12 RACF

May 1998

In this issue

update

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 75067
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 twelve monthly issues, costs £185.00 in the UK; $280.00 in the USA and Canada; £191.00 in Europe; £197.00 in Australasia and Japan; and £195.50 elsewhere. In all cases the price includes postage. Individual issues, starting with the August 1995 issue, are available separately to subscribers for £46.50 ($70.00) each including postage.

Printed in England.

Restricting password changes

RACF provides a standard facility to enforce a maximum interval for password changes, but does not allow a minimum interval to be specified. Without a minimum interval, a user can simply cycle back round to their favourite password when it expires, by repeatedly changing their password.

The password history is one deterent to this, but it is easy to write a small CLIST or REXX to loop round even a large history value. What is really needed is a minimum interval of one day. The exit code below implements this by preventing a password change (other than by an administrator) more than once a day.

This code is implemented as the RACF new password exit ICHPWX01:

```
*********************************************************************
* MODULE : ICHPWXØ1
* RACF NEW PASSWORD EXIT
*
* USED TO ENFORCE PASSWORD CHANGE RESTRICTIONS, CALLED BY:
* 1. NEW PASSWORD ON RACINIT
* 2. ALTUSER COMMAND WITH PASSWORD OPERAND
* 3. PASSWORD COMMAND
*
* PASSWORD CANNOT BE CHANGED ON THE SAME DAY AS LAST CHANGE
* EXCEPT WHEN CHANGED BY ADMINISTRATOR WITH ALU COMMAND
*
* REENTRANT, AUTHORIZED, Y2K COMPLIANT
*********************************************************************
ICHPWXØ1 CSECT
ICHPWXØ1 AMODE 31
ICHPWXØ1 RMODE ANY
         SPLEVEL SET=2
         STM   R14,R12,12(R13)   SAVE REGISTERS
         LR    R12,R15           ADDRESSABILITY
         LR    R11,R1            SAVE PARM ADDR
         USING ICHPWXØ1,R12
         GETMAIN R,LV=WORKLEN    GETMAIN DYNAMIC AREA
         LR    RIØ,R1            RIØ -> DYNAMIC AREA
         USING WORKAREA,RIØ      ADDRESS DYNAMIC AREA
         ST    R13,SAVEAREA+4    SAVE CALLERS SAVEAREA ADDRESS
         ST    RIØ,8(R13)        SAVE SAVEAREA ADDRESS
         LR    R13,RIØ           SAVE AREA PTR
*
* ADDRESS PARAMETER LIST, SET DEFAULT RC
```

```
*
         USING PWXPL,R11         ADDR PARM LIST
         MVC   RETC,ZEROS        DEFAULT ZERO RC
*
* CHECK CALL TYPE AND PROCESS
*
         L     R1,PWXCALLR       LOAD ADDR OF FLAG BYTE
         CLI   Ø(R1),PWXRINIT    IS IT RACINIT CALL
         BE    CHECK             YES
         CLI   Ø(R1),PWXPWORD    IS IT PASSWORD CALL?
         BE    CHECK             YES
         CLI   Ø(R1),PWXALTUS    IS IT ALU CMD CALL?
         BE    ALLOW             YES
         B     EXIT              OTHERS (?)
*
* IF PASSWORD DATE IS SAME AS TODAY THEN FAIL PASSWORD CHANGE
*
CHECK    EQU   *
         TIME  DEC               GET TODAY'S TIME/DATE
         ST    R1,TODAYS         TODAY'S DATE (ØCYYDDDS) 4 BYTE
         L     R2,PWXPLCDA       USERS PASSWORD CHANGE DATE ADDR
         ZAP   DECN,Ø(3,R2)      CONVERT PASSDATE TO DECIMAL
         CVB   R4,DECN           CONVERT PASSDATE TO BINARY
         ZAP   DECN,TODAYS+1(3)  CONVERT TODAY'S TO DECIMAL
         CVB   R5,DECN           CONVERT TODAY'S TO BINARY
         CR    R4,R5             COMPARE VALUES
         BNE   ALLOW             DIFFERENT SO ALLOW
         B     FAIL              SAME DAY - FAIL
*
* PREVENT THIS PASSWORD CHANGE WITH RC = 4
*
FAIL     EQU   *
         WTO   'ICHPWXØ1 - NEW PASSWORD FAILED - SAME DAY',ROUTCDE=11
         MVC   RETC,=F'4'        FAIL ATTEMPT TO CHANGE PASSWORD
         B     EXIT
*
* ALLOW THIS PASSWORD CHANGE WITH RC = Ø
*
ALLOW    EQU   *
         MVC   RETC,ZEROS        ALLOW ATTEMPT TO CHANGE PASSWORD
         B     EXIT
*
* QUIT EXIT, SET RC TO Ø OR 4
*
EXIT     EQU   *
         L     R2,RETC           SAVE RC
         L     R13,SAVEAREA+4    RESTORE R13
         FREEMAIN R,LV=WORKLEN,A=(1Ø) FREE DYNAMIC AREA
         LR    R15,R2            RETURN CODE TO R15
         L     R14,12(R13)       RESTORE R14
         LM    RØ,R12,2Ø(R13)    RESTORE RØ TO R12
```

```
         BR    R14               RETURN
*
         LTORG
*
* CONSTANTS, NON-MODIFIABLE
*
PATCHA   DC    20S(*)            PATCH AREA FOR ZAPS
ZEROS    DC    256XL1'00'        ZEROS
BLANKS   DC    CL255' '          BLANKS
*
* PROGRAM DATA AREA
*
WORKAREA DSECT
SAVEAREA DS    18F               SAVE AREA
RETC     DS    F                 EXIT RETURN CODE
*
ALIGND   DS    0D                ALIGN DOUBLEWORD
DECN     DS    PL8               DECIMAL WORKAREA
TODAYS   DS    XL4               DATE 0CYYDDDS
*
WORKLEN  EQU   *-WORKAREA
*
* DSECTS
*
         ICHPWXP                 NEW PASSWORD EXIT PARM MAPPING
         YREGS
         PRINT NOGEN
         IEZBITS                 BITS
         IHAACEE                 ACEE
         END
```


Encrypting/decrypting datasets – part 2

This month we complete the code for SKED80F, which aims to provide single-key encryption/decryption for 80-byte fixed-length records.

```
TRNLP1A  EQU   *
         LA    R05,USED80
         LA    R06,0
         IC    R06,OFFBYTE
         AR    R05,R06           Point into USED80
         TM    0(R05),X'80'      Has this offset value already been
         BNO   TRNLP1B            used? If so, increment R10 and
```

```
         ST    R14,SAVER14
         BAL   R14,INCRR1Ø         loop back
         L     R14,SAVER14
         B     TRNLP1
TRNLP1B  EQU   *
         MVC   Ø(1,RØ2),OFFBYTE  Store offset value in TVECn
         OI    Ø(RØ5),X'8Ø'      Flag it as having been used
         LA    RØ2,1(,RØ2)       Point to next TVECn byte
         ST    R14,SAVER14
         BAL   R14,INCRR1Ø       Increment R1Ø pointer
         L     R14,SAVER14
         BCT   RØ3,TRNLP1
*
         CH    RØ9,=H'2'
         BLR   R14               Exit if RØ9 is less than 2
*                                (Both vectors have been set up)
         L     RØ5,KEY8Ø+6Ø
         AL    RØ5,KEY8Ø+64
         X     RØ5,KEY8Ø+68
         AL    RØ5,KEY8Ø+72
         X     RØ5,KEY8Ø+76
*
         LA    RØ4,Ø
         D     RØ4,F15921        Calculate an offset
         LA    R1Ø,TAB16ØØØ
         AR    R1Ø,RØ4           Point into TAB16ØØØ
*
         XC    USED8Ø,USED8Ø
         LA    RØ2,TVEC2
         BCT   RØ9,TRNLOOP
*
         BR    R14
*-------------------------------------*
*    Increment the TAB16ØØØ pointer value in R1Ø.                    *
*-------------------------------------*
INCRR1Ø  EQU   *
         LA    R1Ø,1(,R1Ø)
         C     R1Ø,EOTAB16K      Pointing past end of table?
         BLR   R14               BIF not, return
*
         LA    R1Ø,TAB16ØØØ      Set pointer back to start of table
*
         BR    R14
*-------------------------------------*
*    Rotate TAB16ØØØ.                                                *
*-------------------------------------*
T16KROT  EQU   *
         ST    RØ4,SAVERØ4
         ST    R14,SAVER14
```

```
*
         LA    RØ5,Ø
         LA    RØ7,KEYSØ
         LA    RØ8,1Ø
T16LOOP  EQU   *               Create dividend in RØ5
         X     RØ5,Ø(RØ7)
         AL    RØ5,4(RØ7)
         LA    RØ7,8(,RØ7)
         BCT   RØ8,T16LOOP
*
         LA    RØ4,Ø
         LA    RØ1,4ØØ1        Divisor in RØ1 (4ØØØ words maximum)
         DR    RØ4,RØ1
         STM   RØ4,RØ5,REMQUOT Save remainder and quotient
         LA    RØ2,TAB16ØØØ
         LR    RØ9,RØ2         Start of TAB16ØØØ into RØ9
T16ROTLP EQU   *
         L     RØ3,REMQUOT     Use remainder as number of words
         LTR   RØ3,RØ3         As soon as we get a zero
         BZ    T16LAST          remainder then exit
         LR    RØ8,RØ3
         SLL   RØ8,2           Multiply number of words by 4 to
*                               create an offset into TAB16ØØØ
         BAL   R14,ROTATE
*
         S     RØ1,REMQUOT     Subtract remainder from original
*                               divisor to create new divisor
         CH    RØ1,=H'1'       Check for one
         BNH   T16EXIT         BIF not high, exit
*
         AR    RØ9,RØ8         Add offset to next word to be
         LR    RØ2,RØ9          processed
         LA    RØ4,Ø
         L     RØ5,Ø(RØ2)      New dividend value
         AL    RØ5,REMQUOT      plus remainder
         AL    RØ5,REMQUOT+4     plus quotient
         DR    RØ4,RØ1
         STM   RØ4,RØ5,REMQUOT Save remainder and quotient
         B     T16ROTLP
*
T16LAST  EQU   *
         LR    RØ3,RØ1
         BCTR  RØ3,Ø           Divisor minus 1 as number of words
         BAL   R14,ROTATE       for final rotation
*
T16EXIT  EQU   *
         L     RØ4,SAVERØ4
         L     R14,SAVER14
         BR    R14
```

```
*-------------------------------------*
*    Do a variable number of XORs into REC80 from T16KDATA.           *
*    From 1 to 10 XORs will be performed.                             *
*-------------------------------------*
VARIXOR  EQU   *
         XC    REC80,T16KDATA
         LA    R02,T16KDATA+80
         LA    R03,9
VARXLP   EQU   *
         TM    0(R02),B'10000000'
         BNO   VARXLPA
         XC    REC80,0(R02)
VARXLPA  EQU   *
         LA    R02,80(,R02)
         BCT   R03,VARXLP
*
         BR    R14
*
CLOSFILE EQU   *
         ST    R14,SAVER14
         CLOSE (INFILE,,OUTFILE)
         L     R14,SAVER14
*
         BR    R14
*
*-------------------------------------*
*    D a t a   A r e a s                                              *
*-------------------------------------*
         DS    0D
DWORK    DS    D
         ORG   DWORK
TODCLOCK DS    D
SAVEAREA DS    18F
*
ROT_REGS DS    6F
ROT_WORK DS    F
LASTWORD DS    F
REMQUOT  DS    2F
*
SAVER04  DS    F
SAVER14  DS    F
EOTAB16K DC    A(TAB16000+16000)  End address of TAB16000
F15921   DC    F'15921'
FULLW    DS    F
WORKX    DS    F
WORKY    DS    F
HALFW    DS    H
*
INFILE   DCB   DDNAME=INFILE,DSORG=PS,MACRF=GM,EODAD=EOFIN
```

```
*
KEYFILE  DCB   DDNAME=KEYFILE,DSORG=PS,MACRF=GM,EODAD=EOFKEY
*
OUTFILE  DCB   DDNAME=OUTFILE,DSORG=PS,MACRF=PM
*
KEYREC   DS    CL8Ø
         ORG   KEYREC
SAVREC8Ø DS    CL8Ø
WORK8Ø   DS    CL8Ø
REC8Ø    DS    CL8Ø
KEY8Ø    DS    CL8Ø
TVEC1    DS    CL8Ø
TVEC2    DS    CL8Ø
TVEC3    DS    CL8Ø
USED8Ø   DS    CL8Ø
         ORG   USED8Ø
HDRREC   DS    CL8Ø
         ORG   USED8Ø
K8ØSHUFF DS    CL8Ø
*
IV       DS    CL8
RUNIND   DS    C
REC8ØIND DC    X'ØØ'
CBYTE    DC    X'ØØ'
WORKBYTE DC    X'ØØ'
SAVEBYTE DC    X'ØØ'
OFFBYTE  DC    X'ØØ'
KEYCNT   DC    PL2'Ø'
*
HEADER   DC    C'!<CAUTION>! Data encrypted by SKED8ØF '
HEADERL  EQU   *-HEADER
*
CONSTEXT EQU   *     This can be changed to anything you want
         DC    CL29'"To do is to be": Nietzsche, '
         DC    CL26'"To be is to do": Sartre, '
         DC    CL25'"Do be do be do": Sinatra'
*
T16KDATA DS    CL8ØØ    Room for 1Ø 8Ø-byte values
*
TRTAB    DS    CL256
TRTUSED  DS    CL256
*
         LTORG
*
         COPY  T16ØØØ
*
RØ       EQU   Ø
RØØ      EQU   Ø
R1       EQU   1
```

```
R01       EQU   1
R2        EQU   2
R02       EQU   2
R3        EQU   3
R03       EQU   3
R4        EQU   4
R04       EQU   4
R5        EQU   5
R05       EQU   5
R6        EQU   6
R06       EQU   6
R7        EQU   7
R07       EQU   7
R8        EQU   8
R08       EQU   8
R9        EQU   9
R09       EQU   9
R10       EQU   10
R11       EQU   11
R12       EQU   12
R13       EQU   13
R14       EQU   14
R15       EQU   15
          END
T16000.ASSEMBLER
*————————————————————————————————————*
*  The following table contains 16000 hexadecimal bytes derived    *
*  from the irrational number Pi in order to simulate randomness.  *
*————————————————————————————————————*
*
TAB16000 EQU *
*
     DC X'546553070217359BC69D8ED3EF12BBE09F319DA70'
     DC X'AE615AF507601610B00142577FE63202455D09CB'
     DC X'EAEBF71E9A4084BC3A4065737EF8D0F12D7E5C44'
     DC X'13F72140011EC4B2661F54A89F11FBE5C0971803'
     DC X'9A7F1147691C74107FA4ADF18CED613DA9BA6417'
     DC X'E1B5B9BDA1A62093110162F74CE40BAC210ECCBE'
     DC X'B24FD505D6ED998191AFE333F225A4D2A1122F61'
     DC X'91823DE415BC222AA8A341D66F5F186C0E6D9122'
     DC X'FE1EBC528068072312F453EC509643C1574C6611'
     DC X'36F27C351E85B3E7BB91459542C620EB7176043D'
     DC X'BF70664D942133697551EECA2EA024A21F292106'
     DC X'A6DE1E200BA3B6617D2F62A171AAD7E616B40B5E'
     DC X'52023CD18AEBF38E31F8B384F61C921D5CC56B07'
     DC X'710DCAB6721D011714E1C3AD849B1B6E96AF0578'
     DC X'6A14586152B5D1157507C93EA1D15719AD490B3F'
     DC X'A755D5B7EE64C08203013075164845B2111C6469'
     DC X'881352C707112A05F15DB1816F9F239A56D05FF1'
```

```
DC X'087312B7B2ABFCEC60192FD73E4A9C75BD7EB9C1'
DC X'F870914740AA791D339E07615E3273F31E54E009'
DC X'51C91DE6A7164963CF8AB655CF9451D866F20DEC'
DC X'80D322ECE22826EBB7B34660F23224D7F5E1318A'
DC X'0ED80580FF1A05E310BFD83F820627A63790A113'
DC X'E2E348C1EA907100150AA2A119B3F32129AEFE88'
DC X'394D884D81F1930BBF14C49921A10EB1E58F12EF'
DC X'7B65F198FDA86F211EE18071E74BAD4212668007D'
DC X'72400733D9F5F4AC24B0920CC1FF9F962317BBA2'
DC X'CBA118BFAC36239946885116C20E8023328EF931'
DC X'CDC5EF7718200C3817207465C9A6ADE0022C315F'
DC X'16217CB93681C052D25918988B7AB1E5CA156797'
DC X'07172CD41E6DF81BED31AAF220E7BD1619736297'
DC X'19442855D17B1752E980FDE1881115D9D16AA940'
DC X'E7114C09C1271FAF78C3C15DD5F3C421DF7E9B71'
DC X'BF96DEABBF6DF14920 7C4D88CE1868991297FA66'
DC X'2CF69287C17BAB8C421435F2EA1134DCA15D1168'
DC X'6A8611CC9EF9D01E6A786D116E490FA8673ED973'
DC X'1AF3D8E0075A24424602DC2351393C31231563DA'
DC X'B7C1056159572C12913D10F1158F020219F3CB19'
DC X'E2A8DB1225822AA8A273DE4E15313B2CB1D7B05B'
DC X'51163CC9DEA251F14A16ABDB33F1ED81E3161E58'
DC X'1182FE77DE078236A6440295C2053F1B8A7FF403'
DC X'224FD16A09BBAA87523DE9E7D9058972832F3D67'
DC X'98BA5411C16821120653E3941049629E7116CF38'
DC X'EF1B8D9AFED339B8EC52518D10901D114A761234'
DC X'3418EE23700339B24A3A26B2249AA279C1376D8E'
DC X'7D9F42B0321F99A3A0424E9FBE61D8ED812113C3'
DC X'4EAAB127CA1D6512671E2D755392826188F1B5BC'
DC X'81522C47154558653F4BC43EB16446528D214477'
DC X'45922968FEA619963444EA55A42D71FE2BD3EB14'
DC X'19F8EC025268A6881F83F69E010A6873E1E4AFF2'
DC X'DE1752F250E2FFD3924105B0F82C5163D71B1E19'
DC X'787B1559796044D41D87EF885520715C979951FB'
DC X'931C52A9CE3A2F1F804C15AA09FF8213789D38F2'
DC X'1232C6AA13776CB9A18DBA943E180D7D694D3875'
DC X'7611CDAD3C8B14D68170867E7BFA559DC41A3233'
DC X'544A14380B1C801DB1D2BA2151E829E38858517D'
DC X'1663B9DBA201AB27E8224E256596A3EB2A221905'
DC X'56DD192EB2A45186B5078DBA8CB9F5197195E6F8'
DC X'70E97A51C2D9B2BE11CA5215B1DAB33600DF4AFF'
DC X'FC143919FFF237D42D9E3EA724EA1B688F11F500'
DC X'83AC27F504F0089F364541FA43DCB78B0341B15F'
DC X'D591F0164DD99B227BCE36763F85AE32892608C5'
DC X'F93D51205D106CB2653BC15BE3389213337DA21E'
DC X'0DC28571E83BC2F3237790B3417A109934265EC5'
DC X'631AC136E0724A74C697239EDB693550CF48D123'
DC X'B5E5DE183161D99152858FB02371E1D2E137D3F0'
DC X'A922A686B891725ED7FB18BB626BB317EC1A9224'
```

```
DC X'D2E1908217432718A491B47124974841CD03DAD2'
DC X'168DAF26396366EF0A9EAB9E618CF7DC4CCB2E73'
DC X'7F1CFCD55618338F1EB228C67683BEED60A188AC'
DC X'71BD17AEF5966F18D55433D54D891EF147C20176'
DC X'57E4FF1C4B4E523157EBDAB11D515B187896C79A'
DC X'F1C7233B2BBA45F5F013DA6637E9C14DB6A1B466'
DC X'6EBEF744E8631804657DD13A286F2A36B0F68C1A'
DC X'EE3763821DE5B551D10430AC15BA0BF6ED1DD35E'
DC X'41553E6A3B159337C621E94D9A65406203E19002'
DC X'367419003D2DE2041679AEBBC43118133 0898055'
DC X'5B0EA12D18B4420124A87FE32498C69AFA8DEA51'
DC X'E1C7D9F9C7214677FDE2163FC7EDB82B7A2879FE'
DC X'F74E1A6B06FED18666EC8517BC9FE551E8B96350'
DC X'10B900E83A89DADC38B08CCC1242E2A5D1155B15'
DC X'F2D1EB16FBDD244EDC10A15BDAECD452BFD3631E'
DC X'BAE53151FC17A191115D4EA86BD8E840F03FB70D'
DC X'7DE2DABA792DB2C31C640C62C10BCCB43716AA63'
DC X'40523E732BCE2379B70FEBA2DA1927F0B56E81FE'
DC X'F7DDCEF9B3B4A017F08B1521CCE112D4A7544983'
DC X'106EAAD8E1EAB2653A646384B7DDD616C18CB664'
DC X'ED118CD01A9E051F24418548A955A0728E2811C4'
DC X'5EF3A15D7EEB791FE1A40F9D6F8546C2014D80C5'
DC X'C79B66AA43BEC96313D6433261DF32F7531D5845'
DC X'0181F7F9961D9EDEE81216E2D855E15DBDF56A15'
DC X'17236501A15FD21816077EFABA3E196D414F1D92'
DC X'6D233A11D4F432CA9571FD75F25BE8F1913A114E'
DC X'A83DDA16CDF4E17A16DC76153A53CD31E22963E1'
DC X'1562BD1791AAFE891678784B7516F0DC27978618'
DC X'724D9D74EB12FDC2D233B0E5B03E030330815243'
DC X'B3071D2AF3A961D06C97F3B2959AA71BA7FF59B0'
DC X'8E2BCC104C8184516354A4D522FABA6D71D72203'
DC X'641509E0F63766DEB44AB1D10171C9DE36457C03'
DC X'0BB11C00BB2E8FE69B061421024E31FC76D7D41E'
DC X'509B19325BE21EEB19D266F1AB7E76B81CBF3624'
DC X'312281CAA9127029D82185DA11D118BBCE23B246'
DC X'4B2A085740A00BD726FF202233756B81AB26D592'
DC X'211C43CBD989AEDB4119655BBD7E47654C714058'
DC X'3816FD4FBD922528F507592A634C168960E76234'
DC X'9D3D90FD7512D115E7D9021B61BAFC01F1F0CA00'
DC X'1700D3CD9172DFF72321360DC3A24AC6FC75396C7'
DC X'38C1CA8DD4A31D13D3F4D10AA86152FE1A33041E'
DC X'2CA7E2FF6528B3115375481C15D8E485D178763C'
DC X'6B2373DDB95B14632B111D4F9D8592F394083A73'
DC X'F1408E06BF7317D7F9C83F09FF293593425E2371'
DC X'80FCD19BB18553FBBF677A8146B8FAE26A631916'
DC X'1EB196C13159270A16693119114C7E1008C75A9C'
DC X'791318CB425CC8801671CF91F35F24E7B90BA1C7'
DC X'5DFD251076C8C05CBC9B8A610AA38862BC7FE90D'
DC X'2204C6F99A5907F32E71B8FC97BB4F421BFF9288'
```

```
DC X'447A39E0E39633CAD9D1E782F20E330D34BCBF09'
DC X'F017125C0AFD6F35E1F177B875C8A71A157B1CC5'
DC X'C179615D419E1E1D75CB1CAEF48ABBC174510DAB'
DC X'233A10E2B1E4D42F7AC86C52F71F459DE029144C'
DC X'F4BB9A7A2FF9B5A3916193EEFA4F2425987E34DF'
DC X'2160B185A9BE9114A6CC061CC86B1621A40FAFE0'
DC X'1D02745871AA572D95194C2F01B1ADF46E7A1690'
DC X'698E6226F790391F1BA69EB4C88EF1072C71D562'
DC X'95F54F87D6133211D02180C1351142B0144F7188'
DC X'AEA7F1352506A4B77239F97DDA14C07E372127F3'
DC X'1702142D35D89DD9C2541FFB1653B227027F6211'
DC X'BEB1D01EB8D10C61790 9AC511E48B25D3809E369'
DC X'5A2CDD561FCDC12192F444E8A3703DE37C50AA06'
DC X'174CCC20C4E66CB6012E1F546CACF3304E1DA001'
DC X'693FD744DB71439A73EF22AB53D5124EEC345C10'
DC X'20385F42124448E3234CC271617F0A6ECF977164'
DC X'821321CBBA0230EF1EC3C1D438687C20BCC520C6'
DC X'EF12F1A4B7171D222713DC22421B21FB7E2E6CD9'
DC X'A5FC31D8198F03F6BE0B9557D626CD8941340E5B'
DC X'24D0D93282B22E816910D3F4F2A4C55E0B56A5BE'
DC X'AA974D5138C16540DA07B3D583501149E92D2180'
DC X'6C3CD01717883C21FB983FD712D2E68C3208FDD1'
DC X'F820D9D2ACE881914BD1CF907C722A7349E31AA7'
DC X'B72B2D3540C5E1EDB3C6B8521D59EFECF05D25EB'
DC X'AD04ABEAA07047112E57BED1FF7FE274152DF61F'
DC X'D204EC344CA8BBAA53A88E359D2DF889E2C07F3C'
DC X'55217CEB883192121 98282268D4E80B6F6F20DA8'
DC X'BED71E3FA52B1CB1659BADBAAD7B3739A921E71A'
DC X'A74AB138A4EF41229C8EBD23A4172947214505E1'
DC X'52932F472495A6536D7DE1807277486B10F6E613'
DC X'5DF8979823C128EE3E26C431A199350CCCC4A576'
DC X'8436D2468F8F01D6189A7A76A20FEAE45110138D'
DC X'B681569341F6C614848D2B8546621EC9FC3E1994'
DC X'3043925192D3CA158C5C3ACA326D875209F7605B'
DC X'1D19565C6C4BB11A7B6DFD20220A6EB833109DD7'
DC X'C18221F9C0D51B78A388D640B462766CB5E2575D'
DC X'6FD41D5BF18F1FDFC6ADCCF58579721B3C7DFAD8'
DC X'A4709312936B846150182E6512269E3331C50FEE'
DC X'D521F37B61AA177340722E0508275BDEF036123A'
DC X'2AC027905CE3116434C771C5FA38C3116A9E5EA1'
DC X'C34CC3CE75250BD4EA08341051469A772C3D1AA4'
DC X'BDF9FF25A53C436716E79874F20C135DA71325BC'
DC X'C9D852A8A37D736084AE8C130A01872B05A9160D'
DC X'3E1A9132722B8C87B734401BB2EA3B8222066EEB'
DC X'1E7A27C4B18E2228B33359F3DE1095E12F122712'
DC X'5CEE8D29375E4FEB9E0EE11BE2217B78C5ABF0FB'
DC X'B3352010E49ECCB829C9F135556AE0136EC28472'
DC X'419678C671E55F7E107AFA8F62223AE4071247E7'
DC X'B8A1818C94552307A737EF1104FC3C70442E9FCB'
```

```
DC X'883ED188DE2A5C14B4BA42D128518FA4B824BA0A'
DC X'E9D01E446C4445B172C401BC133AB73E21F9302E'
DC X'73248D159951C92A4F206050E08B4886E5A21EA6'
DC X'29F821039B0809221877C1A146319247EB51971A'
DC X'C0013F41B048DD5713D9C2E2EFD3F2DF615B7B28'
DC X'2523C1D08D484B28AEE209388224871C40812B5F'
DC X'3D3620BD154EE7CB4C7F5ED8FD56C19C2915DA1F'
DC X'62BE517157FAE7B21DCB8CEBD1EC079D3020CAC2'
DC X'FFDA4F447217DED11E0AD723F5112DE9D099DDA0'
DC X'AA6F14152A32134AB4D901A66652D186E8610A1'
DC X'CA8FC0351A804F56E1B727E4CD111728CD6167B5'
DC X'227E239F0B153D730F69316F4177ED1F6F049908'
DC X'A21D653235144D7958FF84311C7112769735FFE7'
DC X'F835ECA94C984C69422CE8A22496579879309D26'
DC X'C77BA28283FE3E43EAB1BDEA1043EE940141E030'
DC X'514626D9D6114DB96EC14896E6641EB6B015F615'
DC X'A3D661AEAA9B3E6C907E43339A21A959FBEEC711'
DC X'D118DE3251829D6A68CE33C91375F7898CB2208E'
DC X'B24B0E7D79E39EDFBA117FDF5AA186825F2415C9'
DC X'6DFBDBB52F3872189D7F4D10E651F4CD600230F1'
DC X'03442D16663884F59337B715CC3639215BFD444'
DC X'92450D6CF81DD4411D51C61FAE7D88A94C5524C9'
DC X'B660C1B12D5F0D9845C6551B28735048C77D8219'
DC X'FBDEFE4B45562FC946C20937E970BB93652FB23E'
DC X'616C3821BADC2A241AFF6ECDF18E7BE44549E0F8'
DC X'AE1A5F2C9319F594F0A1075E8C6E17B523E701AB'
DC X'3FA6E127C97761229B4C6351C3FE31A7BB4F0125'
DC X'19CAC0EC21C9131D039B866171167C2FE5E227BB'
DC X'0E681E7C7F6AB8DEA194123B8C14C7D9C3E52CCA'
DC X'C8F110DFDE557D1793A2AE910DBEA09810B2CFD7'
DC X'F209D24D9A6DEF422613E2474CD19BD53FE759F9'
DC X'8AA3E14EAD2D7BC72FD7EAFF275A241EC68F3151'
DC X'384F9A14EF56922173B5B77D80EFF1B419C988C1'
DC X'71236B8E5ED97069B71E10B39B1132A4DAEE2291'
DC X'358342C09C515FC570AC716AF86950196A339918'
DC X'601BE396ECD896636BD865FDAE1A4D8BFA31B142'
DC X'296E417278C91BBEBB472CAB663A71405EEF415A'
DC X'7751D7B251874642E34FA5406BA944CE62EB559E'
DC X'FF7F05E854AE4029AB5CED3100B6A8126D442FD0'
DC X'DD29A8AF59BDB528E955FDFF9BD7E6DF3073D39A'
DC X'C6C0748196D0BDF7D202A36B142ADBA9E43F0414'
DC X'86DD3ACF6A290DF121C5AD802C7D91A89FEFE14B'
DC X'0815E3019D404827B2BB349D3199EE83DB7AEBA1'
DC X'B9DB915854A6AA3616BA9DC349E3C31F0B41E027'
DC X'E51231E2813E60BA268B54D07849773876113 9B4'
DC X'CC5D1346CF7B99BE8827BA0916A32DAD5F079DE5'
DC X'07400037C778E2A011FBCEBF457CD35129A2CFA14'
DC X'0639C98554CD6B9682F28FE95B6AD6F0DF1BC923'
DC X'6A3091C03D5714EEDCC2A8E7D419B03F74E49846'
DC X'F25F1803B8DFB2FF8A6FBC15872CBB3A4277965F'
```

```
DC X'C2A386EA959DBB930CB37AF3D0479F1A85D8BA3F'
DC X'04ADAE3AFAAEF67DFF11CC7841685298FB1C5B47'
DC X'41F097A1952260187224873930 4A0FBE3C29ADAC'
DC X'1AED4500220A52974F7C9D097CE61A5C91B63C2A'
DC X'E2B18228E0B3ADE09EE4399AA7AF0D447C749270'
DC X'2C32B78DF16EB8EB25D6B198DB8FBBF0056D9EBA'
DC X'922549AAED75DE2B08D0F5FCA8E056F77F245D88'
DC X'AC39FF87BB7FD008F5F54AEF44A321C75C7A4C6C'
DC X'E7CB62FA612FE01FBDAEAA100E5B6BBFA31F7BD5'
DC X'E3D2F149187E4858B1AA81B43D0646BBEA51EF84'
DC X'B4792DE5EE5CBE46E2A36DA61DFD72650CCFB292'
DC X'3BAEAC56DD0A3B49851DF5FC2C9B202E5D652260'
DC X'3E25C83E2DF60AF9BD3537A4F7F8E37688CE0011'
DC X'10A4F4542567D27D9FDC0BEC02C2F59DA7EBB255'
DC X'66348F0FC09B1C66007101064004B3BAEF2DBCFA'
DC X'97C67476C9C938196DFE3A1F5F65F285A25C6CAA'
DC X'BBE83DA26AA8E3D6B51CFF38C2D2196662976AAD'
DC X'ADAAB5DF979A85B55152C2AB478C0D03B015851F'
DC X'4CBCCF9D1C2F34C561087062962C7F11510D0D39'
DC X'17905DD0EEE737B7DC3C9407210BB4C7C5B2E33D'
DC X'E637DA4FA6824136062F71078CEF0E8C4F395412'
DC X'436AE210D3D8511D7C43C1148C7A88104A3369CB'
DC X'395C7F9A33B22FFCB46688C51D039CA9F3BA128A'
DC X'96AA9AC3B19B3FF2447317CD7E30C9EF105E8998'
DC X'8ABAD6442644C1CE057B68B112E80B372ED51C68'
DC X'AC395E8FAFB1F109713BBECBD1DB2061560D0F41'
DC X'6D76BB6F89C45A200CAB2B2A30DED3C3782A27F4'
DC X'EDB718D4FF6FC554E761114F907BA672895D0AD9'
DC X'8EB9E1902B9BA080545AF832C8D41E97CF6D0641'
DC X'374952FD7F5905DB8CB14EAB9134B1A4055D3000'
DC X'25A0E3BD21D7BE630E2CEF034238EFCBC88A66A3'
DC X'BE7BD14D1606225FAE670AB42A6C8F8E739FBBED'
DC X'80ACE7A8315D08FABD8481404FC8BB5E964635BC'
DC X'06F53DCF49108410FBD83CA477B1248077798F3A'
DC X'DACA9B7D5C0E53D4C689D7114C517EA3494BF859'
DC X'12620AB61ACE39BC86011647D4AFB9D5A4EDA2D6'
DC X'2ECE7FEE1C8088F65401BCC2C2D5D445E001144F'
DC X'A4FC37CD7C9B9767A321642EF378372DFA52C1D0'
DC X'125A3B4E15506E03473A480E29D5940A10FEDA28'
DC X'3B61DC155DE96FD9E5AFB2FB5A90F69F78F6A210'
DC X'7A0E92ABA84B7560FACB79A40CBB86A6D26A2ECA'
DC X'3F8D96097723413FE07BC94492FD686CC69412FF'
DC X'FC9ED7CCFBED169D1DE168C02FD04C626D3849BA'
DC X'BC295EF23BEDB1D452B3CB1220B84E3FBD33FF7D'
DC X'69DF0AF304CC5AC56104A4FBF9093E2653E78751'
DC X'5BBA4CA2C0D5D780A216F4878456F38BE5E46D3D'
DC X'FC67057EFADA90BFF757C1B79E310EED22D62C8E'
DC X'3F7AE8D65258712F5CCF15C5BFA1203CFA90ACBB'
DC X'F6CC29D599A62EF86DB500AA3D61FEE5F759D0DC'
DC X'D6602730C4DE0551D73231D4E97E4F93445B2A9E'
```

```
DC X'298D1F9C4D7478FA2B8AF357CE4A952D78E7C32D'
DC X'DECCB6FC8229FEAE5F1EB0BC88DDC2F22CC34D92'
DC X'A31C7E9B27AB3681D6BF10F91FC7E118CCFCA57E'
DC X'E480D74B7E8DD30779E58A1CE34A6F95A3B373A'
DC X'9BF0D8F0FDF8A6558743766C95AD1AEDB1FA3F1D'
DC X'C9B5737102571D4D3B1D70E99C0F944E65AF7658'
DC X'56A7E8A94D4267D505E8CD386F6C835F963F5B29'
DC X'51DEDB889AE769013CAC7B54D4C95D5CA873A8E1'
DC X'7501DCDF4A40E9E6F4B36E74C45EFE82A861A728'
DC X'E92D11EB0447F98B490FC1876DD4B6E4F4005377'
DC X'5A32E38D550933238420B22B1C71055CBC0E384F'
DC X'D407F89C9F6474B667C447AEB5D233B34F66F4A5'
DC X'CBCA22B8F44E31E6E5CF09BF62B59C34440D68E3'
DC X'FC8A5BB608BB79CEEE4F4723E04114351162E4F6'
DC X'26DE082555752CF13F6BD93BBCB2CD72A834F178'
DC X'79C44597121D33DCEA7BD53E463A79C517F6776F'
DC X'8754E982D3047D5BC080D956E9A310499D9923E5'
DC X'E02E6EFA8203E31D6E50D365F4825711B6F639E'
DC X'55E1F5908EE1F07FC105D82E2065ED5750D92D6C'
DC X'C2BFF7074D0F9BF1DDC476A08A78A1F7BF2460F3'
DC X'76BC9B81066124FAF608F5A40CB2EECF9AFD7B1B'
DC X'0C561FD2C6F09627689D421CAF42CB8A9B0047CC'
DC X'A6D511FFFCEDDD792A4C604A2EC5703986726D03'
DC X'729E50CCE49805ED2E6BC8805C8DCCEAC89ACE7'
DC X'FC03B3F1E0C8B93444132781B8237F466B1D2DEE'
DC X'CEEFCB34EB99B89D5801067B03FC8F6DC3E1B03C'
DC X'7EEB72EE8E931D9A9B26083A7FFE5A0970C1C127'
DC X'3830B3679231CFDBDF59DCA5702DD75CC3753DF'
DC X'58B9918F4123F9022F775096B18B31ECA68C271F'
DC X'AB1A73C28A918D5112BF5B47CFA2CF293B0EE5DF'
DC X'3FA1AAC0981CCC036EBC53A2D22C4BED9F8FEC37'
DC X'8B599DDFD9E82437CE3F99C1EB09085DEBA611F9'
DC X'F0FFEC3E54918097D71FB93EF29D59076DBE196AB'
DC X'95679CB0E4352279B6063B11DDA76D3306A50B79'
DC X'05EB4FE35BF38FE2E95314D3A0819148DB6ED655'
DC X'63C0D4E825C7A3A75C248C0838B492787D313832'
DC X'A176627EE588F15BBDEECEF4D243993D69040904'
DC X'6D7C5994D6DC470258FA1F4F53CD9B610E148279'
DC X'8668AC73FF0F6AF2B1BB9FA41BF73232054FCE979'
DC X'AAF33206484 0CDAB4900971D062F96731A6A4A6C2'
DC X'7D3BAAAE463A8FFB704C257E36257940A1232E87'
DC X'458EFE2563F934FB493082DB59E49F30883761FF'
DC X'D70C36370F31A190B2F649800318D0E05D3AB9EF'
DC X'F8FAAC53D686F65712CE81846EF3CF5D85F991A3'
DC X'3940A3FBB7A4FD242ADAF0C6093B950F185FD32D'
DC X'8CB1264143E0E01990DC4F06C46A09AF90989606'
DC X'25C89470556E546CF9FC602557E6DBF26CB2B430'
DC X'324FB2A64982E17B4BCAA86B77B8D1EFBD70F911'
DC X'E9345CA8448CF639ED312180B0C71DD62D6A50E7'
DC X'9EC2F2C16E61CD82CC965E07C5092095 0EE9310F'
DC X'8F87DAB114B0D88B583F0C192475F1A79892A13B'
```

```
DC X'24E7A0490A907EF726C6C36078349411614B26FB'
DC X'24E0FA9420BBF4A0BA78E54D90494DB2A8E4DF7F'
DC X'CB6DE85E29EF086641C3EC2EF2E15DBD8889118D'
DC X'266F2E6A66839F456E3E5E9A6A74E30304A29D39'
DC X'706C18A0E342C00C79F03D403C92897BACD2E6E7'
DC X'311B590720C55377117CE4BFC5751E3A882F9ABD'
DC X'851D55D8259F8E45923465D26D0D9F01BE7305D2'
DC X'45CCD23265FE7802F2C58C8A1B31453817D3F6FA'
DC X'E51F93EAE58D17C20FFD2295BC80882AB293D981'
DC X'66EBD7BD5A11AAB3D20D46C54E71A9CAF9C25A85'
DC X'9818EC74CC350D0F21518E6030E9E94A2F9F5CFB'
DC X'1151EA135EC8434FAC8EFC06CF4A5841D7833C6A'
DC X'F6E03ACF760DFDA85877AD1202F504A60832098A'
DC X'92EF4FA66CA77083D9C809C5AF8D9517BD0C729B'
DC X'CEC59B0B55C784500EA65248CC8E04ACDEB3C143'
DC X'9278EF5C05CFA6BD19D1821B62E41EA9395AA527'
DC X'736B14406FEFF852E5E8188EA254F8E83F1D44BA'
DC X'84617E34D1EED9754832634E0EE37760ACE42CFC'
DC X'FF39A6A0894B8EF5F4679C64107D602115458841'
DC X'7802F916ECA23C535CCDF860A5FFB23371BC4786'
DC X'F300478B0255FD5E4934BF19988473864060 5776'
DC X'E896E9EE17CC9F2E7717B5B91325317EEBC57C58'
DC X'23FE46379B2BBA495445 0061A89722B2DF8A3425'
DC X'59ECAE3B378FF5D1C0104E330A5C2E64A1548F2D'
DC X'72BD117F19EE44FFE3A9E2148549425DA8E45CEC'
DC X'AC241E898F5203499306F31B98F19FF0377C4D95'
DC X'B6A9F0314F5487CDB59FAF8D736BE57728BE17FC'
DC X'2A264734B74509944C2010847DC9DAF48BE530BF'
DC X'CC2A901DA62F1D571BD55CED10460C3A12A50EC5'
DC X'0339BBB83ED3DF6FF030F4366DEDBCBE550D7154'
DC X'ACCB93EFE803382840 4DD833F8674BCDFCCA053D'
DC X'90E38D491C0B9220B512FDFB9C0DC0A2A10AEFF0'
DC X'2415DB439A7BF9FA6A5A0C5935E9F4F26FE36A4F'
DC X'4D856CE6C91F72EF3EA536FEE0C9109715E22CA8'
DC X'4BC19C78F601354C4023B694206524E4071AD730'
DC X'3073A1B291D31026780086F02AFBA13D852E59F5'
DC X'8E79A936152C1CB300F25511D9838A880DFF323C'
DC X'20594FF383EA4D58BA540AD1B6364DE6E423FFC3'
DC X'8F920529949BB7B3C33FAF8C429FC37D1623518C'
DC X'E5AB7AA3A3E205C6529FE74069E85E7B5C4CBAFF'
DC X'30A0AA91BEAE21FD8575725560E6E821108E791B'
DC X'20BB610A024B6F1B291C8E2F5230BF4089D266BA'
DC X'DC84FFF875D518006EFB7FA185E0F610C26413CD'
DC X'C055AE71889B846C475A289A7B2751D90857701C'
DC X'8D7D856B15AC31D21535619995BC6C2CFA25116C'
DC X'356D26773D543733DD01604FA0541FDFB27FF335'
DC X'709FBFACDB2332FBB3996B30EEAA8E14F8E5A10A'
DC X'64399F59CF37B0AA169C42834934D7C043CC0740'
DC X'C74FF54A865F7793D0C2593C664C6A3D5BEDC402'
DC X'5118DB173A4236913CD6DDA52C3E2CDCB0E7A15C'
DC X'F94CBB5173B699C1067C9DE1CC5A0F9C4672B60C'
```

```
DC X'00D17A8DCF0860F5F122B00A2F2176DEBAF38434A'
DC X'60CA3C2DF62EA51342970F208C40F3B5C643C820'
DC X'5F3506329787507C1B82CD0312C725531AD1510 4'
DC X'24BD24AF6C99D3C173410C5B55CA91436EBB9E9D'
DC X'B68718FE293B44A0DD3A0E02D2E5439E188312F7'
DC X'B6391409EA9AA3A3DDEB99948836C4B1192AF954'
DC X'606358A075268A10D4E7EA9EFCB291494343CCB7'
DC X'205138004EBAEF5F537097787F992441B151976A'
DC X'E1728B82FE66851664F13DDF03D03B1178AB0E61'
DC X'E1315DC39E3852FAA5A79F74B79A4DAA0830BC68'
DC X'08BC0E14FC4679FC17C9863B848F246CA164628B'
DC X'EB412699B916083BA44A607A8287F6C03BC54626'
DC X'C21964A3D84706A2FFA952307650BF1B56E10E2E'
DC X'81BFF3C317F04B117E05A6536CA863B7997709FF'
DC X'63234A114E7556F2D4ABF0E1A0CD4A697A9FA603'
DC X'516A7D38DB35F7B8C1DD3B49AFF863F05028DE8F'
DC X'EB4A053A0B8EDAEF9468A21AD2EF82D967748B3E'
DC X'3C632EA60D3601C26206 08A0F7CDECA11133972C'
DC X'9204DE7D3607F6B96182D7D2E048D6F6110FD1CC'
DC X'530D89A1ADDD22367F5E15CEB0FF6E361E360BD7'
DC X'C83D523E7848B75C09F12B2B6D55AF9736541118'
DC X'93EE89F4772B44EDF6E781F86B5DCDAA3EFFC7B9'
DC X'DD89ED53644EF74992728C2F1398A62E1CC504CC'
DC X'DCA2C343BBC39F017DA7AD456CF29D31E1B95842'
DC X'DBA8CF5845EA28B4FBACA7E193BAC1F99CAF76DA'
DC X'45B4CA8C13D468C34F79E17B2FD8D112F0F4BA99'
DC X'4E8E59481B718B29920C7010FFCB78B1DC4522F7'
DC X'C4E799141C902895211E9534A047F2E4D9140FE4'
DC X'F46B5C10E1DA30E41401DEEE61A93523634FC261'
DC X'C93E6EAB014A6E1DBD36E38848D2FA467E4363F5'
DC X'5587BACC14CFED56B58476E86A1C7F24819D048E'
DC X'992B8F2FD3E26A13D68FBB77F69B0DFFC23E95CB'
DC X'D8E4C3E4BF065EFBD49E1D61DDBDE5388ACA5ED5'
DC X'E94DDEE0DDCCEC562EB41DCD97D5F25AD9344587'
DC X'2F5AD104D5FC781291D3D793C4707E43D13C009F'
DC X'4A373A56D2777DC97705E06937EEE923741B1464'
DC X'8E26E73DA8FB487306A37C7806D31983DD584DCB'
DC X'0C797C007EF848F1F54E6DFF9BD7E6DF3073D39A'
DC X'C6C074941E654D2C320CAD77011D0D2E5E429761'
DC X'14B5A8F1E8BB97B49C59E012B40735AB4516266D'
DC X'13449273227440F0AFBFF4C827E478257032EA0B'
DC X'25400497CEDBD12D144B5D024273C31F0B41E027'
DC X'E51231E2813E6E8ED36C8DB0DA27DADED16668DD'
DC X'F3BD9AE4BE80AD33F4BF956452C45FEA8BDDA201'
DC X'ADAF05312183AB9FC08E79B3046758CA9849C3D3'
DC X'9DA9A6EA0A41B26300 53A1E311645497F1BC9236'
DC X'A3091C03D5714EEDCC30A4438DE93CB798A1BBF0'
DC X'A6E343FD7080BB1DE59DFAB40887E85337C6C0C2'
DC X'15AB7AE9427049E332BB9C41901D61FF743DDABB'
DC X'0BB2CFF982ABE727E32C1A6D246E262D248DE474'
DC X'1F097A1952260187224873A80A31353BC100970F'
```

```
DC X'26976D40127A944383E1807A0B5D2F2CE98B6D3B'
DC X'19E12B442308DCEBE859C1B8EC7E897C0A894886'
DC X'F65029B901765BF0D42A7D349A62600E7C53BEA8'
DC X'15C49AAED75DE2B08D0F5FCA8E057F683A16AFA2'
DC X'F93780CB18CB676D92C792AC0F2D2BE5835445BB'
DC X'53CF38D5DFF6268EA386BC7E6A00D93E0D0DEAAF'
DC X'12EB4C17AC3B59AA16507C93FBEF4CFC76C40CD2'
DC X'683E16CBD5CBE46E2A36DA61DFD72650CC11F12D'
DC X'1E00D9E23D23DF173A13B78EA5408FD97D186DD7'
DC X'23D4F885CA363FEA04C048493B7759D74E89E2A6'
DC X'ED11B97220 3A4AAA8CAE76ED91FA4167E527478F'
DC X'7C4EF30F372BB4600710106400 4B3BAEF2DBCFA9'
DC X'7E2D4DC9E61EBB27BDA9B761E8150A07F1126FE5'
DC X'8D87E3F372C15C358323883B4B64B10895AD88CE'
DC X'1E918EC84B5FDDDB2B34478FBC30F9D08B8FD091'
DC X'1B73AED1E24FECBD195A012962C7F11510D0D391'
DC X'7905DD01111036F23D077FF8F2B421BF2163068E'
DC X'26097CAB056AD04D5C305E7C9E6B5BF7D5CEDD2F'
DC X'4EA54B9052C810217B47A2623BEEB2842A2F6C2B'
DC X'FBC31E5EDDB5DAE60390C08688EA6CA9F3BA128A'
DC X'96AA9AC3B19B3F114E897850B0C39E1BABD6CA02'
DC X'E9927E058A88E2846C360A3A0C57479586DB125D'
DC X'E846CBB24B9C7C28B25BB713B5E2A155E7F2B868'
DC X'1723E796DEC014C8E11BF7CE07F04F31ADA9A27F'
DC X'4EDB718D4FF6FC554E76113C938E34292D4124DD'
DC X'BCF2B5EB645DF6339760E0D0F2E6C768B3439474'
DC X'6FD5A5F8B38911EDE1436E38C6C5C3365D0850E2'
DC X'38FD950EBFEA10E98E65C950FAFE6EA197ACD72A'
DC X'AB97BD14D1606225FAE670AB42A6E5B5729B8412'
DC X'C7474212CC473D013B1237585079F165B958DA04'
DC X'EC84CC1FB3C006F1133BB1B6FBD30B687F0F6301'
DC X'E617DB57B98E1D7DA6DEC15DBDDD36F856833CF8'
DC X'1BBE06D9A1ACE39BC86011647D4AFB9D5A5A208B'
DC X'CE9E4A0606A3CC40C82BB19E9DA6D853D8CF1478'
DC X'DA688CFB17D7E494C28C1D9E76365D3F4EFED3F5'
DC X'74E57DFD5B9213F9796AEABE22ACA66BC8D4EE09'
DC X'9208551EF1F0EBFD9E5AFB2FB5A90F69F78F6A21'
DC X'2D52F95EF526EF7463C6B1A16F6CFC421D961197'
DC X'864DA6DA6EEF24C60995C165CAE600D18CDD5059'
DC X'D23C109BCB54C87D102D2A9E351D481424F448BD'
DC X'16BFB1E6177F425ABAEEFCB1220B84E3FBD33FF7'
DC X'D69DF0C83D0A926269360F09B26D8A32BA7823D1'
DC X'B920CCC85745E0072A186C3E1B4B8769D6B9CDF3'
DC X'D7B7451803D151F97AB631C1CD1964F41DAD9BA4'
DC X'D9ADBA5D8887CEAADB3EBDF642FA1203CFA90ACB'
DC X'BF6CC29D599BDF2142B140C2D3756F12468FE1A2'
DC X'37E844FB04DCCC37C56B6EDEACA45BA1546B0EC7'
DC X'3C9013DA508F06EAA738A931639E5BAC0A9C4555'
DC X'47BE37DC6BE399C2180F9208D9FDDF99F22CC34D'
DC X'92A31C7E9B27AB36821A0EA1194E9DB5D4FC0673'
DC X'CF7C351595CBF0AEB6542A59E54CF25A25CC203F'
```

```
DC X'86D1502407B516F92559974ADC2131069CAE6711'
DC X'FBFA7C84CB00B3C034F981BAF0985F9AF541C676'
DC X'5856A7E8A94D4267D505E8CF1C5DE7B515723F6C'
DC X'91275FC3EFE08B67ED7303D31237D5B26D81F780'
DC X'62369C25FFA094E8EEDA358DA1CAECCAA0372F7C'
DC X'8110171423668F6DC4E4A34DA46043A6678B983F'
DC X'928E32E38D550933238420B22B1C712ABBD70A96'
DC X'1A7AC7591D480C4E6980A46CECDE11BA4C8AD120'
DC X'B05E579C0328572A4DA689EF721E0B81C3601F31'
DC X'AE0C77DA4377DA95BDB32AD597F67C2FCBB8BA1C'
DC X'A43290EBAE85752CF13F6BD93BBCB2CD72A83C83'
DC X'722DE9A971E7BE9E849F8B011F77C65F97046D49'
DC X'E75E69E71E659DD8DDF8D179ABED33EA18D2EAA1'
DC X'D452FFE1A3F0E6D65EA2634E6EF440B8731F9423'
DC X'E8F904C72FB144179FC105D82E2065ED5750D92D'
DC X'6CDE98BD4455A679D56AC601F7EE748F28D0DA6D'
DC X'429FD2D5A4302A94870316FB1393408EFDA2FD0D'
DC X'C051B8A73131FD0680D6D0E5B859BDEDDF81F7A0'
DC X'9B75C0B98B36356E416963B04A2EC5703986726D'
DC X'03729E50EA30A74490916360700CB57718E39410B'
DC X'D056A916E8ED0C9FC2C4B12BD7D3F3A0877ED6B2'
DC X'1C2EFE27E9909E656079FE8D6A3512469D78DEE'
DC X'7A49CC0B254304A3C6AEF7883246DE5A0970C1C1'
DC X'2738306B3679232122F522B067E31C5F2D734248'
DC X'EEE9C32C99CD692F6FAFA746A9E2CCC2A869D1ED'
DC X'0FA46C0404768F8A9E8CAE45662ECC84C588DE3B'
DC X'AC3ED2F27FC47BF2DC82712007F202756248FEC3'
DC X'78B599DDFD9E82437CE311D4D0804FDD3E7B38F0'
DC X'7A2675CACE2C278495AD1918AD7FA2D1C92B574C'
DC X'5FB691E7927D51FD56D213E26EAF2013497E9CC7'
DC X'AF042F03C854568BDE603C6925129EBD2A38E4BA'
DC X'E13C0D4E825C7A3A75C248C0838CE6445EA6AF92'
DC X'47C6348254636849F15356954B545A3FECD050F9'
DC X'5844D440E429B3FD258CA7B2EFB1E7403C7CE9D1'
DC X'5EC6D3D87ABD416DE66CA10801 9FF26545AE5013'
DC X'54E423696484 0CDAB4900971D062F96731BE78962'
DC X'F597E1E5900538E76DB566C1F93C87A2C415D7E79'
DC X'D055A624CFD1E4D3A775E738B7A8BEDD04272247'
DC X'7A0C25FBC186AD4DB6B29779B14DF016B9973465'
DC X'229143934523C265712CE81846EF3CF5D85F991B'
DC X'A900449D449766F72D92857D4BD7BA4390AC30EA'
DC X'74D4C9B459968FBB9461616024D79E7E7CD6FAEC'
DC X'DDC57CAF787123F91DCAAC99200857E12C16D68EA'
DC X'24F5C72EB9C66D8840FEAA86B77B8D1EFBD70F91'
DC X'1E93469E7F78C5FF7CFB1C74DE6AE09B06344D87'
DC X'52B63533E4F78A66E61AF8D260A5EFCDDF9679DF'
DC X'98AB9604619F61E26EA5417CDD9709114E4D9A94'
DC X'81AFCA96794AAFC9C0596AA673349411614B26FB'
DC X'24E0FA9420BDAA42503986595 65B8F5B94AF5AAD'
DC X'0F9477E096BF6F38BCAB7E4D8D507D4265221FB7'
DC X'D94ECD89A072097BBEA0A4A6D5764B5CFECCDE29'
```

```
DC X'F502819CBD8DCFD489B72E59BD6EE40BACD2E6E7'
DC X'311B590720C5537713FD086AE5DC1971510293E1'
DC X'6AB4405 1D34B8B20F9CC4E8696043D37B7AF2D03'
DC X'8F66F93CCB879B068A64A94882DC84ADE807819B'
DC X'EEF3A338797FC7090413A6A1CFE3274B85A74E81'
DC X'66EBD7BD5A11AAB3D20D46C610AF323EC03A5839'
DC X'ED3DCA4969B1374878DF9A77B300ED691E2B0A9D'
DC X'88482ABA2F9966225A0DE623D9C477DEF4CAF164'
DC X'E96C4F84E2678FF0B334BC95E405296845B01616'
DC X'93CE4FA66CA77083D9C809C5AF8D95D228192C0F'
DC X'C1415D32F2477B66B111BE34497D2B7F580EF688'
DC X'B5D59CDD1307530FB5EF5E49F006D8233F57B04D'
DC X'B48B5C77EAF0F0669E0A1C91D96C74D30C9298BB'
DC X'F9E1DA39EAEED9754832634E0EE37760ACE4C5D7'
DC X'11ABB653841CC45CD0E72F7993BD06E95E7E8BF2'
DC X'FB22F7EE8048DFF5E02346DB1684DA485BAE2B19'
DC X'21001BC0A191149898FD0E458AF04B03BCB78B81'
DC X'AA09D7DF7A8D36057717B5B91325317EEBC57C58'
DC X'24907255A4A19B1B7B5C548F71E1AF8ABEA25F16'
DC X'2A15ED7E1765366CC1471989A85FF279C386B6AF'
DC X'676BB92CEC733FF464B820B41246575A1C3B8F92'
DC X'2664F9FCEBEEEEB8EEBB1EB98F19FF0377C4D95B'
DC X'6A9F0315AAC64EF5851BF551B14F231455EFA5A5'
DC X'BB3C9E0FA6577413DCBE888F8B5A582E462ECE8E'
DC X'80581AE301340BDFA83EDD70A7F12CDCF558C906'
DC X'E8CE7C79EE5A5B42D9A576A1D42FBCBE550D7154'
DC X'ACCB93EFE8033952D843569B4756FE7432C2FCD1'
DC X'4FA079397E4663C78E528A0A7730F501B225D794'
DC X'73087B4962CDACAB210BFC09E818AF3E7313AB3B'
DC X'C4A366CCB715C0DADBDA7EFC1B47457B6FE22CA8'
DC X'4BC19C78F601354C40243E7FD71FF201B7A00FFA'
DC X'DEB4ADD0D00F5027D755FC30247D1DEBA1A40B56'
DC X'B588FB7DF6C738BE06ECAE3BEF00A528D9D872B5'
DC X'392B5B8C4CB1B30D7DB6CE3B5CE0EF3E70409D13'
DC X'8F920529949BB7B3C33FAF8C4C2685D4923D2110'
DC X'1DF05FB4F9ED9FC661D4BF88621F45BACBF2A427'
DC X'A89AD84D4973C304EE14290F47846A546D0E25C9'
DC X'F5C2998519861 0F7E8E86028EDC7DBFDAEA528AE'
DC X'C7B7DFF875D518006EFB7FA185E0F61327DCDC19'
DC X'438E1A31C914FBF911D26EE7887D7DAAB40A614F'
DC X'E401B4F6CBD5192881B7BD8C5FDF8903803EBA32'
DC X'6C303FE4AC568D139C074C274CF2968F6C9A367C'
DC X'4F65C539C27D32FBB3996B30EEAA8E14F8E5A130'
DC X'7B6CE78FB8F5F76BE3EA19026656B3664610F21D'
DC X'2C07EC227C7AE6099AABBF5C3FFF9023143B8B72'
DC X'7C4655E4CBDB584B7289F8B35AC9498ABF6F2FE7'
DC X'F5FD74146436604F1519DE1CC5A0F9C4672B60C0'
DC X'0D17C1022201B3CA954D5686653CFE2227ED987B'
DC X'475C9FF07CB28D45A959E5BD9B21881D8DD326F9'
DC X'17A1441F713B2D45B526756AD8B709E91E310AF5'
DC X'FF913F0E8A6F99DD908F88067CA91436EBB9E9DB'
```

```
DC X'68718FE293CE1180755922C82ACE486D9FD062D'
DC X'5D5DB7D65F446C6B042C0AF2726DE74ED3D621A9'
DC X'C42B4E388584D15A261EBBFB7EED049CEBA06273'
DC X'F53660522153183 6AE0938528DC9C05151976AE1'
DC X'728B82FE66851664113BC77D1691541E05661699'
DC X'84761E390E9B0A84059FF82E576F854A6FFD6557'
DC X'5391B63FF473F07B40671E30699728C0872FC26A'
DC X'51286778A2758043DE94C8FA5077ABF6FC5F7A6C'
DC X'21964A3D84706A2FFA95230873BA7C5A79B98F13'
DC X'64AAF6082CF648C2A6E3492DC2580D92425E8CAC'
DC X'DA9D15E77B545E1AAE5B415E6C1FF54BC47E09FC'
DC X'9A910B5E6EF54C5F10A460242D4864FF94843EF8'
DC X'19EC53A0B8EDAEF9468A21AD2EF82F87D7BCDA20'
DC X'C2A45BAAE45890C094BDAF4261A62F95C91FAC8C'
DC X'23C1F0B7CC44A30516FFC8ACF2755B4C467215DD'
DC X'ABB67B454E0EA38444A2BA07E5DEACE7E815E868'
DC X'BCA1963A99B75C09F12B2B6D55AF9736541140B2'
DC X'6B9E12792AAC4D49F9996B30E3E2C138B791F51A'
DC X'B589D18432080FD5BB32BC19FA4C97BDE24282A5'
DC X'6D2BF9DEB3C3188EAEC85CC2735EA2813B56CF89'
DC X'4B97080B641D764B7CA7E193BAC1F99CAF76DA45'
DC X'B4E782454FC48AAC714ECE62B9600950B347A74D'
DC X'CFD539D6437AED94B575ED4578EC7035838AAD53'
DC X'A22635CBBC3C8B88DEC3D3181BBE95FABD811C74'
DC X'217894F58B9122E49CBC8671A93523634FC261C9'
DC X'3E6EAB0155129D52014504B652D3176C59C17238'
DC X'1B0573F468E476322AAA11204F90BD8EB2CBF89C'
DC X'838B6818A491EAB1B7A583B64988C285BAAEFC97'
DC X'524CEF3DC1CA1EEA7856269E1D01988ACA5ED5E9'
DC X'4DDEE0DDCCEC563561B0B879D6096B5134500059C'
DC X'CA824C375AB032E8C016F4E88B03904E67B13D5D'
DC X'8781C45A22156A14D1639E65FEFE5A422D642245'
DC X'85AE1239133DC219F2787F3ADC694120F6A0E032'
DC X'3740BBCF01D01087EC3D116BE835D2CC7195893F'
DC X'43C312390CE9C4EBF1616BB767B76318B917CB8E'
DC X'0A1749328CE0417BD7A73468DBDE7E9424B0C5F6'
DC X'A3EE89C248B32B67DA7937F11410BBD070A87585'
DC X'EC173BD44D449693B5DAB79 0034DE74E736156B7'
DC X'8E64776D28D3E519634CB2CDD4637446C4FB60B9'
DC X'61C163F3CF19A682DFAD6151EE7E8BAD5F079DE5'
DC X'0740037C778E2A057E6F15D0DE3E92BE90C8732F'
DC X'78CC035C0066E1AE50FB04BF04FCAC57CE4C8EFF'
DC X'43CAF237A3FCF4FF626BA5E1FC3AA69E4A801386'
DC X'E8E7AC5D3B7FCF946C4E2240F63847FB05E74370'
DC X'044A331A3FCEA93393125A1867CCE2CE706605D1'
DC X'C82634E9CF2C9130248CD882E5DDDC034CEF0BEC'
DC X'8E5BFA54F877F2BA617A9A9BFD282B85D88F9BA2'
DC X'0F6308FFBE19BB859AE0D8D5C3B9F3D73FA15DB2'
DC X'9B0DC75BF06C2A5AD389DA0F59EEF1FBDA5343B0'
DC X'FA054A1FF159F60F893E72A7D349A62600E7C53B'
DC X'EA815C49AAED75DE2B08D0F5FCA8E05220D8CCE7'
```

```
DC X'D5E9E432B34EDD606D7DE62E4FB5247CC439A25E'
DC X'75CAABC65ADC6D23F0B5749EAF7DBBC5BBEB9FEE'
DC X'0918584745B5712354A6F3EACAECB3D03629AF61'
DC X'8A071F1C55BFE1E63ABEAB5CABF484BA562EC9FC'
DC X'C70A5DAF1E13D693D70B21CF6909F438AEF82805'
DC X'474F828FDF66A540B1DFF6CD58B0F5FFB0E72EE8'
DC X'8E324BF7B20A3F68406015369091DC251F9F746A'
DC X'22F360E3436E68DEFEA5A5A00FC64200232EE5C6'
DC X'E02FFD9C20CA8D6EA564CFB2FE26BE2AE3C0827F'
DC X'67C23817E478A2DC857543FF38C2D2196662976A'
DC X'ADADAAB1CB567D49C6A70453F2C5996D45BC19D5'
DC X'27E3DBD70B27A016DADA37639AC28E3C6E8F6E51'
DC X'F891BC64C423F8A141F608A672BF423ADE141351'
DC X'6C9E376C1FA4DA1E897FDA967658E091E4280B0A'
DC X'4A8C59B931E730DAC217067648A7D7666D87CBC0'
DC X'6EF026193B97E4E32D9EC4957695821F31C3F7E8'
DC X'5803 53E2A83FF444E91558C42AC1DA063E48ADE62'
DC X'B412E780AE1BEB6405C29A07FFBF9C0977EF7E00'
DC X'486A5C5003353AD42F395D4F2661A06C26CEE447'
DC X'28710B00BAE4B014C8E11BF7CE07F04F31ADA9A2'
DC X'7F4EDB718D4FF6FC554E761549D09170772A46CA'
DC X'C164B75486FC71C2837B314D8330EDEC40905085'
DC X'F605DBD6292F305053E6C4F87E60BF90F0543153'
DC X'7EFA6A9FA21FAA141DD80A7389E769EAA29DA8E2'
DC X'C1241623DC0773DE8252D9750EC775C3EC19F271'
DC X'0AF18E74B7FF11F1DF70AE26F7BA6884B9F32C2E'
DC X'1966E645D915BBAB8260F4CE7C286C5618FFB705'
DC X'45AB5ED7EB9E5C7EE15E8FF36BFA096B836C7BBC'
DC X'DDFC39BBA8CAEEA441757BAA6D04BD244BFB3E9E'
DC X'9E228005EBBAC35247FBC8396B791BC81C638D56'
DC X'FD55050 5647F369B9767A321642EF378372DFA5D'
DC X'79F0E5A4DE7A20CA20DF7E1F08A90541D84FA52B'
DC X'2E17F1BF1A220248981EF40CEB0FE573E4F887DC'
DC X'F1DF99E19BF08DFA556D9BBB04AC5699328B319D'
DC X'A5691847AFF36328A1A9EF22F2672D68E72739B9'
DC X'1A4424A26B1495F99A586E6F4600D1499F3C97F1B'
DC X'BCD6C21317DBAECE73016ECB1D1BC16CE7658EF2'
DC X'829CB20D376DFF32FAE3C842CD82CDD5612E66B3'
DC X'A37DFE716E66BAF3F3B5546A6E32FF7A86C21949'
DC X'6C0746424DE00456DC5EE117484212338A9A4ABB'
DC X'6A3D9ADBA5D8887CEAADB3EBDF642FA1203CFA90'
DC X'ACBBF6CC29D59932C49B530C2E5C0DDC014B5BD1'
DC X'C3753CFBBE1BEBC463A34BDEA74CC951BCC438AD'
DC X'012F6083DC01F1478093F08B7AD7EF84FA9589CF'
DC X'BA5D3857824666B5A548CC5AEBECB3AB05056F93'
DC X'4A23F5D3449730FBB8AECBC153085FD254E13062'
DC X'925CC1BE41825C0B85849123DAD12D3403B6D739'
DC X'2B9FC314541C3A7944140800EC7C28A13AD4AA16'
DC X'447CBE1AA9F3ABC317532D54235C39E56F188C59'
DC X'3A21DD95414ACCFDC34C093794F49E0A8DFD0A99'
```

```
DC X'4DEA95D98FF5E4780811CC57E99D18AA58FAA13F'
DC X'71C5A1750 1DCDF4A40E9E6F4B36E743BFD780448'
DC X'282C2CBC5C920A387CF6EA7AF7522B393AFB1D0E'
DC X'5D717613FBA7639AEB228D29F5E6A4A3B1EDA08E'
DC X'464936FE0A2CC8A129BD6C0A7954B2E101195F99'
DC X'6177779B5AAF3807F53EA179F14AEE6112F9B00A'
DC X'58BAC46AFF78C961E516337A4D5E09AD8D950A34'
DC X'2AF0598DE4B7286AA7299659023CE824598CB0CB'
DC X'46BF65C3765181651A726A45C4A9AD73A8990C04'
DC X'C96CC692EBC7960 4E9B9E337B03D8C36432E2BF0'
DC X'A1C54F3B3A86494E9386E2951C99AAA6D660731F'
DC X'9423E8F904C72FB144179FC105D82E2065ED5750'
DC X'D92D6C3B7EAE164EBC7D4AD8A25D81AA70385E25'
DC X'DC5658544809EC4A88F91C3218F60651B668DF6D'
DC X'B06F6BA4DA3E7A180C5F25928DB8130DB03FF66A'
DC X'356DAC6C9EC6F3F43319B9BF821778EF4B321821'
DC X'AB76A53E25D532EE57E83F543671FF349F38B3D3'
DC X'6FB58DC1F9314FA12FA541785945570D18E00C5C'
DC X'4D20FA80E507597CDD38DD986BB224F412EB2CC0'
DC X'EB7EB2ADF740109D4F382B9707452D3B88617CE2'
DC X'9123B94445AF1C1119F58F11F25E1B2ED92215BE'
DC X'E3D355C7BCD7A48365955109AF9544874D24ECA6'
DC X'8C271FAB1A73C28A918D553EF00FBB6F0EE0F8EA'
DC X'C038A7CD42EAA57C9BE3376C6BB1AB3E724D8EDA'
DC X'AD3EE89E224AB5D57B53F055D90F5DFB45BAC0C7'
DC X'DB946A99FCC0E480233EFC38316 0D813BAF82AE5'
DC X'3F7DD4D0618FCED183DAC0D197A6DE2B6449914A'
DC X'E110A6EC7FAEAF76A096E204AB5924F72EC7C571'
DC X'B0866E2BFA67434D02C292ACD035B78BE9A397EC'
DC X'FD26A7865BA3A511E753EA7E62C51A2BCF7A9EDE'
DC X'474CA89B82891FAE63DF1FC915B297FDC4BBF66F'
DC X'A045C6BE823E1CCBA336BA8F3488019FF26545AE'
DC X'501354E4236964840CDAB490971D062F9673132E'
DC X'88EBA13F0B3E9AA93DC7443C76E2AB4D310F8AE7'
DC X'E2F8E94976A4391443C0047635E9D5D9D06CF14B'
DC X'709A8AE1DB3B2FB0AD082E0B0ED54E58BA74DE35'
DC X'0D91F69D35C26E7F580B24274CBCB3EFA4826347'
DC X'42D3C09D1F1979643376A8AE4CFE061CFA0F59D8'
DC X'EAF5F588DF3A4C9C8F7290DC1278D789BD16E1EF'
DC X'0C4275EB018CFDC3ED8417CFBC761A07F971E896'
DC X'82A626CE259377F92A5A9E8A8812B2566E732874'
DC X'60C347C28B858E521A6A9FC867EF5B86CD2CBC78'
DC X'836933F94C29C753476EAC68405207C50920950E'
DC X'E9310F8F87DAB5486FBD12008ADEBC0B778AFE3A'
DC X'2CDB364A8573AB1403D69DAF09AAA4B1BE595D9E'
DC X'507A1675F19A0208B7B39EF213337474C59AE3EE'
DC X'7CB5E80B26B84C1DF3B3CBAD7BF93C0652D914E3'
DC X'8ABD4091A06641AA9606E2FF805141E0FCF7CD04'
DC X'DC706ED89620F0D8AF3322ABE1A1B7B8597F56EE'
DC X'5982BA50BB60513447843 56B299F6F0631EA44A0'
```

```
DC X'1FA4E551D75D132520491B5664E1FAC618F2236A'
DC X'CF7A26F094B53B8C9B4BD135671B7D1EB8E4CE33'
DC X'A5C8DF06ED369FDB1937090413A6A1CFE3274B85'
DC X'A74E8166EBD7BD5A11AAB3D20D46C19EFEF99F86'
DC X'2E87271C1293EE9587F57BD051A902AFA8B8F4C0'
DC X'1528A76F3944B0DC3670FCFB53BF36A0386E1687'
DC X'CF15E918108545 0B6BE33F13ADECA0307B6DFD71'
DC X'979243D4827641AC7FEE44CEC5204C4B84DB7911'
DC X'7B0CB55C10F6259168E9814A98DD6721E2B241F5'
DC X'2289A4FE9C5D184876ACC5D81C8BA353524CA2E2'
DC X'CC053897E15F18FECE361EAFA37BB6E97822A02F'
DC X'8B1623EDEDEA30ECA7FFD343B0778EA0B5E6CDE8'
DC X'0405 25D291941661300 4A6D61727D0EDF01A3E53'
DC X'3DC2CDD5ED1949F5A611C535CCDF860A5FFB2337'
DC X'1BC472939E394BFB3E051FB9543EA9322475FD93'
DC X'0D34CE84DD4D7E23945653642D42C020C7352D9E'
DC X'E52BAFA4C5C82F2BF0391898B82B2A0C0E9FFAC8'
DC X'57F5F3109 7C124E2CB34F801EB6DC9AEB584EDA4'
DC X'8D5CD4FDA96FD027B46A35411A550FCE00DBDFD4'
DC X'9BDD28A63F9801 12A9A9FA76F8A7D6759349CF95'
DC X'A834F49AC4A818B4FF09C7EDBD21A977A8C8AA64'
DC X'E0BACC46419DE135A4153306440 51337DF6457D4'
DC X'5E83898680A73AD16B08AAD785A2323C5C41627B'
DC X'C6C5057 9B9DE7C79EE5A5B42D9A576A1D42FBCBE'
DC X'550D7154ACCB93EFE803327DF28C0423F445E2A4'
DC X'834510F53056B4FB7BA538ABC7227C6869C86C99'
DC X'0C86661C829FC67394FF023E52121E65721EB761'
DC X'7F701CA59758A1AE8B853E3339A922EB985D8794'
DC X'05B7A3EE6B1165C5EDFADD9A8716C3C462E39111'
DC X'7922 08BCCF83541E91A84724F2BCE1191729CCF5'
DC X'E8F9C0C2B881EB354FA56C1153175AAF0F9E190C'
DC X'9AC384BE90A53A4097D8950899B8FD760DF455A2'
DC X'59E90439BC4D4D95B135E41EE4D338B3B128163D'
DC X'9F5CCC08B190DBC892FEBF897D12693C68E2691F'
DC X'69C16FDE21FCA0AA91BEAE21FD8575725560E21C'
DC X'26F6FDB1BD080279E057417B66D8F34710DF6DC1'
DC X'66A153872A6BEB99B5D714265AEEDC35882829F1'
DC X'2EC4DE460B56DD6A5C456859BA38FD9FA7F4CD90'
DC X'E1A1D4B34221BBB7C6EBB51EBE895AB5FBE46038'
DC X'1A66F929D7BFC4540F9775109621C2FA4FC40FEF'
DC X'AE3E24BC34E6103DC84BBE28053112727FE2BDCA'
DC X'5B2DCCD4B2AC962875F41EAB4E6A18B08D460406'
DC X'509E41404C13B9C8CE06320D78BEC43BD72C855B'
DC X'2C8C2B7ED0782AF8903C4D23E6A497FB56F183A3'
DC X'BD96F2FE7F5FD74146436604F1519DE1CC5A0F9C'
DC X'4672B60C00D1733962A7ADBA64B4B4510009E4F5'
DC X'7E6E8B35946B0FECFB2EF03B007A24C449C0472D'
DC X'A5A41B52B825BAB112EBE5CE7570AA9FAA8CABE4'
DC X'20A130BAD76AF58E5EFAA291DC2D8117FC04F96B'
DC X'A29C5DDE928EF4ABD9849E5540EBC4DE373FECB0'
```

```
DC X'9FA3501B2CF1DF4624831443F0A82F0F156532EA'
DC X'9C509AED491D63C1D49AA5DF2A0448B83A1EDA80'
DC X'91B08C8349F1F0FD18B700683003EEF6C1826B93'
DC X'D08113020FD1C3FD80D39D295683ACC8E683AF20'
DC X'0B849773B76EDE9AE62C1A9C071358DBA668B406'
DC X'BC54BC6808BC0E14FC4679FC17C981E76BA85693'
DC X'62887EBBAABFD8694E83A1D5250BA26EDCD6E084'
DC X'48BF35031624F75D4058E17DD3F29F5A0B9E4BA0'
DC X'E11E70CA5F374387F0A717D86D104797C00E5C59'
DC X'CE8993BA31D6A5C8AC6414CB76FDFFF70D076029'
DC X'23A5A47800814C53ED855A8B8270EBD1BB7E41B0'
DC X'B68883D5EF9A08C0777D28B93897CFED1582F263'
DC X'FE07986C99057B41B8FCE99FF2932A7676A36EA5'
DC X'33A9564E6D5E9E63D3BA660777D33E931D98DDFF'
DC X'A9C0B5688E5346A94B2BF223E60150A85C05DEAC'
DC X'E7E815E868BCA1963A99B75C09F12B2B6D55AF97'
DC X'3654155B6F9E009AD2D8C931C8230517A46165F8'
DC X'E6AEC8AA60234CFEB1895550001437162EF1DF060'
DC X'880556F4B9A11271EA745755A074BF1447B34A3F'
DC X'24DAC4BE3594210D1F1AFBC0883AAA6AEE22FB80'
DC X'D42F2305527D15A2ED3B3C1E14078EC1183634CF'
DC X'513D1A1117EEE52F1CF175B53864D56497798991'
DC X'410E5CBCCC6DE03B194745E6976EEE885DDD3E1E'
DC X'B7752477E8929010F40086BD12BAAB22C1FAB48B'
DC X'6EC87EB713EB2E304ABE8A3352487855EE0623CE'
DC X'81D834016F2228A2F7079843B51DE8924E0D7F24'
DC X'819D048E992B8F2FD3E263141FC718F266547675'
DC X'16DF45CC5D6D20A4E403DDF7BFD76D25468C9B22'
DC X'EAFBAA1A5A4109B89541BE3EC3A2A93256DEB0E5'
```

Philip Rigby (UK)

The article *Transferring code from the Web to a mainframe*, published on page 57 of this issue of *RACF Update*, highlights a modification to overcome problems experienced when downloading *Update* code to a mainframe. A detailed discussion of many of the problems associated with sharing mainframe code can be found on the Xephon Web site in the article *Sharing mainframe code* (http://www.xephon.com/contnote.html/).

A started task RACF trick

The following is a potentially useful trick with started tasks that may be of interest to both RACF people and to systems programmers.

Firstly some background. At the site where I work we attempted to rationalize the set-up of RACF across a multi-company environment. Unfortunately, owing to an error in the changes made, JES2 on one of the systems had its user-id deleted. This was not noticed until some time later when that system was re-IPLed, at which point JES2 was unable to start. Ideally the solution to this would be to use the RACF subsystem to allow console commands to be input to define the user-id. This assumes that the subsystem has been defined, that console commands are available, and indeed that a user has access to console commands anyway! Not only that, it also assumes that the person called out knows the commands.

However, there is another possible way out which might be of use as a temporary measure. If there is another started task around with a similar access level to JES2, then you can try the following procedure:

Modify the JCL for JES2 (you will need an LPAR for this) and then issue the following command:

```
S taskname,JOBNAME=JES2
```

This will start the task with ‘taskname’ authority, yet provide a subsystem link to JES2. Hence the system starts. This idea can be exploited for other tasks, such as TSO, etc, and means that systems programmers have an escape for such errors without requiring emergency RACF access.

Admittedly I’ve only ever had to use this method twice in my career, but even so it could be a worthwhile addition to your box of system tricks.

C A Jacques
Systems Programmer (UK)

A replacement for the RACF Report Writer

BACKGROUND

As you are probably aware, the RACF Report Writer is stabilized at RACF 1.9.2. This means that no new record types will be supported, eg OpenEdition audit records, etc.

So when you move to OS/390 and the OS/390 Security Server, as RACF is called nowadays, you should start looking for a replacement – especially if you plan to use OpenEdition. However, as long as you do not use these parts of OS/390, the RACF Report Writer will keep working properly.

IBM is not supplying any other reporting utility to replace the Report Writer, instead giving you the SMF Data Unload Utility (IRRADU00). This utility unloads SMF type 30, 80, 81, and 83 records in a tabular format and leaves the reporting to you. However, in SYS1.SAMPLIB, members IRRADULD, IRRADUTB, and IRRADUQU are examples, eg DB2 load statements and SQL to extract some relevant reports. If you feel that loading parts of your SMF data into DB2 is OK, you could easily use these samples. However, I feel that loading thousands and thousands of SMF records into DB2 every day is not really what I want – so I have written a reporting tool in SAS. Using SAS you can have SQL-like capabilities without having to keep your SMF data in databases. It is then up to you whether you want to keep your records permanently or temporarily.

The record lay-out is documented in *OS/390: Security Server (RACF) Macros and Interfaces,* SC28-1916-01, and, furthermore, my tool has been updated with APAR OW25202.

INSTALLATION

The tool is made up of a series of macros to make it easier to use. It requires that you execute the load job which loads all the macros into a MACLIB of your choice. This MACLIB then has to be placed or concatenated to your SAS procedure at DD-name SOURCLIB. There

is one macro per recordtype, plus the EXTRACT-macro which is used to control which record types you want to collect.

The unload job reads the relevant SMF records using IRRADBU00 and unloads them for use by the subsequent reporting step.

The report job reads the unloaded SMF data and produces the reports you want using SAS. I have enclosed a few sample reports that produces the same kind of reports usually produced using the RACF Report Writer.

IRRADUOO

```
//jobname   JOB (acct-info),'SMF data unload',
//         CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
//SMFDUMP  EXEC PGM=IFASMFDP
//SYSPRINT DD  SYSOUT=*
//ADUPRINT DD  SYSOUT=*
//SMFIN    DD  DISP=SHR,DSN=SYS1.MAN1,AMP=('BUFND=20')
//SMFOUT   DD  DUMMY
//OUTDD    DD  DSN=<your.unloaded.smfdata>,
//             DISP=MOD
//*OUTDD    DD  DSN=<your.unloaded.smfdata>,
//*             LRECL=5096,RECFM=VB,DISP=(NEW,CATLG,DELETE),
//*             SPACE=(CYL,(50,10),RLSE)
//SYSIN    DD  *
     INDD(SMFIN,OPTIONS(DUMP))
     OUTDD(SMFOUT,TYPE(000:255))
     ABEND(NORETRY)
     USER2(IRRADU00)
     USER3(IRRADU86)
/*
```

SASADUOO

```
//jobname   JOB (acct-info),'IRRABU00 extract',
//         CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID,REGION=8800K
//*  This job reads SMF data extracted from IRRADU00
//*  and formats in a SAS database
//S1       EXEC SAS
//WORK     DD UNIT=WORK,SPACE=(CYL,(20,10))
//SORTWK01 DD SPACE=(CYL,(&SORT)),UNIT=SYSDA
//SORTWK02 DD SPACE=(CYL,(&SORT)),UNIT=SYSDA
//SORTWK03 DD SPACE=(CYL,(&SORT)),UNIT=SYSDA
//SMFIN    DD DSN=<your.unloaded.smfdata>,DISP=SHR
//*RACF     DD DSN=<your.permanent.output>,DISP=(NEW,CATLG,DELETE),
//*            SPACE=(27648,(1500),,,ROUND),
```

```
//*           DCB=(RECFM=FS,LRECL=27648,BLKSIZE=27648,DSORG=PS)
//RACF     DD DSN=&IRRADU,
//            SPACE=(27648,(15ØØ),,,ROUND),
//            DCB=(RECFM=FS,LRECL=27648,BLKSIZE=27648,DSORG=PS)
//SYSIN    DD *
OPTIONS COMPRESS=NO;

  /*                                          */
  /* EXTRACT THE RECORD TYPES RELEVANT FOR YOU */
  /*                                          */

%EXTRACT(
      JOBINIT
      ACCESS
      ADDVOL
      RENAMEDS
      DELRES
      DELVOL
      DEFINE
      ADDSD
      ADDGROUP
      ADDUSER
      ALTDSD
      ALTGROUP
      ALTUSER
      CONNECT
      DELDSD
      DELGROUP
      DELUSER
      PASSWORD
      PERMIT
      RALTER
      RDEFINE
      RDELETE
      REMOVE
      SETROPTS
      RVARY
      APPCLU
      GENERAL
      DIRSRCH
      DACCESS
      FACCESS
      CHAUDIT
      CHDIR
      CHMOD
      CHOWN
      CLRSETID
      EXESETID
      GETPSENT
      INITOEDP
      TERMOEDP
```

```
      KILL
      LINK
      MKDIR
      MKNOD
      MNTFSYS
      OPENFILE
      PTRACE
      RENAMEF
      RMDIR
      SETEGID
      SETEUID
      SETGID
      SETUID
      SYMLINK
      UNLINK
      UMNTFSYS
      CHKFOWN
      CHKPRIV
      OPENSTTY
      RACLINK
      IPCCHK
      IPCGET
      IPCCTL
      SETGROUP
      CKOWN2
      RAUDIT
      RACFINIT
      CLASNAME
      DSAF
      )

  /* Find all of the job initiation records that      */
  /* describe job initiations which have a qualifer   */
  /* other than a normal initiation or termination.   */
PROC PRINT DATA=RACF.JOBINIT LABEL UNIFORM;
  WHERE EVENQUAL NOT IN ('SUCCESS','TERM','RACINITI','RACINITD')
        ;
  TITLE 'Unusual job initiations';

  /* Find all of the resource accesses that           */
  /* were audited because the user associated with    */
  /* the access had an extraordinary authority, such  */
  /* as SPECIAL, OPERATIONS, or AUDITOR, or           */
  /* or granted by an installation exit.              */
PROC PRINT DATA=RACF.ACCESS  LABEL UNIFORM;
   WHERE
       AUTHSPEC = 'YES' |
       AUTHOPER = 'YES' |
       AUTHAUDT = 'YES' |
       AUTHEXIT = 'YES'
       ;
```

```
  TITLE 'Access granted because of RACF priviliges or exit';

  /* Find all of the resource accesses that           */
  /* have a qualifier other than success.             */
PROC PRINT DATA=RACF.ACCESS  LABEL UNIFORM;
   WHERE
       EVENQUAL ¬= 'SUCCESS '
       ;
  TITLE 'Unsuccessful access attempts';

  /* Find all of the ADDVOL attempts that             */
  /* have a qualifier other than success.             */
PROC PRINT DATA=RACF.ADDVOL   LABEL UNIFORM;
   WHERE
       EVENQUAL ¬= 'SUCCESS '
       ;
  TITLE 'Unsuccessful ADDVOL attempts';

  /* Find all of the rename attempts that             */
  /* have a qualifier other than success.             */
PROC PRINT DATA=RACF.RENAMEDS LABEL UNIFORM;
   WHERE
       EVENQUAL ¬= 'SUCCESS '
       ;
  TITLE 'Unsuccessful rename attempts';

  /* Find all of the delete attempts that             */
  /* have a qualifier other than success.             */
PROC PRINT DATA=RACF.DELRES LABEL UNIFORM;
   WHERE
       EVENQUAL ¬= 'SUCCESS '
       ;
  TITLE 'Unsuccessful delete attempts';

  /* Find all of the delete volume attempts           */
PROC PRINT DATA=RACF.DELVOL   LABEL UNIFORM;
  TITLE 'Successful delete attempts';

  /* Find all of the define attempts that             */
  /* have a qualifier other than success.             */
PROC PRINT DATA=RACF.DEFINE LABEL UNIFORM;
   WHERE
       EVENQUAL ¬= 'SUCCESS '
       ;
  TITLE 'Unsuccessful define attempts';

PROC PRINT DATA=RACF.ADDSD    LABEL UNIFORM;
  TITLE 'All ADDSD commands';

PROC PRINT DATA=RACF.ADDGROUP LABEL UNIFORM;
```

```
  TITLE 'All ADDGROUP commands';

PROC PRINT DATA=RACF.ADDUSER  LABEL UNIFORM;
  TITLE 'All ADDUSER commands';

PROC PRINT DATA=RACF.ALTDSD   LABEL UNIFORM;
  TITLE 'All ALTDSD commands';

PROC PRINT DATA=RACF.ALTGROUP LABEL UNIFORM;
  TITLE 'All ALTGROUP commands';

PROC PRINT DATA=RACF.ALTUSER  LABEL UNIFORM;
  TITLE 'All ALTUSER commands';

PROC PRINT DATA=RACF.CONNECT  LABEL UNIFORM;
  TITLE 'All CONNECT commands';

PROC PRINT DATA=RACF.DELDSD   LABEL UNIFORM;
  TITLE 'All DELDSD commands';

PROC PRINT DATA=RACF.DELGROUP LABEL UNIFORM;
  TITLE 'All DELGROUP commands';

PROC PRINT DATA=RACF.DELUSER  LABEL UNIFORM;
  TITLE 'All DELUSER commands';

PROC PRINT DATA=RACF.PASSWORD LABEL UNIFORM;
  TITLE 'All PASSWORD commands';

PROC PRINT DATA=RACF.PERMIT   LABEL UNIFORM;
  TITLE 'All PERMIT commands';

PROC PRINT DATA=RACF.RALTER   LABEL UNIFORM;
  TITLE 'All RALTER commands';

PROC PRINT DATA=RACF.RDEFINE  LABEL UNIFORM;
  TITLE 'All RDEFINE commands';

PROC PRINT DATA=RACF.RDELETE  LABEL UNIFORM;
  TITLE 'All RDELETE commands';

PROC PRINT DATA=RACF.REMOVE   LABEL UNIFORM;
  TITLE 'All REMOVE commands';

PROC PRINT DATA=RACF.SETROPTS LABEL UNIFORM;
  TITLE 'All SETROPTS commands';

PROC PRINT DATA=RACF.RVARY    LABEL UNIFORM;
  TITLE 'All RVARY commands';
```

```
PROC PRINT DATA=RACF.APPCLU   LABEL UNIFORM;
  TITLE ' ';

PROC PRINT DATA=RACF.GENERAL  LABEL UNIFORM;
PROC PRINT DATA=RACF.DIRSRCH  LABEL UNIFORM;
PROC PRINT DATA=RACF.DACCESS  LABEL UNIFORM;
PROC PRINT DATA=RACF.FACCESS  LABEL UNIFORM;
PROC PRINT DATA=RACF.CHAUDIT  LABEL UNIFORM;
PROC PRINT DATA=RACF.CHDIR    LABEL UNIFORM;
PROC PRINT DATA=RACF.CHMOD    LABEL UNIFORM;
PROC PRINT DATA=RACF.CHOWN    LABEL UNIFORM;
PROC PRINT DATA=RACF.CLRSETID LABEL UNIFORM;
PROC PRINT DATA=RACF.EXESETID LABEL UNIFORM;
PROC PRINT DATA=RACF.GETPSENT LABEL UNIFORM;
PROC PRINT DATA=RACF.INITOEDP LABEL UNIFORM;
PROC PRINT DATA=RACF.TERMOEDP LABEL UNIFORM;
PROC PRINT DATA=RACF.KILL     LABEL UNIFORM;
PROC PRINT DATA=RACF.LINK     LABEL UNIFORM;
PROC PRINT DATA=RACF.MKDIR    LABEL UNIFORM;
PROC PRINT DATA=RACF.MKNOD    LABEL UNIFORM;
PROC PRINT DATA=RACF.MNTFSYS  LABEL UNIFORM;
PROC PRINT DATA=RACF.OPENFILE LABEL UNIFORM;
PROC PRINT DATA=RACF.PTRACE   LABEL UNIFORM;
PROC PRINT DATA=RACF.RENAMEF  LABEL UNIFORM;
PROC PRINT DATA=RACF.RMDIR    LABEL UNIFORM;
PROC PRINT DATA=RACF.SETEGID  LABEL UNIFORM;
PROC PRINT DATA=RACF.SETEUID  LABEL UNIFORM;
PROC PRINT DATA=RACF.SETGID   LABEL UNIFORM;
PROC PRINT DATA=RACF.SETUID   LABEL UNIFORM;
PROC PRINT DATA=RACF.SYMLINK  LABEL UNIFORM;
PROC PRINT DATA=RACF.UNLINK   LABEL UNIFORM;
PROC PRINT DATA=RACF.UMNTFSYS LABEL UNIFORM;
PROC PRINT DATA=RACF.CHKFOWN  LABEL UNIFORM;
PROC PRINT DATA=RACF.CHKPRIV  LABEL UNIFORM;
PROC PRINT DATA=RACF.OPENSTTY LABEL UNIFORM;
PROC PRINT DATA=RACF.RACLINK  LABEL UNIFORM;
PROC PRINT DATA=RACF.IPCCHK   LABEL UNIFORM;
PROC PRINT DATA=RACF.IPCGET   LABEL UNIFORM;
PROC PRINT DATA=RACF.IPCCTL   LABEL UNIFORM;
PROC PRINT DATA=RACF.SETGROUP LABEL UNIFORM;
PROC PRINT DATA=RACF.CKOWN2   LABEL UNIFORM;
PROC PRINT DATA=RACF.RAUDIT   LABEL UNIFORM;

  /* Print RACF options in effect                    */
PROC PRINT DATA=RACF.RACFINIT LABEL UNIFORM;
  TITLE 'RACF options at IPL';

  /* Print classes active at IPL                     */
PROC PRINT DATA=RACF.CLASNAME LABEL UNIFORM NOOBS;
  VAR RICSMFID RICDATE RICTIME RICCLASN RICSTATS RICAUDIT RICACTIV
      RICGNRIC RICGNCMD RICGLBAL RICRCLST RICGNLST RICLOGOP
```

```
      ;
  WHERE RICACTIV = 'YES'
        ;
  TITLE 'Classes active at IPL';

PROC PRINT DATA=RACF.DSAF     LABEL UNIFORM;
  TITLE 'Datasets affected by SECLABEL change';

RUN;
/*
```

SASMACS

```
//jobname   JOB ,'Create SAS maclib',
//         CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
//*   THIS JOB CREATES A SAS MACRO LIBRARY        */
//CREATE   EXEC PGM=IEBUPDTE,PARM='NEW'
//SYSUT2   DD DSN=<your.sas.maclib>,DISP=(NEW,CATLG,DELETE),
//            SPACE=(CYL,(3,1,7)),LRECL=8Ø,RECFM=FB,BLKSIZE=2792Ø
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
./        ADD   LIST=ALL,NAME=EXTRACT
%MACRO EXTRACT(PARM) / PARMBUFF;
  %LET ENDLINE = %LENGTH(&SYSPBUFF);
  %IF &ENDLINE = Ø %THEN
    %DO;
      %PUT Error - no parameters specified;
      %GOTO EXIT;
    %END;
  %ELSE
    %DO;
DATA
      %DO I = 1 %TO &ENDLINE;
        %LET SUBTYPE = %SCAN(&SYSPBUFF,&I);
        %IF &SUBTYPE = %THEN
          %LET I = &ENDLINE;
        %ELSE
          %DO;
            %&SUBTYPE(REQ=DEFINE)
          %END;
      %END;
 ;
INFILE SMFIN;
INPUT EVENTYPE $   1-8    /* READ AND KEEP RECORD FOR FURTHER INPUT */
               @ ;      /* READ AND KEEP RECORD FOR FURTHER INPUT */
FORMAT TIMEWRIT TIME8.
       DATEWRIT YYMMDD1Ø.
       READTIME TIME8.
       READDATE YYMMDD1Ø.
         ;
```

```
LABEL EVENTYPE = 'Event type'
      EVENQUAL = 'Event qualifier'
      TIMEWRIT = 'Time written'
      DATEWRIT = 'Date written'
      SMFID    = 'SMF id'
      VIOLAT   = 'Violation?'
      UNDFND   = 'Undefined user?'
      WARNING  = 'Warning?'
      USERID   = 'Userid'
      GRPID    = 'Groupid'
      AUTHNORM = 'Normal authorization?'
      AUTHSPEC = 'Special authorization?'
      AUTHOPER = 'Operatns authorization?'
      AUTHAUDT = 'Audit authorization?'
      AUTHEXIT = 'Exit authorized?'
      AUTHFSFT = 'Failsoft check?'
      AUTHBYPA = 'User BYPASS?'
      AUTHTRST = 'User trusted?'
      LOGCLASS = 'Class audit?'
      LOGUSER  = 'UAUDIT?'
      LOGSPEC  = 'Special audit?'
      LOGACCES = 'Profile audit?'
      LOGRACIN = 'RACINIT fail?'
      LOGALWYS = 'Always audit?'
      LOGCMDVI = 'CMDVIOL audit?'
      LOGGLOBL = 'Global audit?'
      TERMLVL  = 'Terminal level'
      BKOTFAIL = 'Fail in backout?'
      PROFSAME = 'Profile unchanged?'
      TERMID   = 'Terminal'
      JOBNAME  = 'Jobname'
      READTIME = 'Time entered'
      READDATE = 'Date entered'
      SMFUSER  = 'SMF exit userid'
      LOGLEVEL = 'SECLEVEL audit?'
      LOGVMEVN = 'VMEVENT audit?'
      LOGLGOPT = 'LOGOPTIONS audit?'
      LOGSECL  = 'SECLABELAUDT?'
      LOGCMPAT = 'COMPATMODE audit?'
      LOGAPLAU = 'APPLAUDIT?'
      LOGNOMVS = 'Missing OMVS segment?'
      LOGOMVPR = 'OE superuser required?'
      AUTHOMSU = 'OE superuser?'
      AUTHOMSY = 'Requestor is OE'
      USERSECL = 'SECLABEL'
      RACFVERS = 'RACF version'
      ;
SELECT(EVENTYPE); /* SELECT WHICH EVENTS TO EXTRACT */
      %DO I = 1 %TO &ENDLINE;
        %LET SUBTYPE = %SCAN(&SYSPBUFF,&I);
```

```
        %IF &SUBTYPE = %THEN
          %LET I = &ENDLINE;
        %ELSE
          %DO;
            %&SUBTYPE(REQ=EXTRACT)
          %END;
      %END;
  OTHERWISE
    DELETE;
END;
    %END;
%EXIT:
%MEND EXTRACT;
./        ADD   LIST=ALL,NAME=SMFHDR
%MACRO SMFHDR;
   EVENTYPE
%MEND SMFHDR;
./        ADD   LIST=ALL,NAME=SMF8ØHDR
%MACRO SMF8ØHDR(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
      EVENQUAL
      TIMEWRIT
      DATEWRIT
      SMFID
      VIOLAT
      UNDFND
      WARNING
      USERID
      GRPID
      AUTHNORM
      AUTHSPEC
      AUTHOPER
      AUTHAUDT
      AUTHEXIT
      AUTHFSFT
      AUTHBYPA
      AUTHTRST
      LOGCLASS
      LOGUSER
      LOGSPEC
      LOGACCES
      LOGRACIN
      LOGALWYS
      LOGCMDVI
      LOGGLOBL
      TERMLVL
      BKOTFAIL
      PROFSAME
      TERMID
```

```
    JOBNAME
    READTIME
    READDATE
    SMFUSER
    LOGLEVEL
    LOGVMEVN
    LOGLGOPT
    LOGSECL
    LOGCMPAT
    LOGAPLAU
    LOGNOMVS
    LOGOMVPR
    AUTHOMSU
    AUTHOMSY
    USERSECL
    RACFVERS
  %END;
%IF &REQ = EXTRACT %THEN
  %DO;
    EVENQUAL $  1Ø-17
             @19
    TIMEWRIT TIME8.
             @28
    DATEWRIT YYMMDD1Ø.
    SMFID    $  39-42
    VIOLAT   $  44-47
    UNDFND   $  49-52
    WARNING  $  54-57
    USERID   $  59-66
    GRPID    $  68-75
    AUTHNORM $  77-8Ø
    AUTHSPEC $  82-85
    AUTHOPER $  87-9Ø
    AUTHAUDT $  92-95
    AUTHEXIT $  97-1ØØ
    AUTHFSFT $ 1Ø2-1Ø5
    AUTHBYPA $ 1Ø7-11Ø
    AUTHTRST $ 112-115
    LOGCLASS $ 117-12Ø
    LOGUSER  $ 122-125
    LOGSPEC  $ 127-13Ø
    LOGACCES $ 132-135
    LOGRACIN $ 137-14Ø
    LOGALWYS $ 142-145
    LOGCMDVI $ 147-15Ø
    LOGGLOBL $ 152-155
    TERMLVL    157-159
    BKOTFAIL $ 161-164
    PROFSAME $ 166-169
    TERMID   $ 171-178
    JOBNAME  $ 18Ø-187
```

```
                @189
       READTIME TIME8.
                @198
       READDATE YYMMDD1Ø.
       SMFUSER  $ 2Ø9-216
       LOGLEVEL $ 218-221
       LOGVMEVN $ 223-226
       LOGLGOPT $ 228-231
       LOGSECL  $ 233-236
       LOGCMPAT $ 238-241
       LOGAPLAU $ 243-246
       LOGNOMVS $ 248-251
       LOGOMVPR $ 253-256
       AUTHOMSU $ 258-261
       AUTHOMSY $ 263-266
       USERSECL $ 268-275
       RACFVERS $ 277-28Ø
     %END;
%MEND SMF8ØHDR;
./        ADD   LIST=ALL,NAME=JOBINIT
%MACRO JOBINIT(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
      %PUT Including variables from JOBINIT extension;
  RACF.JOBINIT  (KEEP=%SMFHDR
                      %SMF8ØHDR(REQ=DEFINE)
                      INIAPPL
                      INILOGST
                      INIJOBNM
                      INIUSER
                      INIUTKNE
                      INIUPRE
                      INIVFYX
                      ININJEUS
                      INIUAUDT
                      INISPEC
                      INIDEFLT
                      INIUNDEF
                      INIERROR
                      INITRUST
                      INISESTP
                      INISURRO
                      INIREMOT
                      INIPRVLD
                      INISECL
                      INIEXNOD
                      INISUSER
                      INISNODE
                      INISGRP
                      INISPOE
```

```
                      INISPCLS
                      INITKUSR
                      INITKGRP
                      INITKDFT
                      INITKSEC
                      INIAPPCL
                  )
    %END;
  %IF &REQ = EXTRACT %THEN
    %DO;
      %PUT Including datadefinition for JOBINIT extension;
      WHEN('JOBINIT') DO;
        INPUT %SMF80HDR(REQ=EXTRACT)
              INIAPPL  $       282-289
              INILOGST $       291-390
              INIJOBNM $       547-554
              INIUSER  $       556-575
              INIUTKNE $       577-580
              INIUPRE  $       582-585
              INIVFYX  $       587-590
              ININJEUS $       592-595
              INIUAUDT $       597-600
              INISPEC  $       602-605
              INIDEFLT $       607-610
              INIUNDEF $       612-615
              INIERROR $       617-620
              INITRUST $       622-625
              INISESTP $       627-634
              INISURRO $       636-639
              INIREMOT $       641-644
              INIPRVLD $       646-649
              INISECL  $       651-658
              INIEXNOD $       660-667
              INISUSER $       669-676
              INISNODE $       678-685
              INISGRP  $       687-694
              INISPOE  $       696-703
              INISPCLS $       705-712
              INITKUSR $       714-721
              INITKGRP $       723-730
              INITKDFT $       732-735
              INITKSEC $       737-740
              INIAPPCL $       742-757
              ;
LABEL INIAPPL  = 'APPLid'
      INILOGST = 'Logstr'
      INIJOBNM = 'Jobname'
      INIUSER  = 'User name'
      INIUTKNE = 'Utoken encr.?'
      INIUPRE  = 'Pre-1.9?'
      INIVFYX  = 'VERIFYX propagation?'
```

```
      ININJEUS = 'Undefined NJE user?'
      INIUAUDT = 'UAUDIT?'
      INISPEC  = 'RACF special?'
      INIDEFLT = 'Default token?'
      INIUNDEF = 'Undefined user?'
      INIERROR = 'Token in error?'
      INITRUST = 'User trusted?'
      INISESTP = 'Session type'
      INISURRO = 'Surrogate user?'
      INIREMOT = 'Remote job?'
      INIPRVLD = 'Priviliged user?'
      INISECL  = 'User SECLABEL'
      INIEXNOD = 'Execution node'
      INISUSER = 'Submitting user'
      INISNODE = 'Submitting node'
      INISGRP  = 'Submitting group'
      INISPOE  = 'Port of entry'
      INISPCLS = 'Class of POE'
      INITKUSR = 'Userid'
      INITKGRP = 'Groupid'
      INITKDFT = 'Default group?'
      INITKSEC = 'Default SECLABEL?'
      INIAPPCL = 'APPC key link'
      ;
        OUTPUT RACF.JOBINIT;
      END;
    %END;
%MEND JOBINIT;
./        ADD   LIST=ALL,NAME=ACCESS
%MACRO ACCESS(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
    %PUT Including variables from ACCESS extension;
     RACF.ACCESS   (KEEP=%SMFHDR
                         %SMF8ØHDR(REQ=DEFINE)
                         ACCRESNM
                         ACCREQST
                         ACCGRANT
                         ACCLEVEL
                         ACCVOL
                         ACCOLDVL
                         ACCCLASS
                         ACCAPPL
                         ACCTYPE
                         ACCNAME
                         ACCOWNID
                         ACCLOGST
                         ACCRECVR
                         ACCUSERN
                         ACCSECL
```

```
ACCUTKNE
ACCUPRE
ACCUVFYX
ACCUNJEU
ACCUUAUD
ACCUSPEC
ACCUDFLT
ACCUUNDF
ACCUERR
ACCUTRST
ACCUSEST
ACCUSURO
ACCURMT
ACCUPRVL
ACCUSECL
ACCUEXND
ACCUSUSR
ACCUSNOD
ACCUSGRP
ACCUSPOE
ACCUSPCL
ACCUTUSR
ACCUTGRP
ACCUTDFT
ACCUTSEC
ACCRTKNE
ACCRPRE
ACCRVFYX
ACCRNJEU
ACCRUAUD
ACCRSPEC
ACCRDFLT
ACCRUNDF
ACCRERR
ACCRTRST
ACCRSEST
ACCRSURO
ACCRRMT
ACCRPRVL
ACCRSECL
ACCREXND
ACCRSUSR
ACCRSNOD
ACCRSGRP
ACCRSPOE
ACCRSPCL
ACCRTUSR
ACCRTGRP
ACCRTDFT
ACCRTSEC
```

```
                        ACCAPPC
                        ACCDCELK
                        ACCAUTHT
                    )
  %END;
%IF &REQ = EXTRACT %THEN
  %DO;
  %PUT Including datadefinition for ACCESS extension;
 WHEN('ACCESS') DO;
     INPUT %SMF80HDR(REQ=EXTRACT)
           ACCRESNM $       282-481
           ACCREQST $       538-545
           ACCGRANT $       547-554
           ACCLEVEL         556-558
           ACCVOL   $       560-565
           ACCOLDVL $       567-572
           ACCCLASS $       574-581
           ACCAPPL  $       583-590
           ACCTYPE  $       592-599
           ACCNAME  $       601-800
           ACCOWNID $       848-855
           ACCLOGST $       857-1056
           ACCRECVR $      1113-1120
           ACCUSERN $      1122-1141
           ACCSECL  $      1143-1150
           ACCUTKNE $      1152-1155
           ACCUPRE  $      1157-1160
           ACCUVFYX $      1162-1165
           ACCUNJEU $      1167-1170
           ACCUUAUD $      1172-1175
           ACCUSPEC $      1177-1180
           ACCUDFLT $      1182-1185
           ACCUUNDF $      1187-1190
           ACCUERR  $      1192-1195
           ACCUTRST $      1197-1200
           ACCUSEST $      1202-1209
           ACCUSURO $      1211-1214
           ACCURMT  $      1216-1219
           ACCUPRVL $      1221-1224
           ACCUSECL $      1226-1233
           ACCUEXND $      1235-1242
           ACCUSUSR $      1244-1251
           ACCUSNOD $      1253-1260
           ACCUSGRP $      1262-1269
           ACCUSPOE $      1271-1278
           ACCUSPCL $      1280-1287
           ACCUTUSR $      1289-1296
           ACCUTGRP $      1298-1305
           ACCUTDFT $      1307-1310
           ACCUTSEC $      1312-1315
```

```
              ACCRTKNE $      1317-1320
              ACCRPRE  $      1322-1325
              ACCRVFYX $      1327-1330
              ACCRNJEU $      1332-1335
              ACCRUAUD $      1337-1340
              ACCRSPEC $      1342-1345
              ACCRDFLT $      1347-1350
              ACCRUNDF $      1352-1355
              ACCRERR  $      1357-1360
              ACCRTRST $      1362-1365
              ACCRSEST $      1367-1374
              ACCRSURO $      1376-1379
              ACCRRMT  $      1381-1384
              ACCRPRVL $      1386-1389
              ACCRSECL $      1391-1398
              ACCREXND $      1400-1407
              ACCRSUSR $      1409-1416
              ACCRSNOD $      1418-1425
              ACCRSGRP $      1427-1434
              ACCRSPOE $      1436-1443
              ACCRSPCL $      1445-1452
              ACCRTUSR $      1454-1461
              ACCRTGRP $      1463-1470
              ACCRTDFT $      1472-1475
              ACCRTSEC $      1477-1480
              ACCAPPC  $      1482-1497
              ACCDCELK $      1499-1514
              ACCAUTHT $      1516-1528
              ;
LABEL ACCRESNM = 'Resource name'
      ACCREQST = 'Access requested'
      ACCGRANT = 'Access granted'
      ACCLEVEL = 'Resource level'
      ACCVOL   = 'Resource volume'
      ACCOLDVL = 'Resource OLDVOL'
      ACCCLASS = 'Class name'
      ACCAPPL  = 'APPLid'
      ACCTYPE  = 'Type of resource'
      ACCNAME  = 'Profile name'
      ACCOWNID = 'Profile owner'
      ACCLOGST = 'Logstr'
      ACCRECVR = 'Receiver'
      ACCUSERN = 'User name'
      ACCSECL  = 'Resource SECLABEL'
      ACCUTKNE = 'Utkn encrypted?'
      ACCUPRE  = 'Utkn Pre-1.9?'
      ACCUVFYX = 'Utkn VERIFYX propagation?'
      ACCUNJEU = 'Utkn undefined NJE user?'
      ACCUUAUD = 'Utkn UAUDIT?'
      ACCUSPEC = 'Utkn RACF special?'
```

```
 ACCUDFLT = 'Utkn default token?'
 ACCUUNDF = 'Utkn undefined user?'
 ACCUERR  = 'Utkn token in error?'
 ACCUTRST = 'Utkn user trusted?'
 ACCUSEST = 'Utkn session type'
 ACCUSURO = 'Utkn surrogate user?'
 ACCURMT  = 'Utkn remote job?'
 ACCUPRVL = 'Utkn priviliged user?'
 ACCUSECL = 'Utkn user SECLABEL'
 ACCUEXND = 'Utkn execution node'
 ACCUSUSR = 'Utkn submitting user'
 ACCUSNOD = 'Utkn submitting node'
 ACCUSGRP = 'Utkn submitting group'
 ACCUSPOE = 'Utkn port of entry'
 ACCUSPCL = 'Utkn class of POE'
 ACCUTUSR = 'Utkn userid'
 ACCUTGRP = 'Utkn groupid'
 ACCUTDFT = 'Utkn default group?'
 ACCUTSEC = 'Utkn default SECLABEL?'
 ACCRTKNE = 'Rtkn encrypted?'
 ACCRPRE  = 'Rtkn Pre-1.9?'
 ACCRVFYX = 'Rtkn VERIFYX propagation?'
 ACCRNJEU = 'Rtkn undefined NJE user?'
 ACCRUAUD = 'Rtkn UAUDIT?'
 ACCRSPEC = 'Rtkn RACF special?'
 ACCRDFLT = 'Rtkn default token?'
 ACCRUNDF = 'Rtkn undefined user?'
 ACCRERR  = 'Rtkn token in error?'
 ACCRTRST = 'Rtkn user trusted?'
 ACCRSEST = 'Rtkn session type'
 ACCRSURO = 'Rtkn surrogate user?'
 ACCRRMT  = 'Rtkn remote job?'
 ACCRPRVL = 'Rtkn priviliged user?'
 ACCRSECL = 'Rtkn user SECLABEL'
 ACCREXND = 'Rtkn execution node'
 ACCRSUSR = 'Rtkn submitting user'
 ACCRSNOD = 'Rtkn submitting node'
 ACCRSGRP = 'Rtkn submitting group'
 ACCRSPOE = 'Rtkn port of entry'
 ACCRSPCL = 'Rtkn class of POE'
 ACCRTUSR = 'Rtkn userid'
 ACCRTGRP = 'Rtkn groupid'
 ACCRTDFT = 'Rtkn default group?'
 ACCRTSEC = 'Rtkn default SECLABEL?'
 ACCAPPC  = 'APPC key link'
 ACCDCELK = 'DCE link'
 ACCAUTHT = 'Request type'
 ;
  OUTPUT RACF.ACCESS;
END;
```

```
    %END;
%MEND ACCESS;
./        ADD   LIST=ALL,NAME=ADDVOL
%MACRO ADDVOL(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
    %PUT Including variables from ADDVOL extension;
     RACF.ADDVOL   (KEEP=%SMFHDR
                         %SMF8ØHDR(REQ=DEFINE)
                         ADVRESNM
                         ADVGRANT
                         ADVLEVEL
                         ADVVOL
                         ADVOLDVL
                         ADVCLASS
                         ADVOWNID
                         ADVLOGST
                         ADVUSERN
                         ADVUTKNE
                         ADVUPRE
                         ADVUVFYX
                         ADVUNJEU
                         ADVUUAUD
                         ADVUSPEC
                         ADVUDFLT
                         ADVUUNDF
                         ADVUERR
                         ADVUTRST
                         ADVUSEST
                         ADVUSURO
                         ADVURMT
                         ADVUPRVL
                         ADVUSECL
                         ADVUEXND
                         ADVUSUSR
                         ADVUSNOD
                         ADVUSGRP
                         ADVUSPOE
                         ADVUSPCL
                         ADVUTUSR
                         ADVUTGRP
                         ADVUTDFT
                         ADVUTSEC
                         ADVAPPC
                         ADVSPEFD
                    )
    %END;
  %IF &REQ = EXTRACT %THEN
    %DO;
```

```
    %PUT Including datadefinition for ADDVOL extension;
   WHEN('ADDVOL') DO;
       INPUT %SMF80HDR(REQ=EXTRACT)
             ADVRESNM $       282-481
             ADVGRANT $       538-545
             ADVLEVEL         547-549
             ADVVOL   $       551-556
             ADVOLDVL $       558-563
             ADVCLASS $       565-572
             ADVOWNID $       574-581
             ADVLOGST $       583-782
             ADVUSERN $       839-858
             ADVUTKNE $       860-863
             ADVUPRE  $       865-868
             ADVUVFYX $       870-873
             ADVUNJEU $       875-878
             ADVUUAUD $       880-883
             ADVUSPEC $       885-888
             ADVUDFLT $       890-893
             ADVUUNDF $       895-898
             ADVUERR  $       900-903
             ADVUTRST $       905-908
             ADVUSEST $       910-917
             ADVUSURO $       919-922
             ADVURMT  $       924-927
             ADVUPRVL $       929-932
             ADVUSECL $       934-941
             ADVUEXND $       943-950
             ADVUSUSR $       952-959
             ADVUSNOD $       961-968
             ADVUSGRP $       970-977
             ADVUSPOE $       979-986
             ADVUSPCL $       988-995
             ADVUTUSR $       997-1004
             ADVUTGRP $      1006-1013
             ADVUTDFT $      1015-1018
             ADVUTSEC $      1020-1023
             ADVAPPC  $      1025-1040
             ADVSPEFD $      1042-1241
               ;
LABEL ADVRESNM = 'Resource name'
      ADVGRANT = 'Access granted'
      ADVLEVEL = 'Resource level'
      ADVVOL   = 'Resource volume'
      ADVOLDVL = 'Resource OLDVOL'
      ADVCLASS = 'Class name'
      ADVOWNID = 'Profile owner'
      ADVLOGST = 'Logstr'
      ADVUSERN = 'User name'
      ADVUTKNE = 'Utoken encr.?'
```

```
      ADVUPRE  = 'Pre-1.9?'
      ADVUVFYX = 'VERIFYX propagation?'
      ADVUNJEU = 'Undefined NJE user?'
      ADVUUAUD = 'UAUDIT?'
      ADVUSPEC = 'RACF special?'
      ADVUDFLT = 'Default token?'
      ADVUUNDF = 'Undefined user?'
      ADVUERR  = 'Token in error?'
      ADVUTRST = 'User trusted?'
      ADVUSEST = 'Session type'
      ADVUSURO = 'Surrogate user?'
      ADVURMT  = 'Remote job?'
      ADVUPRVL = 'Priviliged user?'
      ADVUSECL = 'User SECLABEL'
      ADVUEXND = 'Execution node'
      ADVUSUSR = 'Submitting user'
      ADVUSNOD = 'Submitting node'
      ADVUSGRP = 'Submitting group'
      ADVUSPOE = 'Port of entry'
      ADVUSPCL = 'Class of POE'
      ADVUTUSR = 'Userid'
      ADVUTGRP = 'Groupid'
      ADVUTDFT = 'Default group?'
      ADVUTSEC = 'Default SECLABEL?'
      ADVAPPC  = 'APPC key link'
      ADVSPEFD = 'RRSF keywords'
      ;
       OUTPUT RACF.ADDVOL;
     END;
    %END;
%MEND ADDVOL;
./        ADD   LIST=ALL,NAME=RENAMEDS
%MACRO RENAMEDS(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
    %PUT Including variables from RENAMEDS extension;
     RACF.RENAMEDS (KEEP=%SMFHDR
                         %SMF8ØHDR(REQ=DEFINE)
                         RENRESNM
                         RENNEWRS
                         RENLEVEL
                         RENVOL
                         RENCLASS
                         RENOWNID
                         RENLOGST
                         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
                        RENSPEFD
                    )
  %END;
%IF &REQ = EXTRACT %THEN
  %DO;
  %PUT Including datadefinition for RENAMEDS extension;
 WHEN('RENAMEDS') DO;
     INPUT %SMF80HDR(REQ=EXTRACT)
           RENRESNM $       282-481
           RENNEWRS $       538-737
           RENLEVEL         794-796
           RENVOL   $       798-803
           RENCLASS $       805-812
           RENOWNID $       814-821
           RENLOGST $       823-1022
           RENUSERN $      1079-1098
           RENUTKNE $      1100-1103
           RENUPRE  $      1105-1108
           RENUVFYX $      1110-1113
           RENUNJEU $      1115-1118
           RENUUAUD $      1120-1123
           RENUSPEC $      1125-1128
           RENUDFLT $      1130-1133
           RENUUNDF $      1135-1138
           RENUERR  $      1140-1143
           RENUTRST $      1145-1148
```

```
              RENUSEST $      1150-1157
              RENUSURO $      1159-1162
              RENURMT  $      1164-1167
              RENUPRVL $      1169-1172
              RENUSECL $      1174-1181
              RENUEXND $      1183-1190
              RENUSUSR $      1192-1199
              RENUSNOD $      1201-1208
              RENUSGRP $      1210-1217
              RENUSPOE $      1219-1226
              RENUSPCL $      1228-1235
              RENUTUSR $      1237-1244
              RENUTGRP $      1246-1253
              RENUTDFT $      1255-1258
              RENUTSEC $      1260-1263
              RENAPPC  $      1265-1280
              RENSPEFD $      1282-1481
                ;
LABEL RENRESNM = 'Old resource name'
      RENNEWRS = 'New resource name'
      RENLEVEL = 'Resource level'
      RENVOL   = 'Resource volume'
      RENCLASS = 'Class name'
      RENOWNID = 'Profile owner'
      RENLOGST = 'Logstr'
      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 = 'Priviliged 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'
       RENSPEFD = 'RRSF keywords'
       ;
        OUTPUT RACF.RENAMEDS;
      END;
     %END;
%MEND RENAMEDS;
./        ADD   LIST=ALL,NAME=DELRES
%MACRO DELRES(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
    %PUT Including variables from DELRES extension;
     RACF.DELRES   (KEEP=%SMFHDR
                         %SMF8ØHDR(REQ=DEFINE)
                         DLRRESNM
                         DLRLEVEL
                         DLRVOL
                         DLRCLASS
                         DLROWNID
                         DLRLOGST
                         DLRUSERN
                         DLRUTKNE
                         DLRUPRE
                         DLRUVFYX
                         DLRUNJEU
                         DLRUUAUD
                         DLRUSPEC
                         DLRUDFLT
                         DLRUUNDF
                         DLRUERR
                         DLRUTRST
                         DLRUSEST
                         DLRUSURO
                         DLRURMT
                         DLRUPRVL
                         DLRUSECL
                         DLRUEXND
                         DLRUSUSR
                         DLRUSNOD
                         DLRUSGRP
                         DLRUSPOE
                         DLRUSPCL
                         DLRUTUSR
                         DLRUTGRP
                         DLRUTDFT
                         DLRUTSEC
                         DLRAPPC
```

```
                        DLRSPEFD
                    )
    %END;
  %IF &REQ = EXTRACT %THEN
    %DO;
    %PUT Including datadefinition for DELRES extension;
   WHEN('DELRES') DO;
       INPUT %SMF80HDR(REQ=EXTRACT)
             DLRRESNM $       282-481
             DLRLEVEL         538-540
             DLRVOL   $       542-547
             DLRCLASS $       549-556
             DLROWNID $       558-565
             DLRLOGST $       567-766
             DLRUSERN $       823-842
             DLRUTKNE $       844-847
             DLRUPRE  $       849-852
             DLRUVFYX $       854-857
             DLRUNJEU $       859-862
             DLRUUAUD $       864-867
             DLRUSPEC $       869-872
             DLRUDFLT $       874-877
             DLRUUNDF $       879-882
             DLRUERR  $       884-887
             DLRUTRST $       889-892
             DLRUSEST $       894-901
             DLRUSURO $       903-906
             DLRURMT  $       908-911
             DLRUPRVL $       913-916
             DLRUSECL $       918-925
             DLRUEXND $       927-934
             DLRUSUSR $       936-943
             DLRUSNOD $       945-952
             DLRUSGRP $       954-961
             DLRUSPOE $       963-970
             DLRUSPCL $       972-979
             DLRUTUSR $       981-988
             DLRUTGRP $       990-997
             DLRUTDFT $       999-1002
             DLRUTSEC $      1004-1007
             DLRAPPC  $      1009-1024
             DLRSPEFD $      1026-1225
               ;
LABEL DLRRESNM = 'Resource name'
      DLRLEVEL = 'Resource level'
      DLRVOL   = 'Resource volume'
      DLRCLASS = 'Class name'
      DLROWNID = 'Profile owner'
      DLRLOGST = 'Logstr'
```

```
       DLRUSERN = 'User name'
       DLRUTKNE = 'Utoken encr.?'
       DLRUPRE  = 'Pre-1.9?'
       DLRUVFYX = 'VERIFYX propagation?'
       DLRUNJEU = 'Undefined NJE user?'
       DLRUUAUD = 'UAUDIT?'
       DLRUSPEC = 'RACF special?'
       DLRUDFLT = 'Default token?'
       DLRUUNDF = 'Undefined user?'
       DLRUERR  = 'Token in error?'
       DLRUTRST = 'User trusted?'
       DLRUSEST = 'Session type'
       DLRUSURO = 'Surrogate user?'
       DLRURMT  = 'Remote job?'
       DLRUPRVL = 'Priviliged user?'
       DLRUSECL = 'User SECLABEL'
       DLRUEXND = 'Execution node'
       DLRUSUSR = 'Submitting user'
       DLRUSNOD = 'Submitting node'
       DLRUSGRP = 'Submitting group'
       DLRUSPOE = 'Port of entry'
       DLRUSPCL = 'Class of POE'
       DLRUTUSR = 'Userid'
       DLRUTGRP = 'Groupid'
       DLRUTDFT = 'Default group?'
       DLRUTSEC = 'Default SECLABEL?'
       DLRAPPC  = 'APPC key link'
       DLRSPEFD = 'RRSF keywords'
       ;
        OUTPUT RACF.DELRES;
      END;
     %END;
%MEND DELRES;
./        ADD   LIST=ALL,NAME=DELVOL
%MACRO DELVOL(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
    %PUT Including variables from DELVOL extension;
     RACF.DELVOL   (KEEP=%SMFHDR
                         %SMF8ØHDR(REQ=DEFINE)
                         DLVRESNM
                         DLVLEVEL
                         DLVVOL
                         DLVCLASS
                         DLVOWNID
                         DLVLOGST
                         DLVUSERN
                         DLVUTKNE
                         DLVUPRE
```

```
                        DLVUVFYX
                        DLVUNJEU
                        DLVUUAUD
                        DLVUSPEC
                        DLVUDFLT
                        DLVUUNDF
                        DLVUERR
                        DLVUTRST
                        DLVUSEST
                        DLVUSURO
                        DLVURMT
                        DLVUPRVL
                        DLVUSECL
                        DLVUEXND
                        DLVUSUSR
                        DLVUSNOD
                        DLVUSGRP
                        DLVUSPOE
                        DLVUSPCL
                        DLVUTUSR
                        DLVUTGRP
                        DLVUTDFT
                        DLVUTSEC
                        DLVAPPC
                        DLVSPEFD
                    )
  %END;
%IF &REQ = EXTRACT %THEN
  %DO;
  %PUT Including datadefinition for DELVOL extension;
 WHEN('DELVOL') DO;
     INPUT %SMF8ØHDR(REQ=EXTRACT)
           DLVRESNM $       282-481
           DLVLEVEL         538-54Ø
           DLVVOL   $       542-547
           DLVCLASS $       549-556
           DLVOWNID $       558-565
           DLVLOGST $       567-766
           DLVUSERN $       823-842
           DLVUTKNE $       844-847
           DLVUPRE  $       849-852
           DLVUVFYX $       854-857
           DLVUNJEU $       859-862
           DLVUUAUD $       864-867
           DLVUSPEC $       869-872
           DLVUDFLT $       874-877
           DLVUUNDF $       879-882
           DLVUERR  $       884-887
           DLVUTRST $       889-892
```

```
              DLVUSEST $       894-901
              DLVUSURO $       903-906
              DLVURMT  $       908-911
              DLVUPRVL $       913-916
              DLVUSECL $       918-925
              DLVUEXND $       927-934
              DLVUSUSR $       936-943
              DLVUSNOD $       945-952
              DLVUSGRP $       954-961
              DLVUSPOE $       963-970
              DLVUSPCL $       972-979
              DLVUTUSR $       981-988
              DLVUTGRP $       990-997
              DLVUTDFT $       999-1002
              DLVUTSEC $      1004-1007
              DLVAPPC  $      1009-1024
              DLVSPEFD $      1026-1225
                ;
LABEL DLVRESNM = 'Resource name'
      DLVLEVEL = 'Resource level'
      DLVVOL   = 'Resource volume'
      DLVCLASS = 'Class name'
      DLVOWNID = 'Profile owner'
      DLVLOGST = 'Logstr'
      DLVUSERN = 'User name'
      DLVUTKNE = 'Utoken encr.?'
      DLVUPRE  = 'Pre-1.9?'
      DLVUVFYX = 'VERIFYX propagation?'
      DLVUNJEU = 'Undefined NJE user?'
      DLVUUAUD = 'UAUDIT?'
      DLVUSPEC = 'RACF special?'
      DLVUDFLT = 'Default token?'
      DLVUUNDF = 'Undefined user?'
      DLVUERR  = 'Token in error?'
      DLVUTRST = 'User trusted?'
      DLVUSEST = 'Session type'
      DLVUSURO = 'Surrogate user?'
      DLVURMT  = 'Remote job?'
      DLVUPRVL = 'Priviliged user?'
      DLVUSECL = 'User SECLABEL'
      DLVUEXND = 'Execution node'
      DLVUSUSR = 'Submitting user'
      DLVUSNOD = 'Submitting node'
      DLVUSGRP = 'Submitting group'
      DLVUSPOE = 'Port of entry'
      DLVUSPCL = 'Class of POE'
      DLVUTUSR = 'Userid'
      DLVUTGRP = 'Groupid'
      DLVUTDFT = 'Default group?'
```

```
      DLVUTSEC = 'Default SECLABEL?'
      DLVAPPC  = 'APPC key link'
      DLVSPEFD = 'RRSF keywords'
      ;
       OUTPUT RACF.DELVOL;
     END;
    %END;
%MEND DELVOL;
./        ADD   LIST=ALL,NAME=DEFINE
%MACRO DEFINE(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
    %PUT Including variables from DEFINE extension;
     RACF.DEFINE   (KEEP=%SMFHDR
                         %SMF8ØHDR(REQ=DEFINE)
                         DEFRESNM
                         DEFLEVEL
                         DEFVOL
                         DEFCLASS
                         DEFMODEL
                         DEFMDVOL
                         DEFOWNID
                         DEFLOGST
                         DEFUSERN
                         DEFUTKNE
                         DEFUPRE
                         DEFUVFYX
                         DEFUNJEU
                         DEFUUAUD
                         DEFUSPEC
                         DEFUDFLT
                         DEFUUNDF
                         DEFUERR
                         DEFUTRST
                         DEFUSEST
                         DEFUSURO
                         DEFURMT
                         DEFUPRVL
                         DEFUSECL
                         DEFUEXND
                         DEFUSUSR
                         DEFUSNOD
                         DEFUSGRP
                         DEFUSPOE
                         DEFUSPCL
                         DEFUTUSR
                         DEFUTGRP
                         DEFUTDFT
                         DEFUTSEC
```

```
                        DEFAPPC
                        DEFSPEFD
                        )
    %END;
  %IF &REQ = EXTRACT %THEN
    %DO;
    %PUT Including datadefinition for DEFINE extension;
   WHEN('DEFINE') DO;
       INPUT %SMF80HDR(REQ=EXTRACT)
             DEFRESNM $       282-481
             DEFLEVEL         538-540
             DEFVOL   $       542-547
             DEFCLASS $       549-556
             DEFMODEL $       558-757
             DEFMDVOL $       814-819
             DEFOWNID $       821-828
             DEFLOGST $       830-1029
             DEFUSERN $      1086-1105
             DEFUTKNE $      1107-1110
             DEFUPRE  $      1112-1115
             DEFUVFYX $      1117-1120
             DEFUNJEU $      1122-1125
             DEFUUAUD $      1127-1130
             DEFUSPEC $      1132-1135
             DEFUDFLT $      1137-1140
             DEFUUNDF $      1142-1145
             DEFUERR  $      1147-1150
             DEFUTRST $      1152-1155
             DEFUSEST $      1157-1164
             DEFUSURO $      1166-1169
             DEFURMT  $      1171-1174
             DEFUPRVL $      1176-1179
             DEFUSECL $      1181-1188
             DEFUEXND $      1190-1197
             DEFUSUSR $      1199-1206
             DEFUSNOD $      1208-1215
             DEFUSGRP $      1217-1224
             DEFUSPOE $      1226-1233
             DEFUSPCL $      1235-1242
             DEFUTUSR $      1244-1251
             DEFUTGRP $      1253-1260
             DEFUTDFT $      1262-1265
             DEFUTSEC $      1267-1270
             DEFAPPC  $      1272-1287
             DEFSPEFD $      1289-1488
             ;
LABEL DEFRESNM = 'Resource name'
      DEFLEVEL = 'Resource level'
      DEFVOL   = 'Resource volume'
      DEFCLASS = 'Class name'
```

```
       DEFMODEL = 'Model profile'
       DEFMDVOL = 'Model volume'
       DEFOWNID = 'Profile owner'
       DEFLOGST = 'Logstr'
       DEFUSERN = 'User name'
       DEFUTKNE = 'Utoken encr.?'
       DEFUPRE  = 'Pre-1.9?'
       DEFUVFYX = 'VERIFYX propagation?'
       DEFUNJEU = 'Undefined NJE user?'
       DEFUUAUD = 'UAUDIT?'
       DEFUSPEC = 'RACF special?'
       DEFUDFLT = 'Default token?'
       DEFUUNDF = 'Undefined user?'
       DEFUERR  = 'Token in error?'
       DEFUTRST = 'User trusted?'
       DEFUSEST = 'Session type'
       DEFUSURO = 'Surrogate user?'
       DEFURMT  = 'Remote job?'
       DEFUPRVL = 'Priviliged user?'
       DEFUSECL = 'User SECLABEL'
       DEFUEXND = 'Execution node'
       DEFUSUSR = 'Submitting user'
       DEFUSNOD = 'Submitting node'
       DEFUSGRP = 'Submitting group'
       DEFUSPOE = 'Port of entry'
       DEFUSPCL = 'Class of POE'
       DEFUTUSR = 'Userid'
       DEFUTGRP = 'Groupid'
       DEFUTDFT = 'Default group?'
       DEFUTSEC = 'Default SECLABEL?'
       DEFAPPC  = 'APPC key link'
       DEFSPEFD = 'RRSF keywords'
       ;
        OUTPUT RACF.DEFINE;
      END;
     %END;
%MEND DEFINE;
./        ADD   LIST=ALL,NAME=ADDSD
%MACRO ADDSD(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
    %PUT Including variables from ADDSD extension;
     RACF.ADDSD    (KEEP=%SMFHDR
                         %SMF8ØHDR(REQ=DEFINE)
                         AD_OWNID
                         AD_USERN
                         AD_SECL
                         AD_UTKNE
                         AD_UPRE
```

```
                        AD_UVFYX
                        AD_UNJEU
                        AD_UUAUD
                        AD_USPEC
                        AD_UDFLT
                        AD_UUNDF
                        AD_UERR
                        AD_UTRST
                        AD_USEST
                        AD_USURO
                        AD_URMT
                        AD_UPRVL
                        AD_USECL
                        AD_UEXND
                        AD_USUSR
                        AD_USNOD
                        AD_USGRP
                        AD_USPOE
                        AD_USPCL
                        AD_UTUSR
                        AD_UTGRP
                        AD_UTDFT
                        AD_UTSEC
                        AD_APPC
                        AD_SECLL
                        AD_DSNAM
                        AD_SPEFD
                        AD_FAILD
                        )
  %END;
%IF &REQ = EXTRACT %THEN
  %DO;
  %PUT Including datadefinition for ADDSD extension;
 WHEN('ADDSD') DO;
     INPUT %SMF80HDR(REQ=EXTRACT)
           AD_OWNID $       282-289
           AD_USERN $       291-310
           AD_SECL  $       312-319
           AD_UTKNE $       321-324
           AD_UPRE  $       326-329
           AD_UVFYX $       331-334
           AD_UNJEU $       336-339
           AD_UUAUD $       341-344
           AD_USPEC $       346-349
           AD_UDFLT $       351-354
           AD_UUNDF $       356-359
           AD_UERR  $       361-364
           AD_UTRST $       366-369
           AD_USEST $       371-378
           AD_USURO $       380-383
```

```
              AD_URMT  $       385-388
              AD_UPRVL $       390-393
              AD_USECL $       395-402
              AD_UEXND $       404-411
              AD_USUSR $       413-420
              AD_USNOD $       422-429
              AD_USGRP $       431-438
              AD_USPOE $       440-447
              AD_USPCL $       449-456
              AD_UTUSR $       458-465
              AD_UTGRP $       467-474
              AD_UTDFT $       476-479
              AD_UTSEC $       481-484
              AD_APPC  $       486-501
              AD_SECLL $       503-518
              AD_DSNAM $       520-563
              AD_SPEFD $       565-764
              AD_FAILD $      1590-1789
                ;
LABEL AD_OWNID = 'Profile owner'
      AD_USERN = 'User name'
      AD_SECL  = 'SECLABEL'
      AD_UTKNE = 'Utoken encr.?'
      AD_UPRE  = 'Pre-1.9?'
      AD_UVFYX = 'VERIFYX propagation?'
      AD_UNJEU = 'Undefined NJE user?'
      AD_UUAUD = 'UAUDIT?'
      AD_USPEC = 'RACF special?'
      AD_UDFLT = 'Default token?'
      AD_UUNDF = 'Undefined user?'
      AD_UERR  = 'Token in error?'
      AD_UTRST = 'User trusted?'
      AD_USEST = 'Session type'
      AD_USURO = 'Surrogate user?'
      AD_URMT  = 'Remote job?'
      AD_UPRVL = 'Priviliged user?'
      AD_USECL = 'User SECLABEL'
      AD_UEXND = 'Execution node'
      AD_USUSR = 'Submitting user'
      AD_USNOD = 'Submitting node'
      AD_USGRP = 'Submitting group'
      AD_USPOE = 'Port of entry'
      AD_USPCL = 'Class of POE'
      AD_UTUSR = 'Userid'
      AD_UTGRP = 'Groupid'
      AD_UTDFT = 'Default group?'
      AD_UTSEC = 'Default SECLABEL?'
      AD_APPC  = 'APPC key link'
      AD_SECLL = 'SECLABEL link'
      AD_DSNAM = 'Dataset name'
```

```
       AD_SPEFD = 'Keywords specified'
       AD_FAILD = 'Keywords failed'
       ;
        OUTPUT RACF.ADDSD;
      END;
     %END;
%MEND ADDSD;
./        ADD   LIST=ALL,NAME=ADDGROUP
%MACRO ADDGROUP(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
    %PUT Including variables from ADDGROUP extension;
     RACF.ADDGROUP (KEEP=%SMFHDR
                         %SMF8ØHDR(REQ=DEFINE)
                         AG_OWNID
                         AG_USERN
                         AG_UTKNE
                         AG_UPRE
                         AG_UVFYX
                         AG_UNJEU
                         AG_UUAUD
                         AG_USPEC
                         AG_UDFLT
                         AG_UUNDF
                         AG_UERR
                         AG_UTRST
                         AG_USEST
                         AG_USURO
                         AG_URMT
                         AG_UPRVL
                         AG_USECL
                         AG_UEXND
                         AG_USUSR
                         AG_USNOD
                         AG_USGRP
                         AG_USPOE
                         AG_USPCL
                         AG_UTUSR
                         AG_UTGRP
                         AG_UTDFT
                         AG_UTSEC
                         AG_APPC
                         AG_GRPID
                         AG_SPEFD
                         AG_FAILD
                         )
    %END;
  %IF &REQ = EXTRACT %THEN
    %DO;
```

```
    %PUT Including datadefinition for ADDGROUP extension;
   WHEN('ADDGROUP') DO;
       INPUT %SMF80HDR(REQ=EXTRACT)
             AG_OWNID $       282-289
             AG_USERN $       291-310
             AG_UTKNE $       312-315
             AG_UPRE  $       317-320
             AG_UVFYX $       322-325
             AG_UNJEU $       327-330
             AG_UUAUD $       332-335
             AG_USPEC $       337-340
             AG_UDFLT $       342-345
             AG_UUNDF $       347-350
             AG_UERR  $       352-355
             AG_UTRST $       357-360
             AG_USEST $       362-369
             AG_USURO $       371-374
             AG_URMT  $       376-379
             AG_UPRVL $       381-384
             AG_USECL $       386-393
             AG_UEXND $       395-402
             AG_USUSR $       404-411
             AG_USNOD $       413-420
             AG_USGRP $       422-429
             AG_USPOE $       431-438
             AG_USPCL $       440-447
             AG_UTUSR $       449-456
             AG_UTGRP $       458-465
             AG_UTDFT $       467-470
             AG_UTSEC $       472-475
             AG_APPC  $       477-492
             AG_GRPID $       494-501
             AG_SPEFD $       503-702
             AG_FAILD $      1528-1727
               ;
LABEL AG_OWNID = 'Profile owner'
      AG_USERN = 'User name'
      AG_UTKNE = 'Utoken encr.?'
      AG_UPRE  = 'Pre-1.9?'
      AG_UVFYX = 'VERIFYX propagation?'
      AG_UNJEU = 'Undefined NJE user?'
      AG_UUAUD = 'UAUDIT?'
      AG_USPEC = 'RACF special?'
      AG_UDFLT = 'Default token?'
      AG_UUNDF = 'Undefined user?'
      AG_UERR  = 'Token in error?'
      AG_UTRST = 'User trusted?'
      AG_USEST = 'Session type'
      AG_USURO = 'Surrogate user?'
      AG_URMT  = 'Remote job?'
```

```
       AG_UPRVL = 'Priviliged user?'
       AG_USECL = 'User SECLABEL'
       AG_UEXND = 'Execution node'
       AG_USUSR = 'Submitting user'
       AG_USNOD = 'Submitting node'
       AG_USGRP = 'Submitting group'
       AG_USPOE = 'Port of entry'
       AG_USPCL = 'Class of POE'
       AG_UTUSR = 'Userid'
       AG_UTGRP = 'Groupid'
       AG_UTDFT = 'Default group?'
       AG_UTSEC = 'Default SECLABEL?'
       AG_APPC  = 'APPC key link'
       AG_GRPID = 'Group name'
       AG_SPEFD = 'Keywords specified'
       AG_FAILD = 'Keywords failed'
       ;
        OUTPUT RACF.ADDGROUP;
     END;
    %END;
%MEND ADDGROUP;
./        ADD   LIST=ALL,NAME=ADDUSER
%MACRO ADDUSER(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
    %PUT Including variables from ADDUSER extension;
     RACF.ADDUSER (KEEP=%SMFHDR
                        %SMF8ØHDR(REQ=DEFINE)
                        AU_OWNID
                        AU_USERN
                        AU_UTKNE
                        AU_UPRE
                        AU_UVFYX
                        AU_UNJEU
                        AU_UUAUD
                        AU_USPEC
                        AU_UDFLT
                        AU_UUNDF
                        AU_UERR
                        AU_UTRST
                        AU_USEST
                        AU_USURO
                        AU_URMT
                        AU_UPRVL
                        AU_USECL
                        AU_UEXND
                        AU_USUSR
                        AU_USNOD
                        AU_USGRP
```

```
                        AU_USPOE
                        AU_USPCL
                        AU_UTUSR
                        AU_UTGRP
                        AU_UTDFT
                        AU_UTSEC
                        AU_APPC
                        AU_NOAGL
                        AU_NOAGR
                        AU_USRID
                        AG_SPEFD
                        AU_FAILD
                        AU_IGNRD
                        )
  %END;
%IF &REQ = EXTRACT %THEN
  %DO;
  %PUT Including datadefinition for ADDUSER extension;
 WHEN('ADDUSER') DO;
     INPUT %SMF80HDR(REQ=EXTRACT)
           AU_OWNID $       282-289
           AU_USERN $       291-310
           AU_UTKNE $       312-315
           AU_UPRE  $       317-320
           AU_UVFYX $       322-325
           AU_UNJEU $       327-330
           AU_UUAUD $       332-335
           AU_USPEC $       337-340
           AU_UDFLT $       342-345
           AU_UUNDF $       347-350
           AU_UERR  $       352-355
           AU_UTRST $       357-360
           AU_USEST $       362-369
           AU_USURO $       371-374
           AU_URMT  $       376-379
           AU_UPRVL $       381-384
           AU_USECL $       386-393
           AU_UEXND $       395-402
           AU_USUSR $       404-411
           AU_USNOD $       413-420
           AU_USGRP $       422-429
           AU_USPOE $       431-438
           AU_USPCL $       440-447
           AU_UTUSR $       449-456
           AU_UTGRP $       458-465
           AU_UTDFT $       467-470
           AU_UTSEC $       472-475
           AU_APPC  $       477-492
           AU_NOAGL $       494-497
           AU_NOAGR $       499-502
```

```
              AU_USRID $       5Ø4-511
              AU_SPEFD $       513-712
              AU_FAILD $      1538-1737
              AU_IGNRD $      2563-2762
                ;
LABEL AU_OWNID = 'Profile owner'
      AU_USERN = 'User name'
      AU_UTKNE = 'Utoken encr.?'
      AU_UPRE  = 'Pre-1.9?'
      AU_UVFYX = 'VERIFYX propagation?'
      AU_UNJEU = 'Undefined NJE user?'
      AU_UUAUD = 'UAUDIT?'
      AU_USPEC = 'RACF special?'
      AU_UDFLT = 'Default token?'
      AU_UUNDF = 'Undefined user?'
      AU_UERR  = 'Token in error?'
      AU_UTRST = 'User trusted?'
      AU_USEST = 'Session type'
      AU_USURO = 'Surrogate user?'
      AU_URMT  = 'Remote job?'
      AU_UPRVL = 'Priviliged user?'
      AU_USECL = 'User SECLABEL'
      AU_UEXND = 'Execution node'
      AU_USUSR = 'Submitting user'
      AU_USNOD = 'Submitting node'
      AU_USGRP = 'Submitting group'
      AU_USPOE = 'Port of entry'
      AU_USPCL = 'Class of POE'
      AU_UTUSR = 'Userid'
      AU_UTGRP = 'Groupid'
      AU_UTDFT = 'Default group?'
      AU_UTSEC = 'Default SECLABEL?'
      AU_APPC  = 'APPC key link'
      AU_NOAGL = 'Missing CLAUTH?'
      AU_NOAGR = 'Missing group auth?'
      AU_USRID = 'User id'
      AG_SPEFD = 'Keywords specified'
      AU_FAILD = 'Keywords failed'
      AU_IGNRD = 'Keywords ignored'
      ;
       OUTPUT RACF.ADDUSER;
     END;
    %END;
%MEND ADDUSER;
./        ADD   LIST=ALL,NAME=ALTDSD
%MACRO ALTDSD(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
    %PUT Including variables from ALTDSD extension;
```

```
   RACF.ALTDSD (KEEP=%SMFHDR
                     %SMF8ØHDR(REQ=DEFINE)
                     LD_OWNID
                     LD_USERN
                     LD_OLSCL
                     LD_UTKNE
                     LD_UPRE
                     LD_UVFYX
                     LD_UNJEU
                     LD_UUAUD
                     LD_USPEC
                     LD_UDFLT
                     LD_UUNDF
                     LD_UERR
                     LD_UTRST
                     LD_USEST
                     LD_USURO
                     LD_URMT
                     LD_UPRVL
                     LD_USECL
                     LD_UEXND
                     LD_USUSR
                     LD_USNOD
                     LD_USGRP
                     LD_USPOE
                     LD_USPCL
                     LD_UTUSR
                     LD_UTGRP
                     LD_UTDFT
                     LD_UTSEC
                     LD_APPC
                     LD_SECLL
                     LD_DSNAM
                     LD_SPEFD
                     LD_FAILD
                     LD_IGNRD
                     )
  %END;
%IF &REQ = EXTRACT %THEN
  %DO;
  %PUT Including datadefinition for ALTDSD extension;
 WHEN('ALTDSD') DO;
     INPUT %SMF8ØHDR(REQ=EXTRACT)
           LD_OWNID $       282-289
           LD_USERN $       291-31Ø
           LD_OLSCL $       312-319
           LD_UTKNE $       321-324
           LD_UPRE  $       326-329
           LD_UVFYX $       331-334
           LD_UNJEU $       336-339
```

```
                LD_UUAUD $       341-344
                LD_USPEC $       346-349
                LD_UDFLT $       351-354
                LD_UUNDF $       356-359
                LD_UERR  $       361-364
                LD_UTRST $       366-369
                LD_USEST $       371-378
                LD_USURO $       380-383
                LD_URMT  $       385-388
                LD_UPRVL $       390-393
                LD_USECL $       395-402
                LD_UEXND $       404-411
                LD_USUSR $       413-420
                LD_USNOD $       422-429
                LD_USGRP $       431-438
                LD_USPOE $       440-447
                LD_USPCL $       449-456
                LD_UTUSR $       458-465
                LD_UTGRP $       467-474
                LD_UTDFT $       476-479
                LD_UTSEC $       481-484
                LD_APPC  $       486-501
                LD_SECLL $       503-518
                LD_DSNAM $       520-563
                LD_SPEFD $       565-764
                LD_FAILD $      1590-1789
                LD_IGNRD $      2615-2814
                  ;
LABEL LD_OWNID = 'Profile owner'
      LD_USERN = 'User name'
      LD_OLSCL = 'SECLABEL deleted'
      LD_UTKNE = 'Utoken encr.?'
      LD_UPRE  = 'Pre-1.9?'
      LD_UVFYX = 'VERIFYX propagation?'
      LD_UNJEU = 'Undefined NJE user?'
      LD_UUAUD = 'UAUDIT?'
      LD_USPEC = 'RACF special?'
      LD_UDFLT = 'Default token?'
      LD_UUNDF = 'Undefined user?'
      LD_UERR  = 'Token in error?'
      LD_UTRST = 'User trusted?'
      LD_USEST = 'Session type'
      LD_USURO = 'Surrogate user?'
      LD_URMT  = 'Remote job?'
      LD_UPRVL = 'Priviliged user?'
      LD_USECL = 'User SECLABEL'
      LD_UEXND = 'Execution node'
      LD_USUSR = 'Submitting user'
      LD_USNOD = 'Submitting node'
      LD_USGRP = 'Submitting group'
```

```
       LD_USPOE = 'Port of entry'
       LD_USPCL = 'Class of POE'
       LD_UTUSR = 'Userid'
       LD_UTGRP = 'Groupid'
       LD_UTDFT = 'Default group?'
       LD_UTSEC = 'Default SECLABEL?'
       LD_APPC  = 'APPC key link'
       LD_SECLL = 'SECLABEL link'
       LD_DSNAM = 'Dataset name'
       LD_SPEFD = 'Keywords specified'
       LD_FAILD = 'Keywords failed'
       LD_IGNRD = 'Keywords ignored'
       ;
        OUTPUT RACF.ALTDSD;
      END;
     %END;
%MEND ALTDSD;
./        ADD   LIST=ALL,NAME=ALTGROUP
%MACRO ALTGROUP(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
    %PUT Including variables from ALTGROUP extension;
     RACF.ALTGROUP (KEEP=%SMFHDR
                         %SMF8ØHDR(REQ=DEFINE)
                         LG_OWNID
                         LG_USERN
                         LG_UTKNE
                         LG_UPRE
                         LG_UVFYX
                         LG_UNJEU
                         LG_UUAUD
                         LG_USPEC
                         LG_UDFLT
                         LG_UUNDF
                         LG_UERR
                         LG_UTRST
                         LG_USEST
                         LG_USURO
                         LG_URMT
                         LG_UPRVL
                         LG_USECL
                         LG_UEXND
                         LG_USUSR
                         LG_USNOD
                         LG_USGRP
                         LG_USPOE
                         LG_USPCL
                         LG_UTUSR
                         LG_UTGRP
```

```
                          LG_UTDFT
                          LG_UTSEC
                          LG_APPC
                          LG_GRPID
                          LG_SPEFD
                          LG_FAILD
                          LG_IGNRD
                          )
    %END;
  %IF &REQ = EXTRACT %THEN
    %DO;
    %PUT Including datadefinition for ALTGROUP extension;
   WHEN('ALTGROUP') DO;
       INPUT %SMF80HDR(REQ=EXTRACT)
             LG_OWNID $       282-289
             LG_USERN $       291-310
             LG_UTKNE $       312-315
             LG_UPRE  $       317-320
             LG_UVFYX $       322-325
             LG_UNJEU $       327-330
             LG_UUAUD $       332-335
             LG_USPEC $       337-340
             LG_UDFLT $       342-345
             LG_UUNDF $       347-350
             LG_UERR  $       352-355
             LG_UTRST $       357-360
             LG_USEST $       362-369
             LG_USURO $       371-374
             LG_URMT  $       376-379
             LG_UPRVL $       381-384
             LG_USECL $       386-393
             LG_UEXND $       395-402
             LG_USUSR $       404-411
             LG_USNOD $       413-420
             LG_USGRP $       422-429
             LG_USPOE $       431-438
             LG_USPCL $       440-447
             LG_UTUSR $       449-456
             LG_UTGRP $       458-465
             LG_UTDFT $       467-470
             LG_UTSEC $       472-475
             LG_APPC  $       477-492
             LG_GRPID $       494-501
             LG_SPEFD $       503-702
             LG_FAILD $      1528-1727
             LG_IGNRD $      2553-2752
               ;
LABEL LG_OWNID = 'Profile owner'
      LG_USERN = 'User name'
      LG_UTKNE = 'Utoken encr.?'
```

```
       LG_UPRE  = 'Pre-1.9?'
       LG_UVFYX = 'VERIFYX propagation?'
       LG_UNJEU = 'Undefined NJE user?'
       LG_UUAUD = 'UAUDIT?'
       LG_USPEC = 'RACF special?'
       LG_UDFLT = 'Default token?'
       LG_UUNDF = 'Undefined user?'
       LG_UERR  = 'Token in error?'
       LG_UTRST = 'User trusted?'
       LG_USEST = 'Session type'
       LG_USURO = 'Surrogate user?'
       LG_URMT  = 'Remote job?'
       LG_UPRVL = 'Priviliged user?'
       LG_USECL = 'User SECLABEL'
       LG_UEXND = 'Execution node'
       LG_USUSR = 'Submitting user'
       LG_USNOD = 'Submitting node'
       LG_USGRP = 'Submitting group'
       LG_USPOE = 'Port of entry'
       LG_USPCL = 'Class of POE'
       LG_UTUSR = 'Userid'
       LG_UTGRP = 'Groupid'
       LG_UTDFT = 'Default group?'
       LG_UTSEC = 'Default SECLABEL?'
       LG_APPC  = 'APPC key link'
       LG_GRPID = 'Group name'
       LG_SPEFD = 'Keywords specified'
       LG_FAILD = 'Keywords failed'
       LG_IGNRD = 'Keywords ignored'
       ;
        OUTPUT RACF.ALTGROUP;
      END;
     %END;
%MEND ALTGROUP;
./        ADD   LIST=ALL,NAME=ALTUSER
%MACRO ALTUSER(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
    %PUT Including variables from ALTUSER extension;
     RACF.ALTUSER (KEEP=%SMFHDR
                        %SMF8ØHDR(REQ=DEFINE)
                        LU_OWNID
                        LU_USERN
                        LU_OLSCL
                        LU_UTKNE
                        LU_UPRE
                        LU_UVFYX
                        LU_UNJEU
                        LU_UUAUD
```

```
                        LU_USPEC
                        LU_UDFLT
                        LU_UUNDF
                        LU_UERR
                        LU_UTRST
                        LU_USEST
                        LU_USURO
                        LU_URMT
                        LU_UPRVL
                        LU_USECL
                        LU_UEXND
                        LU_USUSR
                        LU_USNOD
                        LU_USGRP
                        LU_USPOE
                        LU_USPCL
                        LU_UTUSR
                        LU_UTGRP
                        LU_UTDFT
                        LU_UTSEC
                        LU_APPC
                        LU_NOAGL
                        LU_NOAGR
                        LU_NOAPF
                        LU_USRID
                        LU_SPEFD
                        LU_FAILD
                        LU_IGNRD
                        )
  %END;
%IF &REQ = EXTRACT %THEN
  %DO;
  %PUT Including datadefinition for ALTUSER extension;
 WHEN('ALTUSER') DO;
     INPUT %SMF80HDR(REQ=EXTRACT)
           LU_OWNID $       282-289
           LU_USERN $       291-310
           LU_OLSCL $       312-319
           LU_UTKNE $       321-324
           LU_UPRE  $       326-329
           LU_UVFYX $       331-334
           LU_UNJEU $       336-339
           LU_UUAUD $       341-344
           LU_USPEC $       346-349
           LU_UDFLT $       351-354
           LU_UUNDF $       356-359
           LU_UERR  $       361-364
           LU_UTRST $       366-369
           LU_USEST $       371-378
           LU_USURO $       380-383
```

```
              LU_URMT  $       385-388
              LU_UPRVL $       390-393
              LU_USECL $       395-402
              LU_UEXND $       404-411
              LU_USUSR $       413-420
              LU_USNOD $       422-429
              LU_USGRP $       431-438
              LU_USPOE $       440-447
              LU_USPCL $       449-456
              LU_UTUSR $       458-465
              LU_UTGRP $       467-474
              LU_UTDFT $       476-479
              LU_UTSEC $       481-484
              LU_APPC  $       486-501
              LU_NOAGL $       503-506
              LU_NOAGR $       508-511
              LU_NOAPF $       513-516
              LU_USRID $       518-525
              LU_SPEFD $       527-726
              LU_FAILD $      1552-1751
              LU_IGNRD $      2577-2776
                ;
LABEL LU_OWNID = 'Profile owner'
      LU_USERN = 'User name'
      LU_OLSCL = 'SECLABEL deleted'
      LU_UTKNE = 'Utoken encr.?'
      LU_UPRE  = 'Pre-1.9?'
      LU_UVFYX = 'VERIFYX propagation?'
      LU_UNJEU = 'Undefined NJE user?'
      LU_UUAUD = 'UAUDIT?'
      LU_USPEC = 'RACF special?'
      LU_UDFLT = 'Default token?'
      LU_UUNDF = 'Undefined user?'
      LU_UERR  = 'Token in error?'
      LU_UTRST = 'User trusted?'
      LU_USEST = 'Session type'
      LU_USURO = 'Surrogate user?'
      LU_URMT  = 'Remote job?'
      LU_UPRVL = 'Priviliged user?'
      LU_USECL = 'User SECLABEL'
      LU_UEXND = 'Execution node'
      LU_USUSR = 'Submitting user'
      LU_USNOD = 'Submitting node'
      LU_USGRP = 'Submitting group'
      LU_USPOE = 'Port of entry'
      LU_USPCL = 'Class of POE'
      LU_UTUSR = 'Userid'
      LU_UTGRP = 'Groupid'
      LU_UTDFT = 'Default group?'
      LU_UTSEC = 'Default SECLABEL?'
```

```
       LU_APPC  = 'APPC key link'
       LU_NOAGL = 'Missing CLAUTH?'
       LU_NOAGR = 'Missing group auth?'
       LU_NOAPF = 'Missing profile auth?'
       LU_USRID = 'User id'
       LU_SPEFD = 'Keywords specified'
       LU_FAILD = 'Keywords failed'
       LU_IGNRD = 'Keywords ignored'
       ;
        OUTPUT RACF.ALTUSER;
      END;
     %END;
%MEND ALTUSER;
./        ADD   LIST=ALL,NAME=CONNECT
%MACRO CONNECT(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
    %PUT Including variables from CONNECT extension;
     RACF.CONNECT (KEEP=%SMFHDR
                        %SMF8ØHDR(REQ=DEFINE)
                        CO_OWNID
                        CO_USERN
                        CO_UTKNE
                        CO_UPRE
                        CO_UVFYX
                        CO_UNJEU
                        CO_UUAUD
                        CO_USPEC
                        CO_UDFLT
                        CO_UUNDF
                        CO_UERR
                        CO_UTRST
                        CO_USEST
                        CO_USURO
                        CO_URMT
                        CO_UPRVL
                        CO_USECL
                        CO_UEXND
                        CO_USUSR
                        CO_USNOD
                        CO_USGRP
                        CO_USPOE
                        CO_USPCL
                        CO_UTUSR
                        CO_UTGRP
                        CO_UTDFT
                        CO_UTSEC
                        CO_APPC
                        CO_USRID
```

```
                        CO_SPEFD
                        CO_FAILD
                        )
    %END;
  %IF &REQ = EXTRACT %THEN
    %DO;
    %PUT Including datadefinition for CONNECT extension;
   WHEN('CONNECT') DO;
       INPUT %SMF80HDR(REQ=EXTRACT)
             CO_OWNID $       282-289
             CO_USERN $       291-310
             CO_UTKNE $       312-315
             CO_UPRE  $       317-320
             CO_UVFYX $       322-325
             CO_UNJEU $       327-330
             CO_UUAUD $       332-335
             CO_USPEC $       337-340
             CO_UDFLT $       342-345
             CO_UUNDF $       347-350
             CO_UERR  $       352-355
             CO_UTRST $       357-360
             CO_USEST $       362-369
             CO_USURO $       371-374
             CO_URMT  $       376-379
             CO_UPRVL $       381-384
             CO_USECL $       386-393
             CO_UEXND $       395-402
             CO_USUSR $       404-411
             CO_USNOD $       413-420
             CO_USGRP $       422-429
             CO_USPOE $       431-438
             CO_USPCL $       440-447
             CO_UTUSR $       449-456
             CO_UTGRP $       458-465
             CO_UTDFT $       467-470
             CO_UTSEC $       472-475
             CO_APPC  $       477-492
             CO_USRID $       494-501
             CO_SPEFD $       503-702
             CO_FAILD $      1528-1727
               ;
LABEL CO_OWNID = 'Profile owner'
      CO_USERN = 'User name'
      CO_UTKNE = 'Utoken encr.?'
      CO_UPRE  = 'Pre-1.9?'
      CO_UVFYX = 'VERIFYX propagation?'
      CO_UNJEU = 'Undefined NJE user?'
      CO_UUAUD = 'UAUDIT?'
      CO_USPEC = 'RACF special?'
      CO_UDFLT = 'Default token?'
```

```
       CO_UUNDF = 'Undefined user?'
       CO_UERR  = 'Token in error?'
       CO_UTRST = 'User trusted?'
       CO_USEST = 'Session type'
       CO_USURO = 'Surrogate user?'
       CO_URMT  = 'Remote job?'
       CO_UPRVL = 'Priviliged user?'
       CO_USECL = 'User SECLABEL'
       CO_UEXND = 'Execution node'
       CO_USUSR = 'Submitting user'
       CO_USNOD = 'Submitting node'
       CO_USGRP = 'Submitting group'
       CO_USPOE = 'Port of entry'
       CO_USPCL = 'Class of POE'
       CO_UTUSR = 'Userid'
       CO_UTGRP = 'Groupid'
       CO_UTDFT = 'Default group?'
       CO_UTSEC = 'Default SECLABEL?'
       CO_APPC  = 'APPC key link'
       CO_USRID = 'User id'
       CO_SPEFD = 'Keywords specified'
       CO_FAILD = 'Keywords failed'
       ;
        OUTPUT RACF.CONNECT;
      END;
     %END;
%MEND CONNECT;
./        ADD   LIST=ALL,NAME=DELDSD
%MACRO DELDSD(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
     %PUT Including variables from DELDSD extension;
     RACF.DELDSD (KEEP=%SMFHDR
                       %SMF8ØHDR(REQ=DEFINE)
                       DD_OWNID
                       DD_USERN
                       DD_OLSCL
                       DD_UTKNE
                       DD_UPRE
                       DD_UVFYX
                       DD_UNJEU
                       DD_UUAUD
                       DD_USPEC
                       DD_UDFLT
                       DD_UUNDF
                       DD_UERR
                       DD_UTRST
                       DD_USEST
                       DD_USURO
```

```
                      DD_URMT
                      DD_UPRVL
                      DD_USECL
                      DD_UEXND
                      DD_USUSR
                      DD_USNOD
                      DD_USGRP
                      DD_USPOE
                      DD_USPCL
                      DD_UTUSR
                      DD_UTGRP
                      DD_UTDFT
                      DD_UTSEC
                      DD_APPC
                      DD_SECLL
                      DD_DSNAM
                      DD_SPEFD
                      DD_FAILD
                      )
  %END;
%IF &REQ = EXTRACT %THEN
  %DO;
  %PUT Including datadefinition for DELDSD extension;
 WHEN('DELDSD') DO;
     INPUT %SMF80HDR(REQ=EXTRACT)
           DD_OWNID $       282-289
           DD_USERN $       291-310
           DD_OLSCL $       312-319
           DD_UTKNE $       321-324
           DD_UPRE  $       326-329
           DD_UVFYX $       331-334
           DD_UNJEU $       336-339
           DD_UUAUD $       341-344
           DD_USPEC $       346-349
           DD_UDFLT $       351-354
           DD_UUNDF $       356-359
           DD_UERR  $       361-364
           DD_UTRST $       366-369
           DD_USEST $       371-378
           DD_USURO $       380-383
           DD_URMT  $       385-388
           DD_UPRVL $       390-393
           DD_USECL $       395-402
           DD_UEXND $       404-411
           DD_USUSR $       413-420
           DD_USNOD $       422-429
           DD_USGRP $       431-438
           DD_USPOE $       440-447
           DD_USPCL $       449-456
           DD_UTUSR $       458-465
```

```
              DD_UTGRP $       467-474
              DD_UTDFT $       476-479
              DD_UTSEC $       481-484
              DD_APPC  $       486-5Ø1
              DD_SECLL $       5Ø3-518
              DD_DSNAM $       52Ø-563
              DD_SPEFD $       565-764
              DD_FAILD $      159Ø-1789
              DD_IGNRD $      2615-2814
                ;
LABEL DD_OWNID = 'Profile owner'
      DD_USERN = 'User name'
      DD_OLSCL = 'SECLABEL deleted'
      DD_UTKNE = 'Utoken encr.?'
      DD_UPRE  = 'Pre-1.9?'
      DD_UVFYX = 'VERIFYX propagation?'
      DD_UNJEU = 'Undefined NJE user?'
      DD_UUAUD = 'UAUDIT?'
      DD_USPEC = 'RACF special?'
      DD_UDFLT = 'Default token?'
      DD_UUNDF = 'Undefined user?'
      DD_UERR  = 'Token in error?'
      DD_UTRST = 'User trusted?'
      DD_USEST = 'Session type'
      DD_USURO = 'Surrogate user?'
      DD_URMT  = 'Remote job?'
      DD_UPRVL = 'Priviliged user?'
      DD_USECL = 'User SECLABEL'
      DD_UEXND = 'Execution node'
      DD_USUSR = 'Submitting user'
      DD_USNOD = 'Submitting node'
      DD_USGRP = 'Submitting group'
      DD_USPOE = 'Port of entry'
      DD_USPCL = 'Class of POE'
      DD_UTUSR = 'Userid'
      DD_UTGRP = 'Groupid'
      DD_UTDFT = 'Default group?'
      DD_UTSEC = 'Default SECLABEL?'
      DD_APPC  = 'APPC key link'
      DD_SECLL = 'SECLABEL link'
      DD_DSNAM = 'Dataset name'
      DD_SPEFD = 'Keywords specified'
      DD_FAILD = 'Keywords failed'
      ;
       OUTPUT RACF.DELDSD;
     END;
    %END;
%MEND DELDSD;
./        ADD   LIST=ALL,NAME=DELGROUP
%MACRO DELGROUP(REQ=);
```

```
%LET REQ = %UPCASE(&REQ);
%IF &REQ = DEFINE %THEN
  %DO;
  %PUT Including variables from DELGROUP extension;
   RACF.DELGROUP (KEEP=%SMFHDR
                       %SMF80HDR(REQ=DEFINE)
                       DG_OWNID
                       DG_USERN
                       DG_UTKNE
                       DG_UPRE
                       DG_UVFYX
                       DG_UNJEU
                       DG_UUAUD
                       DG_USPEC
                       DG_UDFLT
                       DG_UUNDF
                       DG_UERR
                       DG_UTRST
                       DG_USEST
                       DG_USURO
                       DG_URMT
                       DG_UPRVL
                       DG_USECL
                       DG_UEXND
                       DG_USUSR
                       DG_USNOD
                       DG_USGRP
                       DG_USPOE
                       DG_USPCL
                       DG_UTUSR
                       DG_UTGRP
                       DG_UTDFT
                       DG_UTSEC
                       DG_APPC
                       DG_GRPID
                       DG_SPEFD
                       )
  %END;
%IF &REQ = EXTRACT %THEN
  %DO;
  %PUT Including datadefinition for DELGROUP extension;
 WHEN('DELGROUP') DO;
     INPUT %SMF80HDR(REQ=EXTRACT)
           DG_OWNID $       282-289
           DG_USERN $       291-310
           DG_UTKNE $       312-315
           DG_UPRE  $       317-320
           DG_UVFYX $       322-325
           DG_UNJEU $       327-330
           DG_UUAUD $       332-335
```

```
                DG_USPEC $       337-340
                DG_UDFLT $       342-345
                DG_UUNDF $       347-350
                DG_UERR  $       352-355
                DG_UTRST $       357-360
                DG_USEST $       362-369
                DG_USURO $       371-374
                DG_URMT  $       376-379
                DG_UPRVL $       381-384
                DG_USECL $       386-393
                DG_UEXND $       395-402
                DG_USUSR $       404-411
                DG_USNOD $       413-420
                DG_USGRP $       422-429
                DG_USPOE $       431-438
                DG_USPCL $       440-447
                DG_UTUSR $       449-456
                DG_UTGRP $       458-465
                DG_UTDFT $       467-470
                DG_UTSEC $       472-475
                DG_APPC  $       477-492
                DG_GRPID $       494-501
                DG_SPEFD $       503-702
                  ;
LABEL DG_OWNID = 'Profile owner'
      DG_USERN = 'User name'
      DG_UTKNE = 'Utoken encr.?'
      DG_UPRE  = 'Pre-1.9?'
      DG_UVFYX = 'VERIFYX propagation?'
      DG_UNJEU = 'Undefined NJE user?'
      DG_UUAUD = 'UAUDIT?'
      DG_USPEC = 'RACF special?'
      DG_UDFLT = 'Default token?'
      DG_UUNDF = 'Undefined user?'
      DG_UERR  = 'Token in error?'
      DG_UTRST = 'User trusted?'
      DG_USEST = 'Session type'
      DG_USURO = 'Surrogate user?'
      DG_URMT  = 'Remote job?'
      DG_UPRVL = 'Priviliged user?'
      DG_USECL = 'User SECLABEL'
      DG_UEXND = 'Execution node'
      DG_USUSR = 'Submitting user'
      DG_USNOD = 'Submitting node'
      DG_USGRP = 'Submitting group'
      DG_USPOE = 'Port of entry'
      DG_USPCL = 'Class of POE'
      DG_UTUSR = 'Userid'
      DG_UTGRP = 'Groupid'
      DG_UTDFT = 'Default group?'
```

```
      DG_UTSEC = 'Default SECLABEL?'
      DG_APPC  = 'APPC key link'
      DG_GRPID = 'Group name'
      DG_SPEFD = 'Keywords specified'
      ;
       OUTPUT RACF.DELGROUP;
     END;
    %END;
%MEND DELGROUP;
./        ADD   LIST=ALL,NAME=DELUSER
%MACRO DELUSER(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
    %PUT Including variables from DELUSER extension;
     RACF.DELUSER (KEEP=%SMFHDR
                        %SMF8ØHDR(REQ=DEFINE)
                        DU_OWNID
                        DU_USERN
                        DU_UTKNE
                        DU_UPRE
                        DU_UVFYX
                        DU_UNJEU
                        DU_UUAUD
                        DU_USPEC
                        DU_UDFLT
                        DU_UUNDF
                        DU_UERR
                        DU_UTRST
                        DU_USEST
                        DU_USURO
                        DU_URMT
                        DU_UPRVL
                        DU_USECL
                        DU_UEXND
                        DU_USUSR
                        DU_USNOD
                        DU_USGRP
                        DU_USPOE
                        DU_USPCL
                        DU_UTUSR
                        DU_UTGRP
                        DU_UTDFT
                        DU_UTSEC
                        DU_APPC
                        DU_USRID
                        DU_SPEFD
                        )
    %END;
  %IF &REQ = EXTRACT %THEN
```

```
    %DO;
    %PUT Including datadefinition for DELUSER extension;
   WHEN('DELUSER') DO;
       INPUT %SMF80HDR(REQ=EXTRACT)
             DU_OWNID $       282-289
             DU_USERN $       291-310
             DU_UTKNE $       312-315
             DU_UPRE  $       317-320
             DU_UVFYX $       322-325
             DU_UNJEU $       327-330
             DU_UUAUD $       332-335
             DU_USPEC $       337-340
             DU_UDFLT $       342-345
             DU_UUNDF $       347-350
             DU_UERR  $       352-355
             DU_UTRST $       357-360
             DU_USEST $       362-369
             DU_USURO $       371-374
             DU_URMT  $       376-379
             DU_UPRVL $       381-384
             DU_USECL $       386-393
             DU_UEXND $       395-402
             DU_USUSR $       404-411
             DU_USNOD $       413-420
             DU_USGRP $       422-429
             DU_USPOE $       431-438
             DU_USPCL $       440-447
             DU_UTUSR $       449-456
             DU_UTGRP $       458-465
             DU_UTDFT $       467-470
             DU_UTSEC $       472-475
             DU_APPC  $       477-492
             DU_USRID $       494-501
             DU_SPEFD $       503-702
               ;
LABEL DU_OWNID = 'Profile owner'
      DU_USERN = 'User name'
      DU_UTKNE = 'Utoken encr.?'
      DU_UPRE  = 'Pre-1.9?'
      DU_UVFYX = 'VERIFYX propagation?'
      DU_UNJEU = 'Undefined NJE user?'
      DU_UUAUD = 'UAUDIT?'
      DU_USPEC = 'RACF special?'
      DU_UDFLT = 'Default token?'
      DU_UUNDF = 'Undefined user?'
      DU_UERR  = 'Token in error?'
      DU_UTRST = 'User trusted?'
      DU_USEST = 'Session type'
      DU_USURO = 'Surrogate user?'
      DU_URMT  = 'Remote job?'
```

```
      DU_UPRVL = 'Priviliged user?'
      DU_USECL = 'User SECLABEL'
      DU_UEXND = 'Execution node'
      DU_USUSR = 'Submitting user'
      DU_USNOD = 'Submitting node'
      DU_USGRP = 'Submitting group'
      DU_USPOE = 'Port of entry'
      DU_USPCL = 'Class of POE'
      DU_UTUSR = 'Userid'
      DU_UTGRP = 'Groupid'
      DU_UTDFT = 'Default group?'
      DU_UTSEC = 'Default SECLABEL?'
      DU_APPC  = 'APPC key link'
      DU_USRID = 'User id'
      DU_SPEFD = 'Keywords specified'
      ;
       OUTPUT RACF.DELUSER;
     END;
    %END;
%MEND DELUSER;
./        ADD   LIST=ALL,NAME=PASSWORD
%MACRO PASSWORD(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
      %PUT Including variables from PASSWORD extension;
      RACF.PASSWORD (KEEP=%SMFHDR
                          %SMF8ØHDR(REQ=DEFINE)
                          PWDOWNID
                          PWDUSERN
                          PWDUTKNE
                          PWDUPRE
                          PWDUVFYX
                          PWDUNJEU
                          PWDUUAUD
                          PWDUSPEC
                          PWDUDFLT
                          PWDUUNDF
                          PWDUERR
                          PWDUTRST
                          PWDUSEST
                          PWDUSURO
                          PWDURMT
                          PWDUPRVL
                          PWDUSECL
                          PWDUEXND
                          PWDUSUSR
                          PWDUSNOD
                          PWDUSGRP
                          PWDUSPOE
```

```
                          PWDUSPCL
                          PWDUTUSR
                          PWDUTGRP
                          PWDUTDFT
                          PWDUTSEC
                          PWDAPPC
                          PWDSPEFD
                          PWDFAILD
                          PWDIGNRD
                       )
    %END;
  %IF &REQ = EXTRACT %THEN
    %DO;
      %PUT Including datadefinition for PASSWORD extension;
      WHEN('PASSWORD') DO;
       INPUT %SMF80HDR(REQ=EXTRACT)
             PWDOWNID $       282-289
             PWDUSERN $       291-310
             PWDUTKNE $       312-315
             PWDUPRE  $       317-320
             PWDUVFYX $       322-325
             PWDUNJEU $       327-330
             PWDUUAUD $       332-335
             PWDUSPEC $       337-340
             PWDUDFLT $       342-345
             PWDUUNDF $       347-350
             PWDUERR  $       352-355
             PWDUTRST $       357-360
             PWDUSEST $       362-369
             PWDUSURO $       371-374
             PWDURMT  $       376-379
             PWDUPRVL $       381-384
             PWDUSECL $       386-393
             PWDUEXND $       395-402
             PWDUSUSR $       404-411
             PWDUSNOD $       413-420
             PWDUSGRP $       422-429
             PWDUSPOE $       431-438
             PWDUSPCL $       440-447
             PWDUTUSR $       449-456
             PWDUTGRP $       458-465
             PWDUTDFT $       467-470
             PWDUTSEC $       472-475
             PWDAPPC  $       477-492
             PWDSPEFD $       494-693
             PWDFAILD $      1519-1718
             PWDIGNRD $      2544-2743
             ;
LABEL PWDOWNID = 'Profile owner'
      PWDUSERN = 'User name'
```

```
      PWDUTKNE = 'Utoken encr.?'
      PWDUPRE  = 'Pre-1.9?'
      PWDUVFYX = 'VERIFYX propagation?'
      PWDUNJEU = 'Undefined NJE user?'
      PWDUUAUD = 'UAUDIT?'
      PWDUSPEC = 'RACF special?'
      PWDUDFLT = 'Default token?'
      PWDUUNDF = 'Undefined user?'
      PWDUERR  = 'Token in error?'
      PWDUTRST = 'User trusted?'
      PWDUSEST = 'Session type'
      PWDUSURO = 'Surrogate user?'
      PWDURMT  = 'Remote job?'
      PWDUPRVL = 'Priviliged user?'
      PWDUSECL = 'User SECLABEL'
      PWDUEXND = 'Execution node'
      PWDUSUSR = 'Submitting user'
      PWDUSNOD = 'Submitting node'
      PWDUSGRP = 'Submitting group'
      PWDUSPOE = 'Port of entry'
      PWDUSPCL = 'Class of POE'
      PWDUTUSR = 'Userid'
      PWDUTGRP = 'Groupid'
      PWDUTDFT = 'Default group?'
      PWDUTSEC = 'Default SECLABEL?'
      PWDAPPC  = 'APPC key link'
      PWDSPEFD = 'User id'
      PWDFAILD = 'Keywords specified'
      PWDIGNRD = 'Keywords ignored'
      ;
        OUTPUT RACF.PASSWORD;
      END;
    %END;
%MEND PASSWORD;
./        ADD   LIST=ALL,NAME=PERMIT
%MACRO PERMIT(REQ=);
  %LET REQ = %UPCASE(&REQ);
  %IF &REQ = DEFINE %THEN
    %DO;
      %PUT Including variables from PERMIT extension;
      RACF.PERMIT (KEEP=%SMFHDR
                        %SMF8ØHDR(REQ=DEFINE)
                        PERCLASS
                        PEROWNID
                        PERUSERN
                        PERUTKNE
                        PERUPRE
                        PERUVFYX
                        PERUNJEU
                        PERUUAUD
```

```
                        PERUSPEC
                        PERUDFLT
                        PERUUNDF
                        PERUERR
                        PERUTRST
                        PERUSEST
                        PERUSURO
                        PERURMT
                        PERUPRVL
                        PERUSECL
                        PERUEXND
                        PERUSUSR
                        PERUSNOD
                        PERUSGRP
                        PERUSPOE
                        PERUSPCL
                        PERUTUSR
                        PERUTGRP
                        PERUTDFT
                        PERUTSEC
                        PERAPPC
                        PERRESNM
                        PERSPEFD
                        PERFAILD
                        PERIGNRD
                    )
  %END;
%IF &REQ = EXTRACT %THEN
  %DO;
    %PUT Including datadefinition for PERMIT extension;
    WHEN('PERMIT') DO;
      INPUT %SMF8ØHDR(REQ=EXTRACT)
            PERCLASS $       282-289
            PEROWNID $       291-298
            PERUSERN $       3ØØ-319
            PERUTKNE $       321-324
            PERUPRE  $       326-329
            PERUVFYX $       331-334
            PERUNJEU $       336-339
            PERUUAUD $       341-344
            PERUSPEC $       346-349
            PERUDFLT $       351-354
            PERUUNDF $       356-359
            PERUERR  $       361-364
            PERUTRST $       366-369
            PERUSEST $       371-378
            PERUSURO $       38Ø-383
            PERURMT  $       385-388
            PERUPRVL $       39Ø-393
            PERUSECL $       395-4Ø2
```

```
              PERUEXND $       404-411
              PERUSUSR $       413-420
              PERUSNOD $       422-429
              PERUSGRP $       431-438
              PERUSPOE $       440-447
              PERUSPCL $       449-456
              PERUTUSR $       458-465
              PERUTGRP $       467-474
              PERUTDFT $       476-479
              PERUTSEC $       481-484
              PERAPPC  $       486-501
              PERRESNM $       503-702
              PERSPEFD $       759-958
              PERFAILD $      1784-1983
              PERIGNRD $      2809-3008
               ;
LABEL PERCLASS = 'Class name'
      PEROWNID = 'Profile owner'
      PERUSERN = 'User name'
      PERUTKNE = 'Utoken encr.?'
      PERUPRE  = 'Pre-1.9?'
      PERUVFYX = 'VERIFYX propagation?'
      PERUNJEU = 'Undefined NJE user?'
      PERUUAUD = 'UAUDIT?'
      PERUSPEC = 'RACF special?'
      PERUDFLT = 'Default token?'
      PERUUNDF = 'Undefined user?'
      PERUERR  = 'Token in error?'
      PERUTRST = 'User trusted?'
      PERUSEST = 'Session type'
      PERUSURO = 'Surrogate user?'
      PERURMT  = 'Remote job?'
      PERUPRVL = 'Priviliged user?'
      PERUSECL = 'User SECLABEL'
      PERUEXND = 'Execution node'
      PERUSUSR = 'Submitting user'
      PERUSNOD = 'Submitting node'
      PERUSGRP = 'Submitting group'
      PERUSPOE = 'Port of entry'
      PERUSPCL = 'Class of POE'
      PERUTUSR = 'Userid'
      PERUTGRP = 'Groupid'
      PERUTDFT = 'Default group?'
      PERUTSEC = 'Default SECLABEL?'
      PERAPPC  = 'APPC key link'
      PERRESNM = 'Resource name'
      PERSPEFD = 'Keywords specified'
      PERFAILD = 'Keywords failed'
      PERIGNRD = 'Keywords ignored'
      ;
```

```
        OUTPUT RACF.PERMIT;
      END;
    %END;
%MEND PERMIT;
```

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

Transferring code from the Web to a mainframe

Editor's note: the ISPF edit macro in this article, or a modified version (once you've identified the 'before' and 'after' hex codes at your site), can be used to overcome problems experienced when downloading Update *code to a mainframe.*

When a colleague of mine recently downloaded an *MVS Update* article from the Xephon Web site to his PC and then uploaded it to his MVS system, he found to his disappointment that the program code would not run properly.

It was a REXX program, and, when he executed it, he received the following message:

```
IRX0013I Error running XXXXXXXX, line nn: Invalid character in program
```

This was rather puzzling, but a quick look at the code revealed that the offending character was a REXX 'not' (that is ^, in a ^= expression), which should be a hex value X'5F', but was instead a X'B0'. The REXX interpreter was rejecting this value. Another odd character turned out to be the '|' operator, which should be X'4F', but was X'6A'.

Having discovered this, it was easy to code an ISPF edit macro to fix this and to cater for it in future uploads:

```
ISREDIT MACRO
ISREDIT CHANGE ALL X'B0' X'5F'
ISREDIT CHANGE ALL X'6A' X'4F'
EXIT
```

The PC was running IBM Personal Communications 3270 Version 4.1 for Windows with an IEEE 802.2 connection to the host, code page 037. The upload was achieved using the IBM 3270 PC File Transfer Program for MVS/TSO Release 1.1.1 using the following command:

```
IND$FILE PUT XEPHFILE.TEXT ASCII CRLF RECFM(V) LRECL(133)
```

It seems that the ASCII to EBCDIC conversion taking place works fine for alphanumeric characters, but is suspect for unusual ones. Readers should be aware of this when transferring code.

Patrick Mullen
MVS Systems Consultant (Canada)

August 1995 – May 1998 index

Items below are references to articles that have appeared in *RACF Update* since August 1995. References show the issue number followed by the page number(s). All these back-issues of *RACF Update* can be ordered from Xephon. See page 2 for details.

RACF news

IBM has announced Version 2 Release 5 of OS/390, which includes extensions to RACF and Distributed Computing Environment, and is said to make it easier to extend the security model to e-business and server consolidation. Specific improvements include the Lightweight Directory Access Protocol, digital certificates, hardware cryptography support for Secure Electronic Transaction, and firewall technologies. Security plays a major role in the new release, coupled with security enhancements to the System/390 architecture generally.

The OS itself includes improvements to security services with the integration of the Triple DES data encryption cryptographic standard, which makes use of the hardware-embedded System/390 cryptographic coprocessor in the System/390 G4 Server.

IBM has also announced the release of Netview Access Services (NAS) Version 2.1.1, which uses RACF definitions to determine whether a user or group of users should access a VTAM application. Version 2.1.1 can also display up to 999 application selection screen pages, each containing as many as 42 applications.

For further information contact your local IBM representative.

* * *

MegaSolve has announced MegaSolve Security Server Administrator, the new version of its RACF add-on software, which is said to reduce the overhead and cost of administering RACF security. MegaSolve/ SSA is modular in design and pricing.

Features include on-line generic search capability; fast and flexible command generation; full-function security; automatic scheduling of jobs or commands; centralized or decentralized password and connect administration without having group/global special; and extensive reporting.

For further information contact:
MegaSolve, 22222 Sherman Way, Suite 208, Canoga Park, CA 91303-1055, USA.
Tel: (888) 798 4388.

* * *

BETA Systems has announced Version 3.1.2 of BETA 88, its host-based Systems' Enterprise Security Manager for RACF administration. The new modification supports the newest RACF features in OS/390 Releases 3 and 4 and makes RACF administration easier by taking over most of the redundant routine jobs. It also improves security administration with added functions not provided by RACF. Detailed analyses of all RACF protected resources can be generated, and user-profiles can be easily managed. Significantly, DB2 security data can now be administered just as RACF data. BETA 88 can be used to view, modify, query, and generate reports from any security-related DB2 data covered by RACF. It also provides the RACF administrator with solutions for VTAM, CICS, and IMS environments, allowing decentralized RACF administration in all departments.

For further information contact:
BETA Systems Software, One Securities Center, 3490 Piedmont Road, Suite 1100, Atlanta, GA 30305, USA.
Tel: (404) 812 1556.
BETA Systems Software, Highlands House, Basingstoke Road, Spencers Wood, Reading, RG7 1NT, UK.
Tel: (01734) 885175.