN VALUE
LA R3,1(R3) NEXT CHAR IN PARM
BCT R4,MOVEV LOOK AGAIN
TPUT =CL4Ø'ZRACUSRØ8 - DELIMITER IS MISSING',4Ø
MVC RC,=F'8' RC 8
TPUT FIELD,3 SAY WHICH
B EXIT OUT
ENDV EQU *
BR R2
******************************************************************
*                                                                *
* DATA AREAS                                                     *
*                                                                *
******************************************************************
LTORG
DATA DS ØF
SIZDATD DC AL1(Ø) DYNAMIC AREA SIZE
DC AL3(ENDDATD-DATD)
PATCHA DC 2ØS(*) PATCH AREA FOR ZAPS
DS ØD
DECNUM DC PL8'Ø' LENGTH IN DECIMAL
WORK1 DC PL8'Ø' WORK AREA FOR RETURN/REASON CODES
WORK2 DC PL8'Ø' WORK AREA FOR RETURN/REASON CODES
BINNUM DC F'Ø' LENGTH IN BINARY
RC DC F'Ø' PROGRAM RETURN CODE - DEFAULT Ø
L6 DC F'6' LENGTH 6
UDLEN DC F'Ø' USERDATA LENGTH
COUNTG DC F'Ø' LOOP COUNT FOR GROUP SPECIAL
RFLD1A DC A(1) NUMBER OF FIELDS FOLLOWING
RFLD1B DC CL8'AUTHOR' NAME OF FIELD (OWNER)
RFLD2A DC A(1) NUMBER OF FIELDS FOLLOWING
RFLD2B DC CL8'SUPGROUP' NAME OF FIELD (SUPERIOR GROUP)
RFLD3A DC A(2) NUMBER OF FIELDS
RFLD3B DC CL8'USRCNT' USER COUNT
RFLD3C DC CL8'USERDATA' USER DATA
WORK3 DC CL8' ' WORK AREA FOR RETURN/REASON CODES
WORK4 DC CL8' ' WORK AREA FOR RETURN/REASON CODES
FIELD DC CL3' ' FIELD NAME
FIELD2 DC C'(' BRACKET (KEEP AFTER FIELD)
VALUE DC CL255' ' VALUE OF FIELD NAME
PICTURE DC CL3'ØØØ' PICTURE FIELD
USRMOD DC CL1' ' USER MOD FLAG
QUOTE DC CL1' ' QUOTES USED FLAG
KEY DC CL8' ' RACF USRDATA MEMBER NAME
ACT DC CL3' ' ACTION TYPE
USR DC CL8' ' USERID
DAT DC CL255' ' DATA
LEN   DC    CL3' '          LENGTH
RMSG  DC    CL40' '          RACF MSG
RESO  DC    CL39' '          FACILITY CLASS RESOURCE NAME
PARM  DC    CL255' '         TSO COMMAND PARM
BLANKS DC    CL255' '        BLANKS
ZEROS DC    100X'00'         ZEROS
MASK  DC    255X'55'         MASK FOR HIDING CLEAR DATA
*
* RACF ACTION AND TEST MACROS
*
* GET FOR MEMBER NAME AS IN KEY
*
ACTN1 ICHEACTN FIELD=USERDATA,TESTS=TEST1
TEST1  ICHETEST FIELD=USRNM,FLDATA=(8,KEY),COND=EQ
*
* DEL FOR MEMBER NAME AS IN KEY, ALSO USED IN PUT SEQUENCE
*
ACTN2 ICHEACTN FIELD=USRNM,FLDATA='DEL',TESTS=TEST2
TEST2  ICHETEST FIELD=USRNM,FLDATA=(8,KEY),COND=EQ
*
* PUT FOR MEMBER NAME AS IN KEY, THE REAL LENGTH IS ZAPPED IN LATER
*
ACTN3 ICHEACTN FIELD=USERDATA,FLDATA=(13,RMemb),TESTS=TEST3
TEST3  ICHETEST FIELD=USRNM,FLDATA=(8,KEY),COND=NE
*
* RACXTRT FOR OWNER AND SUBGROUP
*
RX3   RACROUTE REQUEST=EXTRACT,WORKA=ZEROS,RELEASE=1.8.1, TYPE=EXTRACT,FIELDS=ZEROS,ENTITY=ZEROS,CLASS='USER',MF=L
*
RX5   RACROUTE REQUEST=EXTRACT,WORKA=ZEROS,RELEASE=1.8.1, TYPE=EXTRACT,FIELDS=ZEROS,ENTITY=ZEROS, CLASS='GROUP',MF=L
*
* ICHEINTY WORK AREA
*
ENAME  DC    AL1(Ø)          LENGTH OF ENTITY NAME
EUSER  DC    CL44' '         RACF ENTITY NAME
WST    DS    ØF
       DC    AL4(WEND-WST)   LENGTH OF WORK AREA
WR     DC    CL200' '        RESERVED AREA
WR2    DC    CL200' '        RESERVED AREA
WEND   DS    ØF
*
* USER DATA REPEAT GROUP MEMBER LAYOUT SEE RACF TEMPLATES
*
RMemb DS    ØC
UKEY   DC    CL8' '          KEY NAME OF THIS REPEAT GROUP MEMBER
UDATALEN DC    AL1(Ø)       LENGTH OF VARIABLE DATA
UDATAV  DC    CL256' '       VARIABLE LENGTH DATA (MAX 255)
* A FLAG FIELD IS ADDED AFTER THE DATA VALUE TO COMPLETE THESE FIELDS
* TSO VARIABLE UPDATE FIELDS
*
CVTPTRX EQU 16
CVTTVTX EQU '9C'
ECODE DC (TSVEUPDT) ENTRY CODE FOR VARIABLE UPDATE
TOKEN DC 'F'Ø'
VARNAME DC CL8' ' TSO VARIABLE NAME
NAMEPTR DC A(VARNAME) PTR TO VARIABLE NAME
DATPTR DC A(DAT) PTR TO DATA FIELD
*
* UNINITIALISED STORAGE (GETMAINED)
*
DATD DSECT GETMAINED STORAGE AREA
SAVEAREA DS 18F SAVE AREA
ADDRACEE DS F ACEE ADDR
ISSUER DS CL8 COMMAND ISSUER
GROUPN DS CL8 USER OWNING GROUP
RX4 RACROUTE REQUEST=EXTRACT, WORKA=ZEROS, RELEASE=1.8.1, TYPE=EXTRACT, FIELDS=ZEROS, ENTITY=ZEROS, CLASS='USER', MF=L
RX4L EQU *-RX4 RACXTRT LENGTH
RX6 RACROUTE REQUEST=EXTRACT, WORKA=ZEROS, RELEASE=1.8.1, TYPE=EXTRACT, FIELDS=ZEROS, ENTITY=ZEROS, CLASS='GROUP', MF=L
RX6L EQU *-RX6 RACXTRT LENGTH
*
* RACROUTE WORK AREA
*
RACWORK DS CL512 RACROUTE WORK AREA
ENDDATD EQU *
*
* DSECTS
*
YREGS
PRINT NOGEN
IHAASCB
IHAASXB
IHAPSA
IHAACEE
ICHPCGRP CGRP
IKJTSVT
IKJTCB
IKJRB
IHACDE
IEESMCA SMCA
CVT DSECT=YES
IEZJSCB
END ZRACUSR

© Xephon 1999
RACF users can benefit from Neon Systems’ enterprise security management product, Halo SSO, providing a Single Sign-On (SSO) solution for mainframe applications in interoperable OS/390 and Microsoft SNA Server/Windows NT environments, without custom coding or installation of software at the desktop. OS/390-MVS/Windows security issues can be managed with automated password synchronization from a centralized, secure control point allowing end users to enter one password for access to both the mainframe OS/390 environment and the Windows environment.

Halo SSO includes both MVS and Windows NT components that continuously monitor mainframe or Windows NT-initiated password changes and propagate password changes to the appropriate domains. With logging and monitoring capabilities, Halo SSO also provides single sign-on benefits when used with application log-on scripts. The Halo SSO mainframe component supports RACF, CA-ACF2, and CA-Top Secret.

For further information contact:
Neon Systems, 14141 Southwest Freeway, Suite 6200, Sugar Land, TX 77478, USA.
Tel: (281) 491 4200.
Neon Systems, Third Floor, 26-30 London Road, Twickenham, Middlesex, TW1 3RW, UK.
Tel: (0181) 607 9911.

* * *

IBM has announced OS/390 Version 2 Release 7, with key focus areas including security, network support, and system management. Release 7 includes tighter integration between the Tivoli management framework and the SystemView-based System/390 environment. The Tivoli Management Agent has been integrated into Release 7, allowing mainframe facilities such as RACF to support their Tivoli counterparts, while having a consistent Tivoli view across networks that include System/390s.

A new feature is the Open Cryptographic Services Facility, providing security and cryptographic services for use in applications running in OS/390’s Unix environment.

LDAP services include enhancements to user-id and password authentication through Security Server, client access to Security Server RACF USER and GROUP information, support for multiple LDAP servers in an System/390 Parallel Sysplex cluster environment, and Java support for Java naming and directory interface.

Looking ahead to Release 8, various security changes will include a new non-password-assigned ID type for RACF, SNA Triple DES session level encryption, and TN3270 SSL client authentication.

For further information contact your local IBM representative.