```
UUU           UUU     AAAAAAAAA      FFFFFFFFFFFFFFF
UUU           UUU     AAAAAAAAA      FFFFFFFFFFFFFFF
UUU           UUU     AAAAAAAAA      FFFFFFFFFFFFFFF
UUU           UUU  AAA           AAA  FFF
UUU           UUU  AAA           AAA  FFF
UUU           UUU  AAA           AAA  FFF
UUU           UUU  AAA           AAA  FFF
UUU           UUU  AAA           AAA  FFF
UUU           UUU  AAA           AAA  FFF
UUU           UUU  AAA           AAA  FFFFFFFFFFFF
UUU           UUU  AAA           AAA  FFFFFFFFFFFF
UUU           UUU  AAA           AAA  FFFFFFFFFFFF
UUU           UUU  AAAAAAAAAAAAAAA  FFF
UUU           UUU  AAAAAAAAAAAAAAA  FFF
UUU           UUU  AAAAAAAAAAAAAAA  FFF
UUU           UUU  AAA           AAA  FFF
UUU           UUU  AAA           AAA  FFF
UUU           UUU  AAA           AAA  FFF
UUUUUUUUUUUUUUUU  AAA           AAA  FFF
UUUUUUUUUUUUUUUU  AAA           AAA  FFF
UUUUUUUUUUUUUUUU  AAA           AAA  FFF
```

```
_$2

Val
----
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0003
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
000
000
0000
000
000
000
000
```

```
RRRRRRRR      IIIIII      GGGGGGGG  HH      HH  TTTTTTTTTT    SSSSSSSS  MM      MM    AAAAAA    NN      NN
RRRRRRRR      IIIIII      GGGGGGGG  HH      HH  TTTTTTTTTT    SSSSSSSS  MM      MM    AAAAAA    NN      NN
RR      RR      II      GG          HH      HH      TT      SS          MMMM  MMMM  AA      AA  NN      NN
RR      RR      II      GG          HH      HH      TT      SS          MMMM  MMMM  AA      AA  NN      NN
RR      RR      II      GG          HH      HH      TT      SS          MM  MM  MM  AA      AA  NNNN    NN
RR      RR      II      GG          HH      HH      TT      SS          MM  MM  MM  AA      AA  NNNN    NN
RRRRRRRR        II      GG          HHHHHHHHHH      TT        SSSSSS    MM      MM  AA      AA  NN  NN  NN
RRRRRRRR        II      GG          HHHHHHHHHH      TT        SSSSSS    MM      MM  AA      AA  NN  NN  NN
RR  RR          II      GG  GGGGGG  HH      HH      TT              SS  MM      MM  AAAAAAAAAA  NN    NNNN
RR  RR          II      GG  GGGGGG  HH      HH      TT              SS  MM      MM  AAAAAAAAAA  NN    NNNN
RR    RR        II      GG      GG  HH      HH      TT              SS  MM      MM  AA      AA  NN      NN      ....
RR    RR        II      GG      GG  HH      HH      TT              SS  MM      MM  AA      AA  NN      NN      ....
RR      RR    IIIIII      GGGGGG    HH      HH      TT      SSSSSSSS    MM      MM  AA      AA  NN      NN      ....
RR      RR    IIIIII      GGGGGG    HH      HH      TT      SSSSSSSS    MM      MM  AA      AA  NN      NN      ....

LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

```
;     1          0001  0 MODULE RIGHTSMAN (
;     2          0002  0                 LANGUAGE (BLISS32),
;     3          0003  0                 IDENT = 'V04-000',
;     4          0004  0                 ADDRESSING_MODE (EXTERNAL=GENERAL, nonexternal=general)
;     5          0005  0                 ) =
;     6          0006  1 BEGIN
;     7          0007  1
;     8          0008  1 !
;     9          0009  1 !*****************************************************************************
;    10          0010  1 !*                                                                           *
;    11          0011  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;    12          0012  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
;    13          0013  1 !*  ALL RIGHTS RESERVED.                                                      *
;    14          0014  1 !*                                                                           *
;    15          0015  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
;    16          0016  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
;    17          0017  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
;    18          0018  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
;    19          0019  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
;    20          0020  1 !*  TRANSFERRED.                                                              *
;    21          0021  1 !*                                                                           *
;    22          0022  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
;    23          0023  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT     *
;    24          0024  1 !*  CORPORATION.                                                              *
;    25          0025  1 !*                                                                           *
;    26          0026  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
;    27          0027  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
;    28          0028  1 !*                                                                           *
;    29          0029  1 !*                                                                           *
;    30          0030  1 !*****************************************************************************
;    31          0031  1
;    32          0032  1 !++
;    33          0033  1 ! FACILITY:      System Management Utility Program
;    34          0034  1 !
;    35          0035  1 ! ABSTRACT:
;    36          0036  1 !
;    37          0037  1 !       This program allows the system manager to maintain the rights
;    38          0038  1 !       data base.  The data base is a single ISAM file which is used
;    39          0039  1 !       to record the names, encoding, and holders of all identifiers
;    40          0040  1 !       in the system.
;    41          0041  1 !
;    42          0042  1 ! ENVIRONMENT:
;    43          0043  1 !
;    44          0044  1 ! AUTHOR:        Larry Yetto , CREATION DATE: 22-MAR-1984
;    45          0045  1 !
;    46          0046  1 ! MODIFIED BY:
;    47          0047  1 !
;    48          0048  1 !       V03-005 JRL0018         John R. Lawson, Jr.     02-Jul-1984 15:26
;    49          0049  1 !               ADD/IDENTIFIER FOOBAR /VALUE:UIC=[*,*] is O.K., but
;    50          0050  1 !               ADD/IDENTIFIER FOOBAR /VALUE:UIC=[*,27] isn't.
;    51          0051  1 !
;    52          0052  1 !       V03-004 JRL0002         John R. Lawson, Jr.     14-Jun-1984 15:38
;    53          0053  1 !               Replace all messages with calls to LIB$SIGNAL and place
;    54          0054  1 !               those messages in a .MSG file
;    55          0055  1 !
;    56          0056  1 !       V03-003 LY0495          Larry Yetto             11-JUN-1984 13:01
;    57          0057  1 !               Correct noise error message being reported by ADD/ID/USER=*
```

```
;    58          0058  1 !                caused by improperly initialized flag.
;    59          0059  1 !
;    60          0060  1 !        V03-002 LY0479          Larry Yetto             27-APR-1984 08:28
;    61          0061  1 !                Do not delete the user identifier during a REMOVE command
;    62          0062  1 !                unless there are no longer any UAF records with
;    63          0063  1 !                this UIC
;    64          0064  1 !                Base Level QAR #182  -  Fix accvio in rename/ident
;    65          0065  1 !
;    66          0066  1 !        V03-001 LY0476          Larry Yetto              9-APR-1984 10:29
;    67          0067  1 !                Fix endless loop in SHOW/ID/USER=*
;    68          0068  1 !
;    69          0069  1 !
;    70          0070  1 !--
;    71          0071  1
```

```
;    73          0072  1 !
;    74          0073  1 ! TABLE OF CONTENTS:
;    75          0074  1 !
;    76          0075  1
;    77          0076  1 FORWARD ROUTINE
;    78          0077  1     UAF$ADD_GRP_IDENT           : NOVALUE ,
;    79          0078  1     UAF$ADD_IDENT               : NOVALUE ,
;    80          0079  1     UAF$ADD_IDENT_RECBUF ,
;    81          0080  1     UAF$BUILD_HOLDER ,
;    82          0081  1     UAF$CREATE_RDB              : NOVALUE ,
;    83          0082  1     UAF$DISPLAY_IDENT           : NOVALUE ,
;    84          0083  1     UAF$DISPLAY_IDENT_RECBUF ,
;    85          0084  1     UAF$DISPLAY_RIGHTS          : NOVALUE ,
;    86          0085  1     UAF$DISPLAY_RIGHTS_RECBUF ,
;    87          0086  1     UAF$FIND_UIC ,
;    88          0087  1     UAF$GRANT_IDENT             : NOVALUE ,
;    89          0088  1     UAF$RECORD_FOUND ,
;    90          0089  1     UAF$LIST_IDENT              : NOVALUE ,
;    91          0090  1     UAF$LIST_RIGHTS             : NOVALUE ,
;    92          0091  1     UAF$MODIFY_IDENT            : NOVALUE ,
;    93          0092  1     UAF$REMOVE_GRP_IDENT        : NOVALUE ,
;    94          0093  1     UAF$REMOVE_IDENT            : NOVALUE ,
;    95          0094  1     UAF$REMOVE_IDENT_RECBUF ,
;    96          0095  1     UAF$RENAME_IDENT            : NOVALUE ,
;    97          0096  1     UAF$REVOKE_IDENT            : NOVALUE ,
;    98          0097  1     UAF$SHOW_IDENT              : NOVALUE ,
;    99          0098  1     UAF$SHOW_RIGHTS             : NOVALUE ,
;   100          0099  1     UAF$WRITE_HOLDERS ,
;   101          0100  1     UAF$WRITE_IDENT ,
;   102          0101  1     UAF$WRITE_RIGHTS ;
;   103          0102  1
;   104          0103  1 EXTERNAL ROUTINE
;   105          0104  1     CLI$PRESENT ,
;   106          0105  1     CLI$GET_VALUE ,
;   107          0106  1     FAOOUT ,
;   108          0107  1     FMT_SYS_MSG ,
;   109          0108  1     GETVAL ,
;   110          0109  1     PARSE_UIC ,
;   111          0110  1     PARSE_WILD ,
;   112          0111  1     PARSE_WILD_UIC ,
;   113          0112  1     WILD_USER ;
;   114          0113  1
;   115          0114  1 !
;   116          0115  1 ! INCLUDE FILES:
;   117          0116  1 !
;   118          0117  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;   119          0118  1 LIBRARY 'SYS$LIBRARY:TPAMAC.L32';
;   120          0119  1
;   121          0120  1
;   122          0121  1 !
;   123          0122  1 ! Field, structure, and Macro definitions
;   124          0123  1 !
;   125          0124  1 FIELD
;   126          0125  1     DESCR_FIELDS =                      ! Define the fields for a DESCRIPTOR
;   127          0126  1         SET
;   128          0127  1         LENGTH  = [DSC$W_LENGTH],
;   129          0128  1         DTYPE   = [DSC$B_DTYPE],
```

```
;  130          0129  1          CLASS   = [DSC$B_CLASS],
;  131          0130  1          POINTER = [DSC$A_POINTER]
;  132          0131  1          TES;
;  133          0132  1
;  134          0133  1
;  135          0134  1  !++
;  136          0135  1  ! Structure for all MDL defined blocks.
;  137          0136  1  !
;  138          0137  1  !--
;  139          0138  1  !
;  140          0139  1
;  141          0140  1  MACRO
;  142        M 0141  1      STRINGDESC =
;  143          0142  1          $BBLOCK [DSC$K_S_BLN] FIELD (DESCR_FIELDS)%;
;  144          0143  1
;  145          0144  1  MACRO
;  146        M 0145  1      STATDESC =
;  147        M 0146  1          $BBLOCK [DSC$K_S_BLN] FIELD (DESCR_FIELDS)
;  148        M 0147  1                  PRESET ( [LENGTH]  = 0,
;  149        M 0148  1                           [DTYPE]   = DSC$K_DTYPE_T,
;  150        M 0149  1                           [CLASS]   = DSC$K_CLASS_S,
;  151          0150  1                           [POINTER] = 0)%;
;  152          0151  1
;  153          0152  1  MACRO
;  154        M 0153  1      DYNAMICDESC =
;  155        M 0154  1          $BBLOCK [DSC$K_D_BLN] FIELD (DESCR_FIELDS)
;  156        M 0155  1                  PRESET ( [LENGTH]  = 0,
;  157        M 0156  1                           [DTYPE]   = 0,
;  158        M 0157  1                           [CLASS]   = DSC$K_CLASS_D,
;  159          0158  1                           [POINTER] = 0)%;
;  160          0159  1
;  161          0160  1  !
;  162          0161  1  ! Macro to create string descriptor for command parameters and
;  163          0162  1  !       qualifiers
;  164          0163  1  !
;  165          0164  1  MACRO
;  166        M 0165  1      QUALSTR_DESC (STRING) =
;  167        M 0166  1          $BBLOCK [DSC$K_S_BLN] FIELD (DESCR_FIELDS)
;  168        M 0167  1                  PRESET ( [LENGTH]  = (%CHARCOUNT(STRING)),
;  169        M 0168  1                           [DTYPE]   = DSC$K_DTYPE_T,
;  170        M 0169  1                           [CLASS]   = DSC$K_CLASS_S,
;  171          0170  1                           [POINTER] = (UPLIT BYTE (%STRING(STRING))))%;
;  172          0171  1
;  173          0172  1  MACRO
;  174        M 0173  1      CSTRING[] = (UPLIT BYTE (%CHARCOUNT (%STRING (%REMAINING)),
;  175          0174  1                  %STRING (%REMAINING)))% ;
;  176          0175  1
;  177          0176  1  !
;  178          0177  1  ! Macros to check for success or failure from RMS
;  179          0178  1  !
;  180          0179  1  MACRO
;  181          0180  1      RMSBAD (STRING) = (NOT (RMSERR = STRING)) %,
;  182          0181  1      RMSOK (STRING) = (RMSERR = STRING) % ;
;  183          0182  1
;  184          0183  1  MACRO
;  185          0184  1  !
;  186          0185  1  ! Macros to set up and write an FAO string.
```

```
;  187          0186  1 !
;  188          0187  1
;  189        M 0188  1 FAOMAC (FAOMSG)[] =
;  190        M 0189  1         BEGIN
;  191        M 0190  1         FAODSC[DSC$W_LENGTH] = . (FAOMSG)<0,8>;
;  192        M 0191  1         FAODSC[DSC$A_POINTER] = (FAOMSG) + 1;
;  193        M 0192  1         $FAO (FAODSC, RABPTR[RAB$W_RSZ], DISDSC $COMMA (%REMAINING)
;  194        M 0193  1                         %REMAINING);
;  195        M 0194  1         $PUT (RAB = .RABPTR);
;  196          0195  1         END %,
;  197          0196  1
;  198          0197  1     $COMMA[] =, %,
;  199          0198  1
;  200        M 0199  1     OUTPUT_NULL =
;  201        M 0200  1         BEGIN
;  202        M 0201  1         RABPTR[RAB$W_RSZ] = 0;
;  203        M 0202  1         $PUT (RAB = .RABPTR);
;  204          0203  1         END % ;
;  205          0204  1
;  206          0205  1 LITERAL
;  207          0206  1     FALSE                = 0 ,
;  208          0207  1     TRUE                 = 1 ,
;  209          0208  1     BYTE_LENGTH          = 8 ,
;  210          0209  1     WORD_LENGTH          = 16 ,
;  211          0210  1     LONGWORD_LENGTH      = 32 ;
;  212          0211  1
;  213          0212  1 external literal
;  214          0213  1
;  215          0214  1    CLI$_NEGATED,
;  216          0215  1    DISBUFLEN;
;  217          0216  1
;  218          0217  1 !
;  219          0218  1 !       External messages
;  220          0219  1 !
;  221          0220  1 external routine
;  222          0221  1
;  223          0222  1    LIB$SIGNAL;
;  224          0223  1
;  225          0224  1 external literal
;  226          0225  1
;  227          0226  1    UAF$_BADSPC,          UAF$_GRANTERR,          UAF$_GRANTMSG,
;  228          0227  1    UAF$_IDOUTRNG,        UAF$_LSTERR,            UAF$_LSTMSG1,
;  229          0228  1    UAF$_NOGRPWILD,       UAF$_NOIDNAME,          UAF$_NOTIDFMT,
;  230          0229  1    UAF$_NOTUICFMT,       UAF$_RDBADDERR,         UAF$_RDBADDERRU,
;  231          0230  1    UAF$_RDBADDERRV,      UAF$_RDBADDMSG,         UAF$_RDBADDMSGU,
;  232          0231  1    UAF$_RDBCREERR,       UAF$_RDBCREMSG,         UAF$_RDBMDFYERR,
;  233          0232  1    UAF$_RDBMDFYMSG,      UAF$_RDBREMERR,         UAF$_RDBREMMSG,
;  234          0233  1    UAF$_RDBREMMSGU,      UAF$_REVOKEERR,         UAF$_REVOKEMSG,
;  235          0234  1    UAF$_RLSTMSG,         UAF$_SHOWERR,           UAF$_UICERR,
;  236          0235  1    UAF$_WLDNOTALWD;
;  237          0236  1
;  238          0237  1 !
;  239          0238  1 ! External variables
;  240          0239  1 !
;  241          0240  1 EXTERNAL
;  242          0241  1
;  243          0242  1     DISBUF              : VECTOR [,BYTE],       ! Display buffer
```

```
;  244          0243  1      DISDSC                : BLOCK [,BYTE],
;  245          0244  1      FAODSC                : BLOCK [, BYTE],       ! gen'l purpose fao string desc
;  246          0245  1      FOUND_MATCH           : LONG ,
;  247          0246  1      GRP_WILD              : LONG ,
;  248          0247  1      MEM_WILD              : LONG ,
;  249          0248  1      OUTRAB                : $RAB_DECL
;  250          0249  1      RABPTR                : REF BLOCK [, BYTE],   ! RAB for output
;  251          0250  1      RECBUF                : $BBLOCK ,
;  252          0251  1      RIGHTSLIST_MODIFIED   : BYTE ,
;  253          0252  1      RDB_EXISTS            : LONG ,
;  254          0253  1      RMSERR                : LONG ,
;  255          0254  1      STR_WILD              : LONG ,
;  256          0255  1      TOKENDSC              : $BBLOCK ,
;  257          0256  1      TOKENLEN              : WORD ,
;  258          0257  1      TOKENPTR              : LONG ,
;  259          0258  1      UIC_FLAG              : LONG ;
;  260          0259  1
;  261          0260  1
;  262          0261  1  !
;  263          0262  1  ! Global storage
;  264          0263  1  !
;  265          0264  1  GLOBAL
;  266          0265  1      RDB_HEADER_FLAG       : BYTE ,
;  267          0266  1      RDB_LIST_FLAG         : BYTE ,
;  268          0267  1      ATTRIBUTES            : LONG ,
;  269          0268  1      HOLDER                : $BBLOCK[8] INITIAL ( REP 2 OF (0)) ,
;  270          0269  1      IDENT                 : $BBLOCK[4] INITIAL (0) ;
;  271          0270  1
;  272          0271  1  !
;  273          0272  1  ! Module wide OWN storage
;  274          0273  1  !
;  275          0274  1  OWN
;  276          0275  1      SHOW_ID_FULL          : BYTE ,
;  277          0276  1      STATUS                : LONG ,
;  278          0277  1
;  279          0278  1      RLSTNAM               : $NAM (),              ! needed for the DLT option
;  280          0279  1
;  281        P 0280  1      RLSTPRO               : $XABPRO (
;  282        P 0281  1                            PRO = (RWD,RWD,RW)
;  283          0282  1                            ),
;  284          0283  1
;  285        P 0284  1      RL_TFAB               : $FAB (                ! FAB for UAF listing file
;  286        P 0285  1                            FAC = PUT,
;  287        P 0286  1                            RAT = CR,
;  288        P 0287  1                            FNM = 'RIGHTSLIST.LIS',
;  289        P 0288  1                            SHR = NIL,
;  290        P 0289  1                            ORG = SEQ,
;  291        P 0290  1                            RFM = VAR,
;  292        P 0291  1                            MRS = DISBUFLEN,
;  293        P 0292  1                            NAM = RLSTNAM,
;  294        P 0293  1                            XAB = RLSTPRO
;  295          0294  1                            ),
;  296          0295  1
;  297        P 0296  1      RLSTRAB               : $RAB (                ! RAB for RIGHTLIST listing file
;  298        P 0297  1                            RAC = SEQ,
;  299        P 0298  1                            RBF = DISBUF,
;  300        P 0299  1                            FAB = RLSTFAB
```

```
;  301          0300  1                        ) ;
;  302          0301  1
;  303          0302  1
;  304          0303  1 !
;  305          0304  1 ! Parameter and qualifier string descriptors
;  306          0305  1 !
;  307          0306  1 OWN
;  308          0307  1     SD_ATTRIBRESOURCE   : QUALSTR_DESC ('ATTRIBUTES.RESOURCE'),
;  309          0308  1     SD_BRIEF            : QUALSTR_DESC ('BRIEF'),
;  310          0309  1     SD_FULL             : QUALSTR_DESC ('FULL'),
;  311          0310  1     SD_HOLDER           : QUALSTR_DESC ('HOLDER'),
;  312          0311  1     SD_NAME             : QUALSTR_DESC ('NAME'),
;  313          0312  1     SD_SYSTEM_ID        : QUALSTR_DESC ('SYSTEM_ID'),
;  314          0313  1     SD_TOKEN1           : QUALSTR_DESC ('TOKEN1'),
;  315          0314  1     SD_TOKEN2           : QUALSTR_DESC ('TOKEN2'),
;  316          0315  1     SD_USER             : QUALSTR_DESC ('USER'),
;  317          0316  1     SD_VALUE            : QUALSTR_DESC ('VALUE'),
;  318          0317  1     SD_VALUEIDENTIFIER  : QUALSTR_DESC ('VALUE.IDENTIFIER'),
;  319          0318  1     SD_VALUEUIC         : QUALSTR_DESC ('VALUE.UIC');
;  320          0319  1
```

```
;   322     0320  1 %SBTTL ' UAF$ADD_GRP_IDENT - Add rights group identifier '
;   323     0321  1
;   324     0322  1 GLOBAL ROUTINE UAF$ADD_GRP_IDENT : NOVALUE =
;   325     0323  2 BEGIN
;   326     0324  2 !++
;   327     0325  2 !
;   328     0326  2 ! FUNCTIONAL DESCRIPTION:
;   329     0327  2 !
;   330     0328  2 !       Add an identifier for the uic group if it doesn't already
;   331     0329  2 !       exist.  Use the account name for the name of the identifier.
;   332     0330  2 !
;   333     0331  2 ! INPUTS:
;   334     0332  2 !
;   335     0333  2 !       none
;   336     0334  2 !
;   337     0335  2 ! IMPLICIT INPUTS:
;   338     0336  2 !
;   339     0337  2 !       RECBUF must contain a valid UAF record
;   340     0338  2 !
;   341     0339  2 ! OUTPUTS:
;   342     0340  2 !
;   343     0341  2 !       None
;   344     0342  2 !
;   345     0343  2 ! IMPLICIT OUTPUTS:
;   346     0344  2 !
;   347     0345  2 !       none
;   348     0346  2 !
;   349     0347  2 ! ROUTINE VALUE:
;   350     0348  2 !
;   351     0349  2 !       none
;   352     0350  2 !
;   353     0351  2 ! SIDE EFFECTS:
;   354     0352  2 !
;   355     0353  2 !       Rights data base is modified
;   356     0354  2 !--
;   357     0355  2
;   358     0356  2
;   359     0357  2 LOCAL
;   360     0358  2     BLANK       : BYTE INITIAL (%C' ') ,
;   361     0359  2     NAME_DESC   : STATDESC,
;   362     0360  2     NAME_BUFF   : VECTOR [ KGB$S_NAME, BYTE ] ,
;   363     0361  2     RESULT_ID   : $BBLOCK [4];
;   364     0362  2
;   365     0363  2 RESULT_ID = 0 ;
;   366     0364  2 IDENT = 0 ;
;   367     0365  2
;   368     0366  2 !
;   369     0367  2 ! Get the identifier name from the UAF record account field.
;   370     0368  2 !
;   371     0369  2 CH$COPY ( UAF$S_ACCOUNT, RECBUF[UAF$T_ACCOUNT], %C' ',
;   372     0370  2           KGB$S_NAME, NAME_BUFF );
;   373     0371  2 NAME_DESC[LENGTH]  = MIN ( UAF$S_ACCOUNT,
;   374     0372  2                           KGB$S_NAME,
;   375     0373  2                           (CH$FIND_CH (KGB$S_NAME, NAME_BUFF, %C' ') - NAME_BUFF) ) ;
;   376     0374  2 NAME_DESC[POINTER] = NAME_BUFF ;
;   377     0375  2
;   378     0376  2 !
```

```
;  379          0377  2 ! Build the identifier from the UAF record and then wild card
;  380          0378  2 ! the member number
;  381          0379  2 !
;  382          0380  2 UAF$FIND_UIC () ;
;  383          0381  2 IDENT [UIC$V_MEMBER] = UIC$K_WILD_MEMBER ;
;  384          0382  2
;  385          0383  2 !
;  386          0384  2 ! If the account field is blank then do not try to create
;  387          0385  2 ! an identifier.
;  388          0386  2 !
;  389          0387  2 IF CH$NEQ ( 1, BLANK, .NAME_DESC[LENGTH], .NAME_DESC[POINTER], %C' ')
;  390          0388  2     THEN
;  391          0389  2         !
;  392          0390  2         ! The account is non blank so check if the ID name or
;  393          0391  2         ! value is already in use.
;  394          0392  2         !
;  395          0393  3         IF NOT ( $ASCTOID ( NAME = NAME_DESC ) OR
;  396          0394  3                  $IDTOASC ( ID   = .IDENT ))
;  397          0395  2             THEN
;  398          0396  3                 BEGIN
;  399          0397  3                 !
;  400          0398  3                 ! The identifier is not in use.
;  401          0399  3                 ! Perform the actual service to add the identifier
;  402          0400  3                 !
;  403        P 0401  3                 STATUS = $ADD_IDENT ( NAME   = NAME_DESC,
;  404        P 0402  3                                       ID     = .IDENT,
;  405        P 0403  3                                       ATTRIB = .ATTRIBUTES,
;  406          0404  3                                       RESID  = RESULT_ID ) ;
;  407          0405  3
;  408          0406  3                 IF NOT .STATUS
;  409          0407  3                     THEN
;  410          0408  4                         BEGIN
;  411          0409  4                         IF .IDENT[UIC$V_FORMAT] EQL UIC$K_UIC_FORMAT
;  412          0410  4                             THEN LIB$SIGNAL(UAF$_RDBADDERRU, 3, NAME_DESC,
;  413          0411  4                                 .IDENT[UIC$V_GROUP], .IDENT[UIC$V_MEMBER],
;  414          0412  4                                 .STATUS)
;  415          0413  4                             ELSE LIB$SIGNAL(UAF$_RDBADDERRV, 2, NAME_DESC,
;  416          0414  4                                 .IDENT, .STATUS);
;  417          0415  4                         END
;  418          0416  3                     ELSE
;  419          0417  4                         BEGIN
;  420          0418  4                         RIGHTSLIST_MODIFIED = TRUE ;
;  421          0419  4                         IF .RESULT_ID[UIC$V_FORMAT] EQL UIC$K_UIC_FORMAT
;  422          0420  4                             THEN LIB$SIGNAL(UAF$_RDBADDMSGU, 3, NAME_DESC,
;  423          0421  4                                             .RESULT_ID[UIC$V_GROUP],
;  424          0422  4                                             .RESULT_ID[UIC$V_MEMBER])
;  425          0423  4                             ELSE LIB$SIGNAL(UAF$_RDBADDMSG, 2, NAME_DESC,
;  426          0424  4                                             .RESULT_ID);
;  427          0425  3                         END ;
;  428          0426  2 END ;
;  429          0427  1 END ;                                   ! End of UAF$ADD_GRP_IDENT

                                                        .TITLE  RIGHTSMAN
                                                        .IDENT  \V04-000\

                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2
```

```
                53 49 4C 2E 54 53 49 4C 53 54 48 47 49 52  00000 P.AAA:  .ASCII  \RIGHTSLIST.LIS\
4F 53 45 52 2E 53 45 54 55 42 49 52 54 54 41  0000E P.AAB:  .ASCII  \ATTRIBUTES.RESOURCE\
                                        45 43 52 55  0001D
                                     46 45 49 52 42  00021 P.AAC:  .ASCII  \BRIEF\
                                        4C 4C 55 46  00026 P.AAD:  .ASCII  \FULL\
                                  52 45 44 4C 4F 48  0002A P.AAE:  .ASCII  \HOLDER\
                                        45 4D 41 4E  00030 P.AAF:  .ASCII  \NAME\
                           44 49 5F 4D 45 54 53 59 53  00034 P.AAG:  .ASCII  \SYSTEM_ID\
                                  31 4E 45 4B 4F 54  0003D P.AAH:  .ASCII  \TOKEN1\
                                  32 4E 45 4B 4F 54  00043 P.AAI:  .ASCII  \TOKEN2\
                                        52 45 53 55  00049 P.AAJ:  .ASCII  \USER\
                                     45 55 4C 41 56  0004D P.AAK:  .ASCII  \VALUE\
45 49 46 49 54 4E 45 44 49 2E 45 55 4C 41 56  00052 P.AAL:  .ASCII  \VALUE.IDENTIFIER\
                                              52  00061
                           43 49 55 2E 45 55 4C 41 56  00062 P.AAM:  .ASCII  \VALUE.UIC\

                                                       .PSECT  $OWN$,NOEXE,2

                                                 00000 SHOW_ID_FULL:
                                                       .BLKB   1
                                                 00001       .BLKB   3
                                                 00004 STATUS: .BLKB   4
                                              02 00008 RLSTNAM:.BYTE   2
                                              60 00009       .BYTE   96
                                              00 0000A       .BYTE   0
                                              00 0000B       .BYTE   0
                                      00000000   0000C       .LONG   0
                                              00 00010       .BYTE   0
                                              00 00011       .BYTE   0
                                              00 00012       .BYTE   0
                                              00 00013       .BYTE   0
                                      00000000   00014       .LONG   0
                                      00000000   00018       .LONG   0
                                          0000#  0001C       .WORD   0[8]
                                          0000#  0002C       .WORD   0[3]
                                          0000#  00032       .WORD   0[3]
                                      00000000   00038       .LONG   0
                                      00000000   0003C       .LONG   0
                                              00 00040       .BYTE   0
                                              00 00041       .BYTE   0
                                              00 00042       .BYTE   0
                                              00 00043       .BYTE   0
                                              00 00044       .BYTE   0
                                              00 00045       .BYTE   0
                                              00# 00046      .BYTE   0[2]
                                      00000000   00048       .LONG   0
                                      00000000   0004C       .LONG   0
                                      00000000   00050       .LONG   0
                                      00000000   00054       .LONG   0
                                      00000000   00058       .LONG   0
                                      00000000   0005C       .LONG   0
                                      00000000#  00060       .LONG   0[2]
                                              13 00068 RLSTPRO:.BYTE   19
                                              58 00069       .BYTE   88
                                            0000 0006A       .WORD   0
                                      00000000   0006C       .LONG   0
```

```
                     FC44  00070          .WORD   -956
                       00  00072          .BYTE   0
                       00  00073          .BYTE   0
                0000   0000  00074          .WORD   0, 0
                       00  00078          .BYTE   0
                       00  00079          .BYTE   0
                     0000  0007A          .WORD   0
                 00000000  0007C          .LONG   0
                 00000000  00080          .LONG   0
                     0000  00084          .WORD   0
                     0000  00086          .WORD   0
                 00000000  00088          .LONG   0
                 00000000  0008C          .LONG   0
                           00090          .BLKB   48
                       03  000C0 RLSTFAB: .BYTE   3
                       50  000C1          .BYTE   80
                     0000  000C2          .WORD   0
                 00000000  000C4          .LONG   0
                 00000000  000C8          .LONG   0
                 00000000  000CC          .LONG   0
                 00000000  000D0          .LONG   0
                     0000  000D4          .WORD   0
                       01  000D6          .BYTE   1
                       20  000D7          .BYTE   32
                 00000000  000D8          .LONG   0
                       00  000DC          .BYTE   0
                       00  000DD          .BYTE   0
                       02  000DE          .BYTE   2
                       02  000DF          .BYTE   2
                 00000000  000E0          .LONG   0
                 00000000' 000E4          .ADDRESS RLSTPRO
                 00000000' 000E8          .ADDRESS RLSTNAM
                 00000000' 000EC          .ADDRESS P.AAA
                 00000000  000F0          .LONG   0
                       0E  000F4          .BYTE   14
                       00  000F5          .BYTE   0
                    0000G  000F6          .WORD   DISBUFLEN
                 00000000  000F8          .LONG   0
                     0000  000FC          .WORD   0
                       00  000FE          .BYTE   0
                       00  000FF          .BYTE   0
                 00000000  00100          .LONG   0
                 00000000  00104          .LONG   0
                     0000  00108          .WORD   0
                       00  0010A          .BYTE   0
                       00  0010B          .BYTE   0
                 00000000  0010C          .LONG   0
                       01  00110 RLSTRAB: .BYTE   1
                       44  00111          .BYTE   68
                     0000  00112          .WORD   0
                 00000000  00114          .LONG   0
                 00000000  00118          .LONG   0
                 00000000  0011C          .LONG   0
                    0000#  00120          .WORD   0[3]
                     0000  00126          .WORD   0
                 00000000  00128          .LONG   0
                     0000  0012C          .WORD   0
```

```
                00  0012E          .BYTE   0
                00  0012F          .BYTE   0
              0000  00130          .WORD   0
              0000  00132          .WORD   0
          00000000  00134          .LONG   0
          00000000G 00138          .ADDRESS DISBUF
          00000000  0013C          .LONG   0
          00000000  00140          .LONG   0
                00  00144          .BYTE   0
                00  00145          .BYTE   0
                00  00146          .BYTE   0
                00  00147          .BYTE   0
          00000000  00148          .LONG   0
          00000000' 0014C          .ADDRESS RLSTFAB
          00000000  00150          .LONG   0
              0013  00154 SD_ATTRIBRESOURCE:
                                   .WORD   19
            01  0E  00156          .BYTE   14, 1
          00000000' 00158          .ADDRESS P.AAB
              0005  0015C SD_BRIEF:
                                   .WORD   5
            01  0E  0015E          .BYTE   14, 1
          00000000' 00160          .ADDRESS P.AAC
              0004  00164 SD_FULL:.WORD   4
            01  0E  00166          .BYTE   14, 1
          00000000' 00168          .ADDRESS P.AAD
              0006  0016C SD_HOLDER:
                                   .WORD   6
            01  0E  0016E          .BYTE   14, 1
          00000000' 00170          .ADDRESS P.AAE
              0004  00174 SD_NAME:.WORD   4
            01  0E  00176          .BYTE   14, 1
          00000000' 00178          .ADDRESS P.AAF
              0009  0017C SD_SYSTEM_ID:
                                   .WORD   9
            01  0E  0017E          .BYTE   14, 1
          00000000' 00180          .ADDRESS P.AAG
              0006  00184 SD_TOKEN1:
                                   .WORD   6
            01  0E  00186          .BYTE   14, 1
          00000000' 00188          .ADDRESS P.AAH
              0006  0018C SD_TOKEN2:
                                   .WORD   6
            01  0E  0018E          .BYTE   14, 1
          00000000' 00190          .ADDRESS P.AAI
              0004  00194 SD_USER:.WORD   4
            01  0E  00196          .BYTE   14, 1
          00000000' 00198          .ADDRESS P.AAJ
              0005  0019C SD_VALUE:
                                   .WORD   5
            01  0E  0019E          .BYTE   14, 1
          00000000' 001A0          .ADDRESS P.AAK
              0010  001A4 SD_VALUEIDENTIFIER:
                                   .WORD   16
            01  0E  001A6          .BYTE   14, 1
          00000000' 001A8          .ADDRESS P.AAL
              0009  001AC SD_VALUEUIC:
```

```
                                                        .WORD   9
                                              01  0E    001AE  .BYTE   14, 1
                                          00000000'     001B0  .ADDRESS P.AAM

                                                        .PSECT  $GLOBAL$,NOEXE.2

                                                00000 RDB_HEADER_FLAG::
                                                        .BLKB   1
                                                00001 RDB_LIST_FLAG::
                                                        .BLKB   1
                                                00002   .BLKB   2
                                                00004 ATTRIBUTES::
                                                        .BLKB   4
                                          00000000#     00008 HOLDER::.LONG   0[2]
                                          00000000      00010 IDENT:: .LONG   0

                                                        .EXTRN  CLI$PRESENT, CLI$GET_VALUE
                                                        .EXTRN  FAOOUT, FMT_SYS_MSG
                                                        .EXTRN  GETVAL, PARSE_UIC
                                                        .EXTRN  PARSE_WILD, PARSE_WILD_UIC
                                                        .EXTRN  WILD_USER, CLI$_NEGATED
                                                        .EXTRN  DISBUFLEN, LIB$SIGNAL
                                                        .EXTRN  UAF$_BADSPC, UAF$_GRANTERR
                                                        .EXTRN  UAF$_GRANTMSG, UAF$_IDOUTRNG
                                                        .EXTRN  UAF$_LSTERR, UAF$_LSTMSG1
                                                        .EXTRN  UAF$_NOGRPWILD, UAF$_NOIDNAME
                                                        .EXTRN  UAF$_NOTIDFMT, UAF$_NOTUICFMT
                                                        .EXTRN  UAF$_RDBADDERR, UAF$_RDBADDERRU
                                                        .EXTRN  UAF$_RDBADDERRV
                                                        .EXTRN  UAF$_RDBADDMSG, UAF$_RDBADDMSGU
                                                        .EXTRN  UAF$_RDBCREERR, UAF$_RDBCREMSG
                                                        .EXTRN  UAF$_RDBMDFYERR
                                                        .EXTRN  UAF$_RDBMDFYMSG
                                                        .EXTRN  UAF$_RDBREMERR, UAF$_RDBREMMSG
                                                        .EXTRN  UAF$_RDBREMMSGU
                                                        .EXTRN  UAF$_REVOKEERR, UAF$_REVOKEMSG
                                                        .EXTRN  UAF$_RLSTMSG, UAF$_SHOWERR
                                                        .EXTRN  UAF$_UICERR, UAF$_WLDNOTALWD
                                                        .EXTRN  DISBUF, DISDSC, FAODSC
                                                        .EXTRN  FOUND_MATCH, GRP_WILD
                                                        .EXTRN  MEM_WILD, OUTRAB
                                                        .EXTRN  RABPTR, RECBUF, RIGHTSLIST_MODIFIED
                                                        .EXTRN  RDB_EXISTS, RMSERR
                                                        .EXTRN  STR_WILD, TOKENDSC
                                                        .EXTRN  TOKENLEN, TOKENPTR
                                                        .EXTRN  UIC_FLAG, SYS$ASCTOID
                                                        .EXTRN  SYS$IDTOASC, SYS$ADD_IDENT

                                                        .PSECT  $CODE$,NOWRT,2

                                                  00FC  00000   .ENTRY  UAF$ADD_GRP_IDENT, Save R2,R3,R4,R5,R6,R7  ; 0322
                                      57 00000000G  00    9E  00002 MOVAB  LIB$SIGNAL, R7                          ;
                                      56 00000000'  00    9E  00009 MOVAB  IDENT, R6                               ;
                                      5E            30    C2  00010 SUBL2  #48, SP                                 ;
                                      6E            20    90  00013 MOVB   #32, BLANK                              ; 0323
                                28    AE 010E0000   8F    D0  00016 MOVL   #17694720, NAME_DESC                    ; 0359
                                         2C         AE    D4  0001E CLRL   NAME_DESC+4                             ;
```

```
                                                04   AE  D4 00021          CLRL    RESULT_ID                                       ; 0363
                                                66   D4  00024          CLRL    IDENT                                           ; 0364
                    08   AE 00000000G   00          20  28 00026          MOVC3   #32, RECBUF+52, NAME_BUFF                       ; 0369
                    08   AE             20          20  3A 0002F          LOCC    #32, #32, NAME_BUFF                             ; 0373
                                                     02  12 00034          BNEQ    1$
                                                     51  D4 00036          CLRL    R1
                                        50       08  AE  9E 00038 1$:      MOVAB   NAME_BUFF, R0
                                        51           50  C2 0003C          SUBL2   R0, R1
                                        20           51  D1 0003F          CMPL    R1, #32                                         ; 0371
                                                     03  15 00042          BLEQ    2$
                                        51           20  D0 00044          MOVL    #32, R1
                                  28    AE           51  B0 00047 2$:      MOVW    R1, NAME_DESC
                                  2C    AE       08  AE  9E 0004B          MOVAB   NAME_BUFF, NAME_DESC+4                          ; 0374
                         00000000V      00           00  FB 00050          CALLS   #0, @UAF$FIND_UIC                               ; 0380
                                        66           01  AE 00057          MNEGW   #1, IDENT                                       ; 0381
    28   AE             20             6E           01  2D 0005A          CMPC5   #1, BLANK, #32, NAME_DESC, @NAME_DESC+4         ; 0387
                                                 2C  BE      00060
                                                     01  12 00062          BNEQ    3$
                                                         04 00064          RET
                                        7E           7C  00065 3$:         CLRQ    -(SP)                                           ; 0393
                                                 30  AE  9F 00067          PUSHAB  NAME_DESC
                         00000000G      00           03  FB 0006A          CALLS   #3, SYS$ASCTOID
                                        0F           50  E8 00071          BLBS    R0, 4$
                                                     7E  7C 00074          CLRQ    -(SP)                                           ; 0394
                                                     7E  7C 00076          CLRQ    -(SP)
                                                     7E  D4 00078          CLRL    -(SP)
                                                     66  DD 0007A          PUSHL   IDENT
                         00000000G      00           06  FB 0007C          CALLS   #6, SYS$IDTOASC
                                        01           50  E9 00083 4$:      BLBC    R0, 5$
                                                         04 00086          RET
                                                 04  AE  9F 00087 5$:      PUSHAB  RESULT_ID                                       ; 0404
                                                 F4  A6  DD 0008A          PUSHL   ATTRIBUTES
                                                     66  DD 0008D          PUSHL   IDENT
                                                 34  AE  9F 0008F          PUSHAB  NAME_DESC
                         00000000G      00           04  FB 00092          CALLS   #4, SYS$ADD_IDENT
                         00000000'      00           50  D0 00099          MOVL    R0, STATUS
                                        32           50  E8 000A0          BLBS    R0, 7$                                          ; 0406
                               C0       8F       03  A6  93 000A3          BITB    IDENT+3, #192                                   ; 0409
                                                     1A  12 000A8          BNEQ    6$
                                                     50  DD 000AA          PUSHL   R0                                              ; 0412
                                        7E           66  3C 000AC          MOVZWL  IDENT, -(SP)                                    ; 0411
    7E        02   A6                  0E           00  EF 000AF          EXTZV   #0, #14, IDENT+2, -(SP)
                                                 34  AE  9F 000B5          PUSHAB  NAME_DESC                                       ; 0410
                                                     03  DD 000B8          PUSHL   #3
                                    00000000G    8F       DD 000BA          PUSHL   #UAF$_RDBADDERRU
                                        67           06  FB 000C0          CALLS   #6, LIB$SIGNAL
                                                         04 000C3          RET
                                                     50  DD 000C4 6$:      PUSHL   R0                                              ; 0414
                                                     66  DD 000C6          PUSHL   IDENT
                                                 30  AE  9F 000C8          PUSHAB  NAME_DESC                                       ; 0413
                                                     02  DD 000CB          PUSHL   #2
                                    00000000G    8F       DD 000CD          PUSHL   #UAF$_RDBADDERRV
                                                     23  11 000D3          BRB     8$
                         00000000G      00           01  90 000D5 7$:      MOVB    #1, RIGHTSLIST_MODIFIED                         ; 0418
                               C0       8F       07  AE  93 000DC          BITB    RESULT_ID+3, #T92                               ; 0419
                                                     19  12 000E1          BNEQ    9$
                                        7E       04  AE  3C 000E3          MOVZWL  RESULT_ID, -(SP)                                ; 0422
```

```
            7E      0A   AE            0E                00  EF 000E7          EXTZV   #0, #14, RESULT_ID+2, -(SP)          ; 0421
                                                    30    AE  9F 000ED          PUSHAB  NAME_DESC                            ; 0420
                                                          03  DD 000F0          PUSHL   #3
                                                00000000G 8F  DD 000F2          PUSHL   #UAF$_RDBADDMSGU
                                                    67        05  FB 000F8 8$:  CALLS   #5, LIB$SIGNAL
                                                              04  000FB        RET
                                                    04    AE  DD 000FC 9$:    PUSHL   RESULT_ID                            ; 0424
                                                    2C    AE  9F 000FF        PUSHAB  NAME_DESC                            ; 0423
                                                          02  DD 00102        PUSHL   #2
                                                00000000G 8F  DD 00104        PUSHL   #UAF$_RDBADDMSG
                                                    67        04  FB 0010A    CALLS   #4, LIB$SIGNAL
                                                              04  0010D        RET                                          ; 0427

; Routine Size:  270 bytes,    Routine Base:  $CODE$ + 0000

;  430          0428  1
```

```
;  432     0429  1 %SBTTL ' UAF$ADD_IDENT  -  Add rights identifier '
;  433     0430  1
;  434     0431  1 GLOBAL ROUTINE UAF$ADD_IDENT : NOVALUE =
;  435     0432  2 BEGIN
;  436     0433  2 !++
;  437     0434  2 !
;  438     0435  2 ! FUNCTIONAL DESCRIPTION:
;  439     0436  2 !
;  440     0437  2 !       This routine will add an identifier to the rights data base.
;  441     0438  2 !       It is invoked via the ADD/IDENT command.
;  442     0439  2 !
;  443     0440  2 ! INPUTS:
;  444     0441  2 !
;  445     0442  2 !       none
;  446     0443  2 !
;  447     0444  2 ! IMPLICIT INPUTS:
;  448     0445  2 !
;  449     0446  2 !       Command line
;  450     0447  2 !
;  451     0448  2 ! OUTPUTS:
;  452     0449  2 !
;  453     0450  2 !       None
;  454     0451  2 !
;  455     0452  2 ! IMPLICIT OUTPUTS:
;  456     0453  2 !
;  457     0454  2 !       none
;  458     0455  2 !
;  459     0456  2 ! ROUTINE VALUE:
;  460     0457  2 !
;  461     0458  2 !       none
;  462     0459  2 !
;  463     0460  2 ! SIDE EFFECTS:
;  464     0461  2 !
;  465     0462  2 !       Rights data base is modified
;  466     0463  2 !
;  467     0464  2 !--
;  468     0465  2
;  469     0466  2
;  470     0467  2 LOCAL
;  471     0468  2     NAME_DESC    : STATDESC,
;  472     0469  2     NAME_BUFF    : VECTOR [ KGB$S_NAME, BYTE ] ,
;  473     0470  2     RESULT_ID    : $BBLOCK [4];
;  474     0471  2
;  475     0472  2 RESULT_ID = 0 ;
;  476     0473  2 IDENT = 0 ;
;  477     0474  2
;  478     0475  2 !
;  479     0476  2 ! Set the resource attribute if /RESOURCE was specified.
;  480     0477  2 ! This is done first since it is required whether or not
;  481     0478  2 ! P1 or /USER is specified.
;  482     0479  2 !
;  483     0480  2 IF CLI$PRESENT (SD_ATTRIBRESOURCE)
;  484     0481  2     THEN ATTRIBUTES = KGB$M_RESOURCE
;  485     0482  2     ELSE ATTRIBUTES = 0 ;
;  486     0483  2
;  487     0484  2 !
;  488     0485  2 ! We will now get the id name parameter from
```

```
;  489     0486  2 ! the command line.  If the parameter is missing
;  490     0487  2 ! then the /USER qualifier must be present
;  491     0488  2 !
;  492     0489  2 IF CLI$PRESENT ( SD_TOKEN1 )
;  493     0490  2     THEN
;  494     0491  3         BEGIN
;  495     0492  3         CLI$GET_VALUE ( SD_TOKEN1, TOKENDSC ) ;
;  496     0493  3
;  497     0494  3         CH$COPY ( .TOKENLEN, .TOKENPTR, %C' ', KGB$S_NAME, NAME_BUFF );
;  498     0495  3         NAME_DESC[LENGTH]  = MIN ( .TOKENLEN , KGB$S_NAME ) ;
;  499     0496  3         NAME_DESC[POINTER] = NAME_BUFF ;
;  500     0497  3
;  501     0498  3         !
;  502     0499  3         ! If the /VALUE qualifier was specified then get it's value
;  503     0500  3         !
;  504     0501  3         IF NOT CLI$PRESENT ( SD_VALUE )
;  505     0502  3             THEN
;  506     0503  4                 BEGIN
;  507     0504  4                 IDENT [UIC$V_FORMAT]  = 0 ;
;  508     0505  4                 IDENT [UIC$V_ID_CODE] = 0 ;
;  509     0506  4                 END
;  510     0507  3             ELSE
;  511     0508  4                 BEGIN
;  512     0509  4                 !
;  513     0510  4                 ! /VALUE is there now determine what keyword was used
;  514     0511  4                 ! and interpret the data as appropriate.
;  515     0512  4                 ! First check for IDENTIFIER:
;  516     0513  4                 IF CLI$GET_VALUE ( SD_VALUEIDENTIFIER, TOKENDSC )
;  517     0514  4                     THEN
;  518     0515  5                         BEGIN
;  519     0516  5                         LOCAL
;  520     0517  5                             TMP_IDENT : LONG INITIAL (0) ;
;  521     0518  5
;  522     0519  5                         STATUS = GETVAL ( TMP_IDENT, LONGWORD_LENGTH );
;  523     0520  5                         IF NOT .STATUS
;  524     0521  5                             THEN RETURN ;
;  525     0522  5                         IDENT [UIC$V_FORMAT]  = UIC$K_ID_FORMAT ;
;  526     0523  5                         IDENT [UIC$V_ID_CODE] = .TMP_IDENT ;
;  527     0524  5                         IF (.IDENT LSSU %X'80008000') OR
;  528     0525  6                            (.IDENT GTRU UIC$K_LAST_ID)
;  529     0526  5                             THEN
;  530     0527  6                                 BEGIN
;  531     0528  6                                 LIB$SIGNAL(UAF$_IDOUTRNG);
;  532     0529  6                                 RETURN ;
;  533     0530  5                                 END ;
;  534     0531  5                         END
;  535     0532  4                 ELSE
;  536     0533  5                     BEGIN
;  537     0534  5                     !
;  538     0535  5                     ! Now check for UIC
;  539     0536  5                     ! These two keywords are synonyms
;  540     0537  5                     !
;  541     0538  5                     IF CLI$PRESENT (SD_VALUEUIC)
;  542     0539  5                         THEN IF NOT PARSE_WILD (SD_VALUEUIC, TRUE )
;  543     0540  5                             THEN RETURN ;
;  544     0541  5                     IF .UIC_FLAG
;  545     0542  5                         THEN
```

```
;  546     0543  6                                BEGIN
;  547     0544  6                                IF .GRP_WILD and not .MEM_WILD
;  548     0545  6                                    THEN
;  549     0546  7                                        BEGIN
;  550     0547  7                                        !
;  551     0548  7                                        ! Wild cards are not allowed for group numbers
;  552     0549  7                                        !
;  553     0550  7                                        LIB$SIGNAL(UAF$_NOGRPWILD);
;  554     0551  7                                        RETURN ;
;  555     0552  6                                        END ;
;  556     0553  6                                !
;  557     0554  6                                ! The UIC format was used,  parse it to fine the
;  558     0555  6                                ! group and member number.
;  559     0556  6                                ! Wild card group and member numbers are allowed
;  560     0557  6                                !
;  561     0558  6                                IF NOT PARSE_UIC ( IDENT[UIC$V_GROUP],
;  562     0559  6                                                   IDENT[UIC$V_MEMBER],
;  563     0560  6                                                   TRUE )
;  564     0561  6                                    THEN
;  565     0562  7                                        BEGIN
;  566     0563  7                                        LIB$SIGNAL(UAF$_UICERR, 2, .TOKENLEN,
;  567     0564  7                                                                .TOKENPTR);
;  568     0565  7                                        RETURN ;
;  569     0566  6                                        END ;
;  570     0567  6                                IDENT[UIC$V_FORMAT] = UIC$K_UIC_FORMAT ;
;  571     0568  6                                END
;  572     0569  5                            ELSE
;  573     0570  6                                BEGIN
;  574     0571  6                                !
;  575     0572  6                                ! A user name was specified.  Get the UAF record
;  576     0573  6                                ! for this user and pull the UIC out.
;  577     0574  6                                !
;  578     0575  6                                IF .STR_WILD
;  579     0576  6                                    THEN
;  580     0577  7                                        BEGIN
;  581     0578  7                                        !
;  582     0579  7                                        ! Wild cards are not allowed for user specs
;  583     0580  7                                        !
;  584     0581  7                                        LIB$SIGNAL(UAF$_WLDNOTALWD);
;  585     0582  7                                        RETURN ;
;  586     0583  6                                        END ;
;  587     0584  6                                FOUND_MATCH = FALSE ;
;  588     0585  7                                IF RMSBAD ( WILD_USER (UAF$FIND_UIC))
;  589     0586  6                                    THEN
;  590     0587  7                                        BEGIN
;  591     0588  7                                        IF .RMSERR EQL RMS$_RNF
;  592     0589  7                                            THEN LIB$SIGNAL(UAF$_BADSPC)
;  593     0590  7                                            ELSE LIB$SIGNAL(UAF$_RDBADDERR, 1,
;  594     0591  7                                                            NAME_DESC, .RMSERR);
;  595     0592  7                                        RETURN ;
;  596     0593  6                                        END ;
;  597     0594  5                                END ;
;  598     0595  5
;  599     0596  4                            END;
;  600     0597  3                    END ;
;  601     0598  3
;  602     0599  3            !
```

```
;  603          0600  3          ! Perform the actual service to add the identifier
;  604          0601  3          !
;  605        P 0602  3          STATUS = $ADD_IDENT ( NAME    = NAME_DESC,
;  606        P 0603  3                                ID      = .IDENT,
;  607        P 0604  3                                ATTRIB  = .ATTRIBUTES,
;  608          0605  3                                RESID   = RESULT_ID ) ;
;  609          0606  3
;  610          0607  3          IF NOT .STATUS
;  611          0608  3              THEN
;  612          0609  4                  BEGIN
;  613          0610  4                  IF .IDENT[UIC$V_FORMAT] EQL UIC$K_UIC_FORMAT
;  614          0611  4                      THEN LIB$SIGNAL(UAF$_RDBADDERRU, 3, NAME_DESC,
;  615          0612  4                                      .IDENT[UIC$V_GROUP],
;  616          0613  4                                      .IDENT[UIC$V_MEMBER], .STATUS)
;  617          0614  4                      ELSE LIB$SIGNAL(UAF$_RDBADDERRV, 2, NAME_DESC, .IDENT
;  618          0615  4                                      .STATUS);
;  619          0616  4                  END
;  620          0617  3              ELSE
;  621          0618  4                  BEGIN
;  622          0619  4                  RIGHTSLIST_MODIFIED = TRUE ;
;  623          0620  4                  IF .RESULT_ID[UIC$V_FORMAT] EQL UIC$K_UIC_FORMAT
;  624          0621  4                      THEN LIB$SIGNAL(UAF$_RDBADDMSGU, 3, NAME_DESC,
;  625          0622  4                                      .RESULT_ID[UIC$V_GROUP],
;  626          0623  4                                      .RESULT_ID[UIC$V_MEMBER] )
;  627          0624  4                      ELSE LIB$SIGNAL(UAF$_RDBADDMSG, 2, NAME_DESC, .RESULT_ID);
;  628          0625  3                  END ;
;  629          0626  3              END
;  630          0627  2          ELSE
;  631          0628  3              BEGIN
;  632          0629  3              IF CLI$PRESENT (SD_USER)
;  633          0630  3                  THEN
;  634          0631  4                      BEGIN
;  635          0632  4                      IF NOT PARSE_WILD (SD_USER, TRUE )
;  636          0633  4                          THEN RETURN ;
;  637          0634  4                      END
;  638          0635  3                  ELSE
;  639          0636  4                      BEGIN
;  640          0637  4                      LIB$SIGNAL(UAF$_NOIDNAME);
;  641          0638  4                      RETURN ;
;  642          0639  3                      END ;
;  643          0640  3              FOUND_MATCH = FALSE ;
;  644          0641  4              IF RMSBAD ( WILD_USER (UAF$ADD_IDENT_RECBUF))
;  645          0642  3                  THEN
;  646          0643  4                      BEGIN
;  647          0644  4                      IF .RMSERR EQL RMS$_RNF
;  648          0645  4                          THEN LIB$SIGNAL(UAF$_BADSPC)
;  649          0646  4                          ELSE LIB$SIGNAL(UAF$_RDBADDERR, 1, NAME_DESC, .RMSERR);
;  650          0647  4                      RETURN ;
;  651          0648  3                      END ;
;  652          0649  2              END ;
;  653          0650  2
;  654          0651  2 RETURN
;  655          0652  2
;  656          0653  1 END ;                                   ! End of UAF$ADD_IDENT
```

```
                                                0FFC 00000          .ENTRY  UAF$ADD_IDENT, Save R2,R3,R4,R5,R6,R7,R8,-      ; 0431
                                                                            R9,R10,R11
                                          5B 00000000G   00  9E 00002       MOVAB   RMSERR, R11
                                          5A 00000000G   00  9E 00009       MOVAB   LIB$SIGNAL, R10
                                          59 00000000G   00  9E 00010       MOVAB   CLI$PRESENT, R9
                                          58 00000000'   00  9E 00017       MOVAB   STATUS, R8
                                          57 00000000'   00  9E 0001E       MOVAB   IDENT, R7
                                          5E                 30  C2 00025       SUBL2   #48, SP
                                    28    AE 010E0000      8F  D0 00028       MOVL    #17694720, NAME_DESC                    ; 0468
                                                     2C    AE  D4 00030       CLRL    NAME_DESC+4
                                                     04    AE  D4 00033       CLRL    RESULT_ID                               ; 0472
                                                           67  D4 00036       CLRL    IDENT                                   ; 0473
                                                   0150    C8  9F 00038       PUSHAB  SD_ATTRIBRESOURCE                       ; 0480
                                          69                 01  FB 0003C       CALLS   #1, CLI$PRESENT
                                          06                 50  E9 0003F       BLBC    R0, 1$
                                    F4    A7                 01  D0 00042       MOVL    #1, ATTRIBUTES                          ; 0481
                                                             03  11 00046       BRB     2$
                                                     F4    A7  D4 00048 1$:     CLRL    ATTRIBUTES                              ; 0482
                                                   0180    C8  9F 0004B 2$:     PUSHAB  SD_TOKEN1                               ; 0489
                                          69                 01  FB 0004F       CALLS   #1, CLI$PRESENT
                                          03                 50  E8 00052       BLBS    R0, 3$
                                                         01B4    31  00055       BRW     21$
                                             000000000G    00  9F 00058 3$:     PUSHAB  TOKENDSC                                ; 0492
                                                   0180    C8  9F 0005E       PUSHAB  SD_TOKEN1
                                    00000000G      00          02  FB 00062       CALLS   #2, CLI$GET_VALUE
                                          56 00000000G   00  3C 00069       MOVZWL  TOKENLEN, R6                            ; 0494
                                          50 00000000G   00  D0 00070       MOVL    TOKENPTR, R0
20                  20                    60                 56  2C 00077       MOVC5   R6, (R0), #32, #32, NAME_BUFF
                                                     08    AE      0007C
                                          50                 56  D0 0007E       MOVL    R6, R0                                  ; 0495
                                          20                 50  B1 00081       CMPW    R0, #32
                                                             03  1B 00084       BLEQU   4$
                                          50                 20  D0 00086       MOVL    #32, R0
                                    28    AE                 50  B0 00089 4$:     MOVW    R0, NAME_DESC
                                    2C    AE           08    AE  9E 0008D       MOVAB   NAME_BUFF, NAME_DESC+4                  ; 0496
                                                     0198    C8  9F 00092       PUSHAB  SD_VALUE                                ; 0501
                                          69                 01  FB 00096       CALLS   #1, CLI$PRESENT
                                          0A                 50  E8 00099       BLBS    R0, 6$
                                          67 CFFFFFFF        8F  CA 0009C       BICL2   #-805306369, IDENT                      ; 0504
                                                         00E5    31  000A3 5$:     BRW     16$                                     ; 0501
                                             000000000G    00  9F 000A6 6$:     PUSHAB  TOKENDSC                                ; 0513
                                                   01A0    C8  9F 000AC       PUSHAB  SD_VALUEIDENTIFIER
                                    00000000G      00          02  FB 000B0       CALLS   #2, CLI$GET_VALUE
                                          3D                 50  E9 000B7       BLBC    R0, 9$
                                                             6E  D4 000BA       CLRL    TMP_IDENT                               ; 0515
                                                             20  DD 000BC       PUSHL   #32                                     ; 0519
                                                     04    AE  9F 000BE       PUSHAB  TMP_IDENT
                                    00000000G      00          02  FB 000C1       CALLS   #2, GETVAL
                                          68                 50  D0 000C8       MOVL    R0, STATUS
                                          01                 68  E8 000CB       BLBS    STATUS, 7$                              ; 0520
                                                             04  000CE       RET
03  A7                  02                06                 02  F0 000CF 7$:     INSV    #2, #6, #2, IDENT+3                     ; 0522
    67                  1C                00                 6E  F0 000D5       INSV    TMP_IDENT, #0, #28, IDENT               ; 0523
                                          50                 67  D0 000DA       MOVL    IDENT, R0                               ; 0524
                                80008000  8F                 50  D1 000DD       CMPL    R0, #-2147450880
```

```
                                   09  1F 000E4        BLSSU   8$
          8FFFFFFF  8F             50  D1 000E6        CMPL    R0, #-1879048193                    ; 0525
                                   B4  1B 000ED        BLEQU   5$
                     00000000G     8F  DD 000EF  8$:   PUSHL   #UAF$_IDOUTRNG                      ; 0528
                                   75  11 000F5        BRB     14$
                          01A8     C8  9F 000F7  9$:   PUSHAB  SD_VALUEUIC                         ; 0538
                    69             01  FB 000FB        CALLS   #1, CLI$PRESENT
                    11             50  E9 000FE        BLBC    R0, 10$
                                   01  DD 00101        PUSHL   #1                                  ; 0539
                          01A8     C8  9F 00103        PUSHAB  SD_VALUEUIC
          00000000G 00             02  FB 00107        CALLS   #2, PARSE_WILD
                    01             50  E8 0010E        BLBS    R0, 10$
                                   04  00111           RET
                    46 00000000G   00  E9 00112  10$:  BLBC    UIC_FLAG, 13$                       ; 0541
                    0F 00000000G   00  E9 00119        BLBC    GRP_WILD, 11$                       ; 0544
                    08 00000000G   00  E8 00120        BLBS    MEM_WILD, 11$
                       00000000G   8F  DD 00127        PUSHL   #UAF$_NOGRPWILD                     ; 0550
                                   3D  11 0012D        BRB     14$
                                   01  DD 0012F  11$:  PUSHL   #1                                  ; 0559
                                   57  DD 00131        PUSHL   R7
                             02    A7  9F 00133        PUSHAB  IDENT+2                             ; 0558
          00000000G 00             03  FB 00136        CALLS   #3, PARSE_UIC                       ; 0559
                    18             50  E8 0013D        BLBS    R0, 12$
                       00000000G   00  DD 00140        PUSHL   TOKENPTR                            ; 0564
                    7E 00000000G   00  3C 00146        MOVZWL  TOKENLEN, -(SP)                     ; 0563
                                   02  DD 0014D        PUSHL   #2
                       00000000G   8F  DD 0014F        PUSHL   #UAF$_UICERR
                                 0113  31 00155        BRW     27$
                03  A7          C0 8F  8A 00158  12$:  BICB2   #192, IDENT+3                       ; 0567
                                   2C  11 0015D        BRB     16$                                 ; 0541
                    09 00000000G   00  E9 0015F  13$:  BLBC    STR_WILD, 15$                       ; 0575
                       00000000G   8F  DD 00166        PUSHL   #UAF$_WLDNOTALWD                    ; 0581
                                 00EB  31 0016C  14$:  BRW     25$
                       00000000G   00  D4 0016F  15$:  CLRL    FOUND_MATCH                         ; 0584
                       00000000V   00  9F 00175        PUSHAB  UAF$FIND_UIC                        ; 0585
          00000000G 00             01  FB 0017B        CALLS   #1, WILD_USER
                    6B             50  D0 00182        MOVL    R0, RMSERR
                    03             50  E8 00185        BLBS    R0, 16$
                                 00BD  31 00188        BRW     24$
                             04    AE  9F 0018B  16$:  PUSHAB  RESULT_ID                           ; 0605
                             F4    A7  DD 0018E        PUSHL   ATTRIBUTES
                                   67  DD 00191        PUSHL   IDENT
                             34    AE  9F 00193        PUSHAB  NAME_DESC
          00000000G 00             04  FB 00196        CALLS   #4, SYS$ADD_IDENT
                    68             50  D0 0019D        MOVL    R0, STATUS
                    32             50  E8 001A0        BLBS    R0, 18$                             ; 0607
                C0  8F       03    A7  93 001A3        BITB    IDENT+3, #192                       ; 0610
                                   1A  12 001A8        BNEQ    17$
                                   50  DD 001AA        PUSHL   R0                                  ; 0613
                    7E             67  3C 001AC        MOVZWL  IDENT, -(SP)
  7E        02  A7  0E             00  EF 001AF        EXTZV   #0, #14, IDENT+2, -(SP)             ; 0612
                             34    AE  9F 001B5        PUSHAB  NAME_DESC                           ; 0611
                                   03  DD 001B8        PUSHL   #3
                       00000000G   8F  DD 001BA        PUSHL   #UAF$_RDBADDERRU
                    6A             06  FB 001C0        CALLS   #6, LIB$SIGNAL
                                   04  001C3           RET
                                   50  DD 001C4  17$:  PUSHL   R0                                  ; 0615
```

```
                                                67  DD 001C6          PUSHL   IDENT                                   ; 0614
                                            30  AE  9F 001C8          PUSHAB  NAME_DESC                               ;
                                                02  DD 001CB          PUSHL   #2                                      ;
                                        00000000G   8F  DD 001CD          PUSHL   #UAF$_RDBADDERRV                        ;
                                                23  11 001D3          BRB     19$                                     ;
                                 00000000G  00          01  90 001D5 18$:    MOVB    #1, RIGHTSLIST_MODIFIED         ; 0619
                                        C0  8F      07  AE  93 001DC          BITB    RESULT_ID+3, #T92               ; 0620
                                                19  12 001E1          BNEQ    20$                                     ;
                                            7E      04  AE  3C 001E3          MOVZWL  RESULT_ID, -(SP)                ; 0623
                    7E      0A  AE              0E      00  EF 001E7          EXTZV   #0, #14, RESULT_ID+2, -(SP)     ; 0622
                                            30  AE  9F 001ED          PUSHAB  NAME_DESC                               ; 0621
                                                03  DD 001F0          PUSHL   #3                                      ;
                                        00000000G   8F  DD 001F2          PUSHL   #UAF$_RDBADDMSGU                        ;
                                            6A          05  FB 001F8 19$:    CALLS   #5, LIB$SIGNAL                  ;
                                                04  001FB          RET                                             ;
                                            04  AE  DD 001FC 20$:    PUSHL   RESULT_ID                               ; 0624
                                            2C  AE  9F 001FF          PUSHAB  NAME_DESC                               ;
                                                02  DD 00202          PUSHL   #2                                      ;
                                        00000000G   8F  DD 00204          PUSHL   #UAF$_RDBADDMSG                         ;
                                                5F  11 0020A          BRB     27$                                     ;
                                          0190      C8  9F 0020C 21$:    PUSHAB  SD_USER                                 ; 0629
                                            69          01  FB 00210          CALLS   #1, CLI$PRESENT                 ;
                                            11          50  E9 00213          BLBC    R0, 22$                         ;
                                                01  DD 00216          PUSHL   #1                                      ; 0632
                                          0190      C8  9F 00218          PUSHAB  SD_USER                                 ;
                                 00000000G  00          02  FB 0021C          CALLS   #2, PARSE_WILD                  ;
                                            09          50  E8 00223          BLBS    R0, 23$                         ;
                                                04  00226          RET                                             ; 0633
                                        00000000G   8F  DD 00227 22$:    PUSHL   #UAF$_NOIDNAME                          ; 0637
                                                2B  11 0022D          BRB     25$                                     ;
                                        00000000G   00  D4 0022F 23$:    CLRL    FOUND_MATCH                             ; 0640
                                        00000000V   00  9F 00235          PUSHAB  UAF$ADD_IDENT_RECBUF                    ; 0641
                             00000000G  00          01  FB 0023B          CALLS   #1, WILD_USER                   ;
                                        6B          50  D0 00242          MOVL    R0, RMSERR                              ;
                                        26          50  E8 00245          BLBS    R0, 28$                         ;
                                        50          6B  D0 00248 24$:    MOVL    RMSERR, R0                              ; 0644
                             00018282   8F          50  D1 0024B          CMPL    R0, #98994                              ;
                                                0A  12 00252          BNEQ    26$                                     ;
                                        00000000G   8F  DD 00254          PUSHL   #UAF$_BADSPC                            ; 0645
                                            6A          01  FB 0025A 25$:    CALLS   #1, LIB$SIGNAL                  ;
                                                04  0025D          RET                                             ;
                                                50  DD 0025E 26$:    PUSHL   R0                                      ; 0646
                                            2C  AE  9F 00260          PUSHAB  NAME_DESC                               ;
                                                01  DD 00263          PUSHL   #1                                      ;
                                        00000000G   8F  DD 00265          PUSHL   #UAF$_RDBADDERR                         ;
                                            6A          04  FB 0026B 27$:    CALLS   #4, LIB$SIGNAL                  ;
                                                04  0026E 28$:    RET                                             ; 0653

; Routine Size:  623 bytes,    Routine Base:  $CODE$ + 010E

;  657          0654  1
```

```
;  659          0655  1 %SBTTL ' UAF$ADD_IDENT_RECBUF  -  Add rights identifier '
;  660          0656  1
;  661          0657  1 GLOBAL ROUTINE UAF$ADD_IDENT_RECBUF =
;  662          0658  2 BEGIN
;  663          0659  2 !++
;  664          0660  2 !
;  665          0661  2 ! FUNCTIONAL DESCRIPTION:
;  666          0662  2 !
;  667          0663  2 !       This routine add identifiers to the rights data base for
;  668          0664  2 !       a user and his group.  The rouyine uses a UAF record as input
;  669          0665  2 !       to determine what identifiers need to be added.
;  670          0666  2 !
;  671          0667  2 !
;  672          0668  2 ! INPUTS:
;  673          0669  2 !
;  674          0670  2 !       none
;  675          0671  2 !
;  676          0672  2 ! IMPLICIT INPUTS:
;  677          0673  2 !
;  678          0674  2 !       RECBUF must contain a valid UAF record
;  679          0675  2 !
;  680          0676  2 ! OUTPUTS:
;  681          0677  2 !
;  682          0678  2 !       None
;  683          0679  2 !
;  684          0680  2 ! IMPLICIT OUTPUTS:
;  685          0681  2 !
;  686          0682  2 !       none
;  687          0683  2 !
;  688          0684  2 ! ROUTINE VALUE:
;  689          0685  2 !
;  690          0686  2 !       Always true
;  691          0687  2 !
;  692          0688  2 ! SIDE EFFECTS:
;  693          0689  2 !
;  694          0690  2 !       Rights data base is modified
;  695          0691  2 !--
;  696          0692  2
;  697          0693  2
;  698          0694  2 LOCAL
;  699          0695  2     NAME_DESC    : STATDESC,
;  700          0696  2     NAME_BUFF    : VECTOR [ KGB$S_NAME, BYTE ] ,
;  701          0697  2     RESULT_ID    : $BBLOCK [4];
;  702          0698  2
;  703          0699  2 RESULT_ID = 0 ;
;  704          0700  2 IDENT = 0 ;
;  705          0701  2 FOUND_MATCH = TRUE;
;  706          0702  2
;  707          0703  2 !
;  708          0704  2 ! Get the identifier name from the UAF record username field.
;  709          0705  2 !
;  710          0706  2 CH$COPY ( UAF$S_USERNAME, RECBUF[UAF$T_USERNAME], %C' ',
;  711          0707  2           KGB$S_NAME,  NAME_BUFF );
;  712          0708  2 NAME_DESC[LENGTH]  = MIN ( UAF$S_USERNAME,
;  713          0709  2                           KGB$S_NAME,
;  714          0710  2                           (CH$FIND_CH (KGB$S_NAME, NAME_BUFF, %C' ') - NAME_BUFF) ) ;
;  715          0711  2 NAME_DESC[POINTER] = NAME_BUFF ;
```

```
;  716          0712  2
;  717          0713  2 !
;  718          0714  2 ! Build the identifier from the UAF record
;  719          0715  2 !
;  720          0716  2 UAF$FIND_UIC () ;
;  721          0717  2
;  722          0718  2 !
;  723          0719  2 ! Perform the actual service to add the identifier
;  724          0720  2 !
;  725        P 0721  2 STATUS = $ADD_IDENT ( NAME    = NAME_DESC,
;  726        P 0722  2                       ID      = .IDENT,
;  727        P 0723  2                       ATTRIB  = .ATTRIBUTES,
;  728          0724  2                       RESID   = RESULT_ID ) ;
;  729          0725  2
;  730          0726  2 IF NOT .STATUS
;  731          0727  2     THEN
;  732          0728  3         BEGIN
;  733          0729  3         IF .IDENT[UIC$V_FORMAT] EQL UIC$K_UIC_FORMAT
;  734          0730  3             THEN LIB$SIGNAL(UAF$_RDBADDERRU, 3, NAME_DESC,
;  735          0731  3                                 .IDENT[UIC$V_GROUP], .IDENT[UIC$V_MEMBER],
;  736          0732  3                                 .STATUS)
;  737          0733  3             ELSE LIB$SIGNAL(UAF$_RDBADDERRV, 2, NAME_DESC, .IDENT, .STATUS);
;  738          0734  3         END
;  739          0735  2     ELSE
;  740          0736  3         BEGIN
;  741          0737  3         RIGHTSLIST_MODIFIED = TRUE ;
;  742          0738  3         IF .RESULT_ID[UIC$V_FORMAT] EQL UIC$K_UIC_FORMAT
;  743          0739  3             THEN LIB$SIGNAL(UAF$_RDBADDMSGU, 3, NAME_DESC,
;  744          0740  3                             .RESULT_ID[UIC$V_GROUP], .RESULT_ID[UIC$V_MEMBER])
;  745          0741  3             ELSE LIB$SIGNAL(UAF$_RDBADDMSG, 2, NAME_DESC, .RESULT_ID);
;  746          0742  2         END ;
;  747          0743  2
;  748          0744  2 !
;  749          0745  2 ! We will now check for the presence of the group identifier
;  750          0746  2 ! and create it if it doesn't exist.
;  751          0747  2 !
;  752          0748  2 UAF$ADD_GRP_IDENT () ;
;  753          0749  2
;  754          0750  2 RETURN TRUE ;
;  755          0751  2
;  756          0752  1 END ;                                   ! End of UAF$ADD_IDENT_RECBUF


                                                00FC 00000          .ENTRY  UAF$ADD_IDENT_RECBUF, Save R2,R3,R4,R5,R6,- ; 0657
                                                                            R7                                              ;
                                    57 00000000G   00   9E 00002    MOVAB   LIB$SIGNAL, R7                                  ;
                                    56 00000000'   00   9E 00009    MOVAB   IDENT, R6                                       ;
                                    5E             28   C2 00010    SUBL2   #40, SP                                         ;
                                20  AE 010E0000    8F   D0 00013    MOVL    #17694720, NAME_DESC                            ; 0695
                                       24  AE       D4   D4 0001B    CLRL    NAME_DESC+4                                     ;
                                                   7E   D4 0001E    CLRL    RESULT_ID                                       ; 0699
                                                   66   D4 00020    CLRL    IDENT                                           ; 0700
                    00000000G   00                 01   D0 00022    MOVL    #1, FOUND_MATCH                                 ; 0701
                04  AE 00000000G   00             20   28 00029    MOVC3   #32, RECBUF+4, NAME_BUFF                        ; 0706
```

```
                04   AE              20                  20  3A 00032          LOCC    #32, #32, NAME_BUFF                 ; 0710
                                                         02  12 00037          BNEQ    1$
                                                         51  D4 00039          CLRL    R1
                                     50          04  AE  9E 0003B  1$:        MOVAB   NAME_BUFF, R0
                                     51          50  C2 0003F               SUBL2   R0, R1
                                     20          51  D1 00042               CMPL    R1, #32                             ; 0708
                                                         03  15 00045          BLEQ    2$
                                     51          20  D0 00047               MOVL    #32, R1
                          24         AE          51  B0 0004A  2$:        MOVW    R1, NAME_DESC
                          28         AE          04  AE  9E 0004E          MOVAB   NAME_BUFF, NAME_DESC+4              ; 0711
                00000000V            00          00  FB 00053          CALLS   #0, UAF$FIND_UIC                    ; 0716
                                                         5E  DD 0005A          PUSHL   SP                                  ; 0724
                                     F4          A6  DD 0005C          PUSHL   ATTRIBUTES
                                                         66  DD 0005F          PUSHL   IDENT
                                     30          AE  9F 00061          PUSHAB  NAME_DESC
                00000000G            00              04  FB 00064          CALLS   #4, SYS$ADD_IDENT
                00000000'            00              50  D0 0006B          MOVL    R0, STATUS
                                     33              50  E8 00072          BLBS    R0, 4$                              ; 0726
                          C0         8F          03  A6  93 00075          BITB    IDENT+3, #192                       ; 0729
                                                         1B  12 0007A          BNEQ    3$
                                                         50  DD 0007C          PUSHL   R0                                  ; 0732
                                     7E                  66  3C 0007E          MOVZWL  IDENT, -(SP)                        ; 0731
          7E    02   A6              0E                  00  EF 00081          EXTZV   #0, #14, IDENT+2, -(SP)
                                     30          AE  9F 00087          PUSHAB  NAME_DESC                           ; 0730
                                                         03  DD 0008A          PUSHL   #3
                                00000000G            8F  DD 0008C          PUSHL   #UAF$_RDBADDERRU
                                     67                  06  FB 00092          CALLS   #6, LIB$SIGNAL
                                                         48  11 00095          BRB     7$
                                                         50  DD 00097  3$:        PUSHL   R0                                  ; 0733
                                                         66  DD 00099          PUSHL   IDENT
                                         2C          AE  9F 0009B          PUSHAB  NAME_DESC                           ;
                                                         02  DD 0009E          PUSHL   #2
                                00000000G            8F  DD 000A0          PUSHL   #UAF$_RDBADDERRV
                                                         22  11 000A6          BRB     5$                                  ;
                00000000G            00              01  90 000A8  4$:        MOVB    #1, RIGHTSLIST_MODIFIED             ; 0737
                          C0         8F          03  AE  93 000AF          BITB    RESULT_ID+3, #192                   ; 0738
                                                         19  12 000B4          BNEQ    6$
                                     7E                  6E  3C 000B6          MOVZWL  RESULT_ID, -(SP)                    ; 0740
          7E    06   AE              0E                  00  EF 000B9          EXTZV   #0, #14, RESULT_ID+2, -(SP)
                                         2C          AE  9F 000BF          PUSHAB  NAME_DESC                           ; 0739
                                                         03  DD 000C2          PUSHL   #3
                                00000000G            8F  DD 000C4          PUSHL   #UAF$_RDBADDMSGU
                                     67                  05  FB 000CA  5$:        CALLS   #5, LIB$SIGNAL
                                                         10  11 000CD          BRB     7$
                                                         6E  DD 000CF  6$:        PUSHL   RESULT_ID                           ; 0741
                                         28          AE  9F 000D1          PUSHAB  NAME_DESC
                                                         02  DD 000D4          PUSHL   #2
                                00000000G            8F  DD 000D6          PUSHL   #UAF$_RDBADDMSG
                                     67                  04  FB 000DC          CALLS   #4, LIB$SIGNAL
                          FB9F       CF                  00  FB 000DF  7$:        CALLS   #0, UAF$ADD_GRP_IDENT               ; 0748
                                     50                  01  D0 000E4          MOVL    #1, R0                              ; 0750
                                                             04  000E7          RET                                         ; 0752

; Routine Size:  232 bytes,    Routine Base:  $CODE$ + 037D

;  757          0753  1
```

```
;  759     0754  1 %SBTTL ' UAF$BUILD_HOLDER -  Build a holder quadword from UAF record'
;  760     0755  1
;  761     0756  1 GLOBAL ROUTINE UAF$BUILD_HOLDER=
;  762     0757  2 BEGIN
;  763     0758  2 !++
;  764     0759  2 !
;  765     0760  2 ! FUNCTIONAL DESCRIPTION:
;  766     0761  2 !
;  767     0762  2 !       This routine is used to build a quadword holder from
;  768     0763  2 !       a UAF record
;  769     0764  2 !
;  770     0765  2 ! INPUTS:
;  771     0766  2 !
;  772     0767  2 !       none
;  773     0768  2 !
;  774     0769  2 ! IMPLICIT INPUTS:
;  775     0770  2 !
;  776     0771  2 !       RECBUF must contain a valid UAF record
;  777     0772  2 !
;  778     0773  2 ! OUTPUTS:
;  779     0774  2 !
;  780     0775  2 !       None
;  781     0776  2 !
;  782     0777  2 ! IMPLICIT OUTPUTS:
;  783     0778  2 !
;  784     0779  2 !       HOLDER is initialized
;  785     0780  2 !
;  786     0781  2 ! ROUTINE VALUE:
;  787     0782  2 !
;  788     0783  2 !       Always true
;  789     0784  2 !
;  790     0785  2 ! SIDE EFFECTS:
;  791     0786  2 !
;  792     0787  2 !       none
;  793     0788  2 !--
;  794     0789  2
;  795     0790  2
;  796     0791  2 BIND HOLDER_VEC = HOLDER : VECTOR [2,LONG] ;
;  797     0792  2
;  798     0793  2 FOUND_MATCH = TRUE;
;  799     0794  2
;  800     0795  2 HOLDER[UIC$V_FORMAT] = UIC$K_UIC_FORMAT ;
;  801     0796  2 HOLDER[UIC$V_GROUP]  = .RECBUF [UAF$W_GRP] ;
;  802     0797  2 HOLDER[UIC$V_MEMBER] = .RECBUF [UAF$W_MEM] ;
;  803     0798  2
;  804     0799  2 HOLDER_VEC[1] = 0 ;
;  805     0800  2
;  806     0801  2 RETURN TRUE ;
;  807     0802  2
;  808     0803  1 END ;                                   ! End of UAF$BUILD_HOLDER


                                                  HOLDER_VEC=          HOLDER

                                0004 00000          .ENTRY  UAF$BUILD_HOLDER, Save R2                    ; 0756
```

```
                                52 00000000'  00  9E 00002         MOVAB   HOLDER, R2                                      ;
                      00000000G 00            01  D0 00009         MOVL    #1, FOUND_MATCH                                 ; 0793
                             03 A2        C0  8F  8A 00010         BICB2   #192, HOLDER+3                                  ; 0795
      02    A2              0E 00 00000000G   00  F0 00015         INSV    RECBUF+38, #0, #14, HOLDER+2                    ; 0796
                               62 00000000G   00  B0 0001F         MOVW    RECBUF+36, HOLDER                               ; 0797
                                        04    A2  D4 00026         CLRL    HOLDER_VEC+4                                    ; 0799
                               50               01  D0 00029         MOVL    #1, R0                                          ; 0801
                                                   04 0002C         RET                                                     ; 0803

; Routine Size:  45 bytes,    Routine Base:  $CODE$ + 0465

;  809          0804  1
```

```
;  811          0805  1 %SBTTL ' UAF$CREATE_RDB  -  Create a new RIGHTSLIST'
;  812          0806  1
;  813          0807  1 GLOBAL ROUTINE UAF$CREATE_RDB : NOVALUE =
;  814          0808  2 BEGIN
;  815          0809  2 !++
;  816          0810  2 !
;  817          0811  2 ! FUNCTIONAL DESCRIPTION:
;  818          0812  2 !
;  819          0813  2 !       This routine is used to create a new rights data base.  It
;  820          0814  2 !       is invoked via the CREATE/RIGHTS command. It does not, however,
;  821          0815  2 !       create the file if it already exists.
;  822          0816  2 !
;  823          0817  2 ! INPUTS:
;  824          0818  2 !
;  825          0819  2 !       none
;  826          0820  2 !
;  827          0821  2 ! IMPLICIT INPUTS:
;  828          0822  2 !
;  829          0823  2 !       command line
;  830          0824  2 !
;  831          0825  2 ! OUTPUTS:
;  832          0826  2 !
;  833          0827  2 !       None
;  834          0828  2 !
;  835          0829  2 ! IMPLICIT OUTPUTS:
;  836          0830  2 !
;  837          0831  2 !       none
;  838          0832  2 !
;  839          0833  2 ! ROUTINE VALUE:
;  840          0834  2 !
;  841          0835  2 !       none
;  842          0836  2 !
;  843          0837  2 ! SIDE EFFECTS:
;  844          0838  2 !
;  845          0839  2 !       A new rights data base id created and initialized
;  846          0840  2 !
;  847          0841  2 !--
;  848          0842  2
;  849          0843  2
;  850          0844  2 LOCAL
;  851          0845  2     I                       : BYTE,
;  852          0846  2     SYSTEM_ID               : VECTOR [2,LONG],
;  853          0847  2     SYSTEM_ID_PTR           : LONG ;
;  854          0848  2
;  855          0849  2 !
;  856          0850  2 ! If a system ID was specified then get the first two numeric
;  857          0851  2 ! values.  If more than two exist then ignore the extra,  If less
;  858          0852  2 ! than two exist fill in with zeros
;  859          0853  2 !
;  860          0854  2 SYSTEM_ID_PTR = 0 ;
;  861          0855  2 SYSTEM_ID[0] = 0 ;
;  862          0856  2 SYSTEM_ID[1] = 0 ;
;  863          0857  2 I = 0 ;
;  864          0858  2 WHILE (CLI$GET_VALUE ( SD_SYSTEM_ID, TOKENDSC )) AND
;  865          0859  2       (.I LEQ 1) DO
;  866          0860  3     BEGIN
;  867          0861  3     STATUS = GETVAL ( SYSTEM_ID [.I] , LONGWORD_LENGTH );
```

```
;  868     0862  3     IF NOT .STATUS
;  869     0863  3         THEN RETURN ;
;  870     0864  3     I = .I + 1 ;
;  871     0865  3     SYSTEM_ID_PTR = SYSTEM_ID ;
;  872     0866  2     END ;
;  873     0867  2
;  874     0868  2 !
;  875     0869  2 ! Now call the service to create the file
;  876     0870  2 !
;  877     0871  2
;  878     0872  2 STATUS = $CREATE_RDB(sysid=.SYSTEM_ID_PTR);
;  879     0873  2 if not .STATUS then
;  880     0874  2    LIB$SIGNAL(UAF$_RDBCREERR, 0, .STATUS)
;  881     0875  2 else
;  882     0876  3    begin
;  883     0877  3         RIGHTSLIST_MODIFIED = TRUE;
;  884     0878  3         RDB_EXISTS = TRUE;
;  885     0879  3         LIB$SIGNAL(UAF$_RDBCREMSG);
;  886     0880  2    end;
;  887     0881  2
;  888     0882  2 return;
;  889     0883  2
;  890     0884  1 end;                                    ! End of UAF$CREATE_RDB


                                                        .EXTRN  SYS$CREATE_RDB

                                    003C 00000          .ENTRY  UAF$CREATE_RDB, Save R2,R3,R4,R5        ; 0807
                 55 00000000G   00   9E 00002          MOVAB   LIB$SIGNAL, R5
                 54 00000000'   00   9E 00009          MOVAB   STATUS, R4
                 5E                  04 C2 00010       SUBL2   #4, SP
                                     53 D4 00013       CLRL    SYSTEM_ID_PTR                           ; 0854
                                     7E D4 00015       CLRL    SYSTEM_ID                               ; 0855
                           04        AE D4 00017       CLRL    SYSTEM_ID+4                             ; 0856
                                     52 94 0001A       CLRB    I                                       ; 0857
                    00000000G   00   9F 0001C 1$:      PUSHAB  TOKENDSC                                ; 0858
                        0178        C4 9F 00022       PUSHAB  SD_SYSTEM_ID
00000000G         00                02 FB 00026       CALLS   #2, CLI$GET_VALUE
                  22                50 E9 0002D       BLBC    R0, 2$
                  01                52 91 00030       CMPB    I, #1                                   ; 0859
                                    1D 1A 00033       BGTRU   2$
                                    20 DD 00035       PUSHL   #32                                     ; 0861
                  50                52 9A 00037       MOVZBL  I, R0
                           04 AE40  DF 0003A          PUSHAL  SYSTEM_ID[R0]
00000000G         00                02 FB 0003E       CALLS   #2, GETVAL
                  64                50 D0 00045       MOVL    R0, STATUS
                  3B                64 E9 00048       BLBC    STATUS, 4$                              ; 0862
                                    52 96 0004B       INCB    I                                       ; 0864
                  53                6E 9E 0004D       MOVAB   SYSTEM_ID, SYSTEM_ID_PTR                ; 0865
                                    CA 11 00050       BRB     1$                                      ; 0858
                                    53 DD 00052 2$:   PUSHL   SYSTEM_ID_PTR                           ; 0872
00000000G         00                01 FB 00054       CALLS   #1, SYS$CREATE_RDB
                  64                50 D0 0005B       MOVL    R0, STATUS
                  0E                50 E8 0005E       BLBS    R0, 3$                                  ; 0873
                                    50 DD 00061       PUSHL   R0                                      ; 0874
                                    7E D4 00063       CLRL    -(SP)
```

```
                                 000000000G  8F  DD 00065          PUSHL   #UAF$_RDBCREERR                          ;
                              65             03  FB 0006B          CALLS   #3, LIB$SIGNAL                          ;
                                             04  0006E          RET                                             ;
                 000000000G   00             01  90 0006F 3$:   MOVB    #1, RIGHTSLIST_MODIFIED                  ; 0877
                 000000000G   00             01  D0 00076       MOVL    #1, RDB_EXISTS                           ; 0878
                                 000000000G  8F  DD 0007D          PUSHL   #UAF$_RDBCREMSG                         ; 0879
                              65             01  FB 00083          CALLS   #1, LIB$SIGNAL                          ;
                                             04  00086 4$:        RET                                             ; 0884

; Routine Size:  135 bytes,    Routine Base:  $CODE$ + 0492

;  891          0885  1
```

```
;  893          0886  1 %SBTTL ' UAF$DISPLAY_IDENT  -  Display rights identifiers '
;  894          0887  1
;  895          0888  1 GLOBAL ROUTINE UAF$DISPLAY_IDENT : NOVALUE =
;  896          0889  2 BEGIN
;  897          0890  2 !++
;  898          0891  2 !
;  899          0892  2 ! FUNCTIONAL DESCRIPTION:
;  900          0893  2 !
;  901          0894  2 !       This is the common routine used by the SHOW/IDENT and LIST/IDENT
;  902          0895  2 !       command.  It will interogate the rights data base and output the
;  903          0896  2 !       desird information
;  904          0897  2 !
;  905          0898  2 !
;  906          0899  2 ! INPUTS:
;  907          0900  2 !
;  908          0901  2 !       none
;  909          0902  2 !
;  910          0903  2 ! IMPLICIT INPUTS:
;  911          0904  2 !
;  912          0905  2 !       RABPTR must point to the output RAB
;  913          0906  2 !       RDB_LIST_FLAG = 1 => list operation
;  914          0907  2 !                     = 0 => show operation
;  915          0908  2 !       SHOW_ID_FULL  = 1 => show/list full
;  916          0909  2 !                       0 => show/list brief
;  917          0910  2 !
;  918          0911  2 ! OUTPUTS:
;  919          0912  2 !
;  920          0913  2 !       None
;  921          0914  2 !
;  922          0915  2 ! IMPLICIT OUTPUTS:
;  923          0916  2 !
;  924          0917  2 !       none
;  925          0918  2 !
;  926          0919  2 ! ROUTINE VALUE:
;  927          0920  2 !
;  928          0921  2 !       none
;  929          0922  2 !
;  930          0923  2 ! SIDE EFFECTS:
;  931          0924  2 !
;  932          0925  2 !       none
;  933          0926  2 !--
;  934          0927  2
;  935          0928  2
;  936          0929  2 LOCAL
;  937          0930  2     WILD_CARD    : LONG,
;  938          0931  2     NAME_DESC    : STATDESC,
;  939          0932  2     NAME_BUFF    : VECTOR [ KGB$S_NAME, BYTE ] ;
;  940          0933  2
;  941          0934  2 IDENT = 0 ;
;  942          0935  2
;  943          0936  2 !
;  944          0937  2 ! We will now get the id name parameter from
;  945          0938  2 ! the command line.  If it isn't present then we
;  946          0939  2 ! MUST have either the /VALUE or /OWNER qualifier present.
;  947          0940  2 !
;  948          0941  2
;  949          0942  2 IF NOT CLI$PRESENT ( SD_TOKEN1 )
```

```
;  950          0943  2     THEN
;  951          0944  3         IF NOT  (CLI$PRESENT ( SD_VALUE ) OR
;  952          0945  3                  CLI$PRESENT ( SD_USER ))
;  953          0946  2             THEN
;  954          0947  3                 BEGIN
;  955          0948  3                 LIB$SIGNAL(UAF$_NOIDNAME);
;  956          0949  3                 RETURN ;
;  957          0950  2                 END ;
;  958          0951  2
;  959          0952  2 IF .RDB_LIST_FLAG
;  960          0953  2     THEN LIB$SIGNAL(UAF$_LSTMSG1);
;  961          0954  2 RDB_HEADER_FLAG = TRUE ;
;  962          0955  2
;  963          0956  2 IF CLI$GET_VALUE ( SD_TOKEN1, TOKENDSC )
;  964          0957  2     THEN
;  965          0958  3         BEGIN
;  966          0959  3
;  967          0960  3         CH$COPY ( .TOKENLEN, .TOKENPTR, %C' ', KGB$S_NAME,  NAME_BUFF );
;  968          0961  3         NAME_DESC[LENGTH]  = MIN ( .TOKENLEN , KGB$S_NAME ) ;
;  969          0962  3         NAME_DESC[POINTER] = NAME_BUFF ;
;  970          0963  3
;  971          0964  3         !
;  972          0965  3         ! Check for a wild card
;  973          0966  3         !
;  974          0967  3         IF CH$EQL ( .NAME_DESC[LENGTH], .NAME_DESC[POINTER],
;  975          0968  3                     1, UPLIT(BYTE('*')), %C' ')
;  976          0969  3             THEN
;  977          0970  4                 BEGIN
;  978          0971  4                 IDENT = UIC$K_MATCH_ALL ; ! Wild card specified by ID value of -1
;  979          0972  4                 WILD_CARD = TRUE ;
;  980          0973  4                 END
;  981          0974  3             ELSE
;  982          0975  4                 BEGIN
;  983          0976  4                 !
;  984          0977  4                 ! Get the IDENT from the ascii string
;  985          0978  4                 !
;  986        P 0979  4                 STATUS = $ASCTOID ( NAME = NAME_DESC,
;  987          0980  4                                     ID   = IDENT ) ;
;  988          0981  4                 IF NOT .STATUS
;  989          0982  4                     THEN
;  990          0983  5                         BEGIN
;  991          0984  5                             IF .RDB_LIST_FLAG
;  992          0985  5                             THEN
;  993          0986  6                                 BEGIN
;  994          0987  6                                 LIB$SIGNAL(UAF$_LSTERR, 0, .STATUS);
;  995          0988  6                                 RLSTFAB [FAB$V_DLT] = TRUE ;
;  996          0989  6                                 END
;  997          0990  5                             ELSE LIB$SIGNAL(UAF$_SHOWERR, 0, .STATUS);
;  998          0991  5                         RETURN ;
;  999          0992  4                         END ;
; 1000          0993  4                 WILD_CARD = FALSE ;
; 1001          0994  3                 END ;
; 1002          0995  3
; 1003          0996  3         STATUS = UAF$WRITE_IDENT ( .IDENT, .SHOW_ID_FULL, .WILD_CARD ) ;
; 1004          0997  3         IF NOT .STATUS
; 1005          0998  3             THEN IF .RDB_LIST_FLAG
; 1006          0999  3                 THEN
```

```
;  1007          1000  4                    BEGIN
;  1008          1001  4                    LIB$SIGNAL(UAF$_LSTERR, 0, .STATUS);
;  1009          1002  4                    RLSTFAB [FAB$V_DLT] = TRUE ;
;  1010          1003  4                    RETURN ;
;  1011          1004  4                    END
;  1012          1005  3                ELSE LIB$SIGNAL(UAF$_SHOWERR, 0, .STATUS);
;  1013          1006  2            END ;
;  1014          1007  2
;  1015          1008  2 !
;  1016          1009  2 ! If the /VALUE qualifier was specified then get it's value
;  1017          1010  2 ! First check fo IDENTIFIER:
;  1018          1011  2 !
;  1019          1012  2 IF CLI$GET_VALUE ( SD_VALUEIDENTIFIER, TOKENDSC )
;  1020          1013  2     THEN
;  1021          1014  3         BEGIN
;  1022          1015  3         LOCAL
;  1023          1016  3             TMP_IDENT : LONG INITIAL (0) ;
;  1024          1017  3
;  1025          1018  3         STATUS = GETVAL ( TMP_IDENT, LONGWORD_LENGTH );
;  1026          1019  3         IF NOT .STATUS
;  1027          1020  3             THEN RETURN ;
;  1028          1021  3         IDENT [UIC$V_FORMAT]  = UIC$K_ID_FORMAT ;
;  1029          1022  3         IDENT [UIC$V_ID_CODE] = .TMP_IDENT ;
;  1030          1023  3         STATUS = UAF$WRITE_IDENT ( .IDENT, .SHOW_ID_FULL, FALSE ) ;
;  1031          1024  3         IF NOT .STATUS
;  1032          1025  3             THEN IF .RDB_LIST_FLAG
;  1033          1026  3                 THEN
;  1034          1027  4                     BEGIN
;  1035          1028  4                     LIB$SIGNAL(UAF$_LSTERR, 0, .STATUS);
;  1036          1029  4                     RLSTFAB [FAB$V_DLT] = TRUE ;
;  1037          1030  4                     RETURN ;
;  1038          1031  4                     END
;  1039          1032  3                 ELSE LIB$SIGNAL(UAF$_SHOWERR, 0, .STATUS);
;  1040          1033  3             END
;  1041          1034  2 ELSE IF CLI$PRESENT (SD_VALUEUIC)
;  1042          1035  2     THEN
;  1043          1036  3         BEGIN
;  1044          1037  3         !
;  1045          1038  3         ! Now check for UIC
;  1046          1039  3         !
;  1047          1040  3         IF NOT PARSE_WILD (SD_VALUEUIC, TRUE )
;  1048          1041  3             THEN RETURN ;
;  1049          1042  3         IF .UIC_FLAG
;  1050          1043  3             THEN
;  1051          1044  4                 BEGIN
;  1052          1045  4                 !
;  1053          1046  4                 ! The UIC format was used,  parse it to fine the
;  1054          1047  4                 ! group and member number.
;  1055          1048  4                 ! Wild card group and member numbers are allowed
;  1056          1049  4                 !
;  1057          1050  4                 IF NOT PARSE_UIC ( IDENT[UIC$V_GROUP],
;  1058          1051  4                                    IDENT[UIC$V_MEMBER],
;  1059          1052  4                                    TRUE )
;  1060          1053  4                     THEN
;  1061          1054  5                         BEGIN
;  1062          1055  5                         LIB$SIGNAL(UAF$_UICERR, 2, .TOKENLEN, .TOKENPTR);
;  1063          1056  5                         RETURN ;
```

```
;1064     1057  4                    END ;
;1065     1058  4                WILD_CARD = FALSE ;
;1066     1059  4                IDENT[UIC$V_FORMAT] = UIC$K_UIC_FORMAT ;
;1067     1060  4                END
;1068     1061  3            ELSE
;1069     1062  4                BEGIN
;1070     1063  4                !
;1071     1064  4                ! A user name was specified.  Get the UAF record
;1072     1065  4                ! for this user and pull the UIC out.
;1073     1066  4                !
;1074     1067  4                WILD_CARD = .STR_WILD ;
;1075     1068  4                IF .STR_WILD
;1076     1069  4                    THEN IDENT = UIC$K_MATCH_ALL
;1077     1070  4                    ELSE
;1078     1071  5                        BEGIN
;1079     1072  5                        FOUND_MATCH = FALSE ;
;1080     1073  6                        IF RMSBAD ( WILD_USER (UAF$FIND_UIC))
;1081     1074  5                            THEN
;1082     1075  6                                BEGIN
;1083     1076  6                                IF .RMSERR EQL RMS$_RNF
;1084     1077  6                                    THEN LIB$SIGNAL(UAF$_BADSPC)
;1085     1078  6                                    ELSE IF .RDB_LIST_FLAG
;1086     1079  6                                        THEN LIB$SIGNAL(UAF$_LSTERR, 0,
;1087     1080  6                                                        .STATUS)
;1088     1081  6                                        ELSE LIB$SIGNAL(UAF$_SHOWERR, 0,
;1089     1082  6                                                        .STATUS);
;1090     1083  6                                RETURN ;
;1091     1084  5                                END ;
;1092     1085  4                        END ;
;1093     1086  3                END ;
;1094     1087  3            STATUS = UAF$WRITE_IDENT ( .IDENT, .SHOW_ID_FULL, .WILD_CARD ) ;
;1095     1088  3            IF NOT .STATUS
;1096     1089  3                THEN IF .RDB_LIST_FLAG
;1097     1090  3                    THEN
;1098     1091  4                        BEGIN
;1099     1092  4                        LIB$SIGNAL(UAF$_LSTERR, 0, .STATUS);
;1100     1093  4                        RLSTFAB [FAB$V_DLT] = TRUE ;
;1101     1094  4                        RETURN ;
;1102     1095  4                        END
;1103     1096  3                    ELSE LIB$SIGNAL(UAF$_SHOWERR, 0, .STATUS);
;1104     1097  2            END;
;1105     1098  2 IF CLI$PRESENT (SD_USER)
;1106     1099  2     THEN IF PARSE_WILD (SD_USER, TRUE )
;1107     1100  2         THEN
;1108     1101  3             BEGIN
;1109     1102  3             FOUND_MATCH = FALSE ;
;1110     1103  4             IF RMSBAD ( WILD_USER (UAF$DISPLAY_IDENT_RECBUF))
;1111     1104  3                 THEN
;1112     1105  4                     BEGIN
;1113     1106  4                     IF .RMSERR EQL RMS$_RNF
;1114     1107  4                         THEN LIB$SIGNAL(UAF$_BADSPC)
;1115     1108  4                         ELSE IF .RDB_LIST_FLAG
;1116     1109  4                             THEN
;1117     1110  5                                 BEGIN
;1118     1111  5                                 LIB$SIGNAL(UAF$_LSTERR, 0, .STATUS);
;1119     1112  5                                 RLSTFAB [FAB$V_DLT] = TRUE ;
;1120     1113  5                                 RETURN ;
```

```
;  1121          1114  5                                END
;  1122          1115  4                            ELSE LIB$SIGNAL(UAF$_SHOWERR, 0, .STATUS);
;  1123          1116  3                        END ;
;  1124          1117  2                END;
;  1125          1118  2
;  1126          1119  2 IF .RDB_LIST_FLAG
;  1127          1120  2     THEN LIB$SIGNAL(UAF$_RLSTMSG);
;  1128          1121  2
;  1129          1122  2 RETURN
;  1130          1123  2
;  1131          1124  1 END ;                                   ! End of UAF$DISPLAY_IDENT


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                      0006B     .BLKB   1
                                                   2A 0006C P.AAN:  .ASCII  \*\                                                    ;


                                                                .PSECT  $CODE$,NOWRT,2

                                                      0FFC 00000    .ENTRY  UAF$DISPLAY_IDENT, Save R2,R3,R4,R5,R6,R7,-           ; 0888
                                                                            R8,R9,R10,R11
                                                5B 00000000G  8F  D0 00002      MOVL    #UAF$_SHOWERR, R11                      ;
                                                5A 00000000G  00  9E 00009      MOVAB   CLI$PRESENT, R10                        ;
                                                59 00000000G  00  9E 00010      MOVAB   LIB$SIGNAL, R9                          ;
                                                58 00000000'  00  9E 00017      MOVAB   IDENT, R8                               ;
                                                57 00000000'  00  9E 0001E      MOVAB   STATUS, R7                              ;
                                                5E            2C  C2 00025      SUBL2   #44, SP                                 ;
                                          24    AE 010E0000   8F  D0 00028      MOVL    #17694720, NAME_DESC                    ; 0931
                                                      28      AE  D4 00030      CLRL    NAME_DESC+4                             ;
                                                              68  D4 00033      CLRL    IDENT                                   ; 0934
                                                    0180      C7  9F 00035      PUSHAB  SD_TOKEN1                               ; 0942
                                                6A            01  FB 00039      CALLS   #1, CLI$PRESENT                         ;
                                                1D            50  E8 0003C      BLBS    R0, 1$                                  ;
                                                    0198      C7  9F 0003F      PUSHAB  SD_VALUE                                ; 0944
                                                6A            01  FB 00043      CALLS   #1, CLI$PRESENT                         ;
                                                13            50  E8 00046      BLBS    R0, 1$                                  ;
                                                    0190      C7  9F 00049      PUSHAB  SD_USER                                 ; 0945
                                                6A            01  FB 0004D      CALLS   #1, CLI$PRESENT                         ;
                                                09            50  E8 00050      BLBS    R0, 1$                                  ;
                                                   00000000G  8F  DD 00053      PUSHL   #UAF$_NOIDNAME                          ; 0948
                                                         023E 31  00059      BRW     29$                                     ;
                                                09       F1   A8  E9 0005C 1$:   BLBC    RDB_LIST_FLAG, 2$                       ; 0952
                                                   00000000G  8F  DD 00060      PUSHL   #UAF$_LSTMSG1                           ; 0953
                                                69            01  FB 00066      CALLS   #1, LIB$SIGNAL                          ;
                                          F0    A8            01  90 00069 2$:   MOVB    #1, RDB_HEADER_FLAG                     ; 0954
                                                   00000000G  00  9F 0006D      PUSHAB  TOKENDSC                                ; 0956
                                                    0180      C7  9F 00073      PUSHAB  SD_TOKEN1                               ;
                                       00000000G  00            02  FB 00077      CALLS   #2, CLI$GET_VALUE                       ;
                                                03            50  E8 0007E      BLBS    R0, 3$                                  ;
                                                         007E 31  00081      BRW     8$                                      ;
                                                56 00000000G  00  3C 00084 3$:   MOVZWL  TOKENLEN, R6                           ; 0960
                                                50 00000000G  00  D0 0008B      MOVL    TOKENPTR, R0                            ;
                                    20    20          60            56  2C 00092      MOVC5   R6, (R0), #32, #32, NAME_BUFF           ;
```

```
                                    04    AE    00097                                                          :
                              50          56 D0 00099          MOVL    R6, R0                                  : 0961
                              20          50 B1 0009C          CMPW    R0, #32                                 :
                                          03 1B 0009F          BLEQU   4$                                      :
                              50          20 D0 000A1          MOVL    #32, R0                                 :
                       24     AE          50 B0 000A4 4$:      MOVW    R0, NAME_DESC                           :
                       28     AE    04    AE 9E 000A8          MOVAB   NAME_BUFF, NAME_DESC+4                  : 0962
       01        20    28     BE    24    AE 2D 000AD          CMPC5   NAME_DESC, @NAME_DESC+4, #32, #1, P.AAN : 0967
                                00000000' 00    000B4                                                          :
                                          08 12 000B9          BNEQ    5$                                      :
                              68          01 CE 000BB          MNEGL   #1, IDENT                               : 0971
                              52          01 D0 000BE          MOVL    #1, WILD_CARD                           : 0972
                                          1D 11 000C1          BRB     7$                                      : 0967
                                          7E D4 000C3 5$:      CLRL    -(SP)                                   : 0980
                                          58 DD 000C5          PUSHL   R8                                      :
                                    2C    AE 9F 000C7          PUSHAB  NAME_DESC                               :
                     00000000G  00        03 FB 000CA          CALLS   #3, SYS$ASCTOID                         :
                              67          50 D0 000D1          MOVL    R0, STATUS                              :
                              07          50 E8 000D4          BLBS    R0, 6$                                  : 0981
                              73    F1    A8 E8 000D7          BLBS    RDB_LIST_FLAG, 12$                      : 0984
                                        0127 31 000DB          BRW     20$                                     :
                                          52 D4 000DE 6$:      CLRL    WILD_CARD                               : 0993
                                          52 DD 000E0 7$:      PUSHL   WILD_CARD                               : 0996
                              7E    FC    A7 9A 000E2          MOVZBL  SHOW_ID_FULL, -(SP)                     :
                                          68 DD 000E6          PUSHL   IDENT                                   :
                     00000000V  00        03 FB 000E8          CALLS   #3, UAF$WRITE_IDENT                     :
                              67          50 D0 000EF          MOVL    R0, STATUS                              :
                              0D          50 E8 000F2          BLBS    R0, 8$                                  : 0997
                              55    F1    A8 E8 000F5          BLBS    RDB_LIST_FLAG, 12$                      : 0998
                                          50 DD 000F9          PUSHL   R0                                      : 1005
                                          7E D4 000FB          CLRL    -(SP)                                   :
                                          5B DD 000FD          PUSHL   R11                                     :
                              69          03 FB 000FF          CALLS   #3, LIB$SIGNAL                          :
                                00000000G 00 9F 00102 8$:      PUSHAB  TOKENDSC                                : 1012
                                     01A0 C7 9F 00108          PUSHAB  SD_VALUEIDENTIFIER                      :
                     00000000G  00        02 FB 0010C          CALLS   #2, CLI$GET_VALUE                       :
                              42          50 E9 00113          BLBC    R0, 14$                                 :
                                          6E D4 00116          CLRL    TMP_IDENT                               : 1014
                                          20 DD 00118          PUSHL   #32                                     : 1018
                                    04    AE 9F 0011A          PUSHAB  TMP_IDENT                               :
                     00000000G  00        02 FB 0011D          CALLS   #2, GETVAL                              :
                              67          50 D0 00124          MOVL    R0, STATUS                              :
                              01          67 E8 00127          BLBS    STATUS, 9$                              : 1019
                                             04 0012A          RET                                             :
   03     A8             02   06          02 F0 0012B 9$:      INSV    #2, #6, #2, IDENT+3                     : 1021
          68             1C   00          6E F0 00131          INSV    TMP_IDENT, #0, #28, IDENT               : 1022
                                          7E D4 00136          CLRL    -(SP)                                   : 1023
                              7E    FC    A7 9A 00138 10$:     MOVZBL  SHOW_ID_FULL, -(SP)                     :
                                          68 DD 0013C          PUSHL   IDENT                                   :
                     00000000V  00        03 FB 0013E          CALLS   #3, UAF$WRITE_IDENT                     :
                              67          50 D0 00145          MOVL    R0, STATUS                              :
                              03          50 E9 00148          BLBC    R0, 12$                                 : 1024
                                        00CF 31 0014B 11$:     BRW     24$                                     :
                              03    F1    A8 E9 0014E 12$:     BLBC    RDB_LIST_FLAG, 13$                      : 1025
                                        011E 31 00152          BRW     26$                                     :
                                        00BC 31 00155 13$:     BRW     23$                                     :
                                    01A8  C7 9F 00158 14$:     PUSHAB  SD_VALUEUIC                             : 1034
```

```
                        6A                01  FB 0015C          CALLS    #1, CLI$PRESENT          ;
                        E9                50  E9 0015F          BLBC     R0, 11$                  ;
                                          01  DD 00162          PUSHL    #1                       ; 1040
                                 01A8     C7  9F 00164          PUSHAB   SD_VALUEUIC              ;
              00000000G 00                02  FB 00168          CALLS    #2, PARSE_WILD           ;
                        01                50  E8 0016F          BLBS     R0, 15$                  ;
                                              04 00172          RET                               ;
                     33 00000000G         00  E9 00173 15$:     BLBC     UIC_FLAG, 17$            ; 1042
                                          01  DD 0017A          PUSHL    #1                       ; 1051
                                          58  DD 0017C          PUSHL    R8                       ;
                                   02     A8  9F 0017E          PUSHAB   IDENT+2                  ; 1050
              00000000G 00                03  FB 00181          CALLS    #3, PARSE_UIC            ; 1051
                        19                50  E8 00188          BLBS     R0, 16$                  ;
                           00000000G      00  DD 0018B          PUSHL    TOKENPTR                 ; 1055
                     7E 00000000G         00  3C 00191          MOVZWL   TOKENLEN, -(SP)          ;
                                          02  DD 00198          PUSHL    #2                       ;
                           00000000G      8F  DD 0019A          PUSHL    #UAF$_UICERR             ;
                        69                04  FB 001A0          CALLS    #4, LIB$SIGNAL           ;
                                              04 001A3          RET                               ; 1054
                                          52  D4 001A4 16$:     CLRL     WILD_CARD                ; 1058
                   03   A8         C0     8F  8A 001A6          BICB2    #192, IDENT+3            ; 1059
                                          62  11 001AB          BRB      22$                      ; 1042
                     50 00000000G         00  D0 001AD 17$:     MOVL     STR_WILD, R0             ; 1067
                        52                50  D0 001B4          MOVL     R0, WILD_CARD            ;
                        05                50  E9 001B7          BLBC     R0, 18$                  ; 1068
                        68                01  CE 001BA          MNEGL    #1, IDENT                ; 1069
                                          50  11 001BD          BRB      22$                      ;
                           00000000G      00  D4 001BF 18$:     CLRL     FOUND_MATCH              ; 1072
                           00000000V      00  9F 001C5          PUSHAB   UAF$FIND_UIC             ; 1073
              00000000G 00                01  FB 001CB          CALLS    #1, WILD_USER            ;
              00000000G 00                50  D0 001D2          MOVL     R0, RMSERR               ;
                        33                50  E8 001D9          BLBS     R0, 22$                  ;
              000182B2  8F 00000000G      00  D1 001DC          CMPL     RMSERR, #98994           ; 1076
                                          09  12 001E7          BNEQ     19$                      ;
                           00000000G      8F  DD 001E9          PUSHL    #UAF$_BADSPC             ; 1077
                                        00A8  31 001EF          BRW      29$                      ;
                        50                67  D0 001F2 19$:     MOVL     STATUS, R0               ; 1080
                        0C         F1     A8  E9 001F5          BLBC     RDB_LIST_FLAG, 20$       ; 1078
                                          50  DD 001F9          PUSHL    R0                       ; 1080
                                          7E  D4 001FB          CLRL     -(SP)                    ; 1079
                           00000000G      8F  DD 001FD          PUSHL    #UAF$_LSTERR             ;
                                          06  11 00203          BRB      21$                      ;
                                          50  DD 00205 20$:     PUSHL    R0                       ; 1082
                                          7E  D4 00207          CLRL     -(SP)                    ; 1081
                                          5B  DD 00209          PUSHL    R11                      ;
                        69                03  FB 0020B 21$:     CALLS    #3, LIB$SIGNAL           ;
                                              04 0020E          RET                               ; 1075
                                          52  DD 0020F 22$:     PUSHL    WILD_CARD                ; 1087
                                        FF24  31 00211          BRW      10$                      ;
                                          50  DD 00214 23$:     PUSHL    R0                       ; 1096
                                          7E  D4 00216          CLRL     -(SP)                    ;
                                          5B  DD 00218          PUSHL    R11                      ;
                        69                03  FB 0021A          CALLS    #3, LIB$SIGNAL           ;
                                 0190     C7  9F 0021D 24$:     PUSHAB   SD_USER                  ; 1098
                        6A                01  FB 00221          CALLS    #1, CLI$PRESENT          ;
                        69                50  E9 00224          BLBC     R0, 28$                  ;
                                          01  DD 00227          PUSHL    #1                       ; 1099
```

```
                                 0190   C7  9F 00229          PUSHAB  SD_USER                                    ;
                    00000000G    00     02  FB 0022D          CALLS   #2, PARSE_WILD                            ;
                                 59     50  E9 00234          BLBC    R0, 28$                                   ;
                                        00  D4 00237          CLRL    FOUND_MATCH                               ; 1102
                                        00  9F 0023D          PUSHAB  UAF$DISPLAY_IDENT_RECBUF                  ; 1103
                    00000000G    00     01  FB 00243          CALLS   #1, WILD_USER                             ;
                    00000000G    00     50  D0 0024A          MOVL    R0, RMSERR                                ;
                                 3C     50  E8 00251          BLBS    R0, 28$                                   ;
                    000182B2     8F     00  D1 00254          CMPL    RMSERR, #98994                            ; 1106
                                        0B  12 0025F          BNEQ    25$                                       ;
                                 00000000G  8F  DD 00261          PUSHL   #UAF$_BADSPC                          ; 1107
                                 69         01  FB 00267          CALLS   #1, LIB$SIGNAL                        ;
                                            24  11 0026A          BRB     28$                                   ;
                                 50         67  D0 0026C 25$:     MOVL    STATUS, R0                            ; 1111
                                 14      F1 A8  E9 0026F          BLBC    RDB_LIST_FLAG, 27$                    ; 1108
                                            50  DD 00273 26$:     PUSHL   R0                                    ; 1111
                                            7E  D4 00275          CLRL    -(SP)                                 ;
                                 00000000G  8F  DD 00277          PUSHL   #UAF$_LSTERR                          ;
                                 69         03  FB 0027D          CALLS   #3, LIB$SIGNAL                        ;
                          00C1   C7      80  8F  88 00280          BISB2   #128, RLSTFAB+5                       ; 1112
                                            04  00286          RET                                           ; 1110
                                            50  DD 00287 27$:     PUSHL   R0                                    ; 1115
                                            7E  D4 00289          CLRL    -(SP)                                 ;
                                            5B  DD 0028B          PUSHL   R11                                   ;
                                 69         03  FB 0028D          CALLS   #3, LIB$SIGNAL                        ;
                                 09      F1 A8  E9 00290 28$:     BLBC    RDB_LIST_FLAG, 30$                    ; 1119
                                 00000000G  8F  DD 00294          PUSHL   #UAF$_RLSTMSG                         ; 1120
                                 69         01  FB 0029A 29$:     CALLS   #1, LIB$SIGNAL                        ;
                                            04  0029D 30$:     RET                                           ; 1124

; Routine Size:  670 bytes,    Routine Base:  $CODE$ + 0519

; 1132          1125  1
```

```
;  1134          1126  1 %SBTTL ' UAF$DISPLAY_IDENT_RECBUF - Display identifier from RDB'
;  1135          1127  1
;  1136          1128  1 GLOBAL ROUTINE UAF$DISPLAY_IDENT_RECBUF =
;  1137          1129  2 BEGIN
;  1138          1130  2 !++
;  1139          1131  2 !
;  1140          1132  2 ! FUNCTIONAL DESCRIPTION:
;  1141          1133  2 !
;  1142          1134  2 !       Display the identifier which corresponds to the UIC in the current
;  1143          1135  2 !       record in RECBUF
;  1144          1136  2 !
;  1145          1137  2 ! INPUTS:
;  1146          1138  2 !
;  1147          1139  2 !       none
;  1148          1140  2 !
;  1149          1141  2 ! IMPLICIT INPUTS:
;  1150          1142  2 !
;  1151          1143  2 !       RECBUF must contain a valid UAF record
;  1152          1144  2 !       RABPTR must point to the output RAB
;  1153          1145  2 !       SHOW_ID_FULL  = 1 => show/list full
;  1154          1146  2 !                       0 => show/list brief
;  1155          1147  2 !
;  1156          1148  2 ! OUTPUTS:
;  1157          1149  2 !
;  1158          1150  2 !       None
;  1159          1151  2 !
;  1160          1152  2 ! IMPLICIT OUTPUTS:
;  1161          1153  2 !
;  1162          1154  2 !       none
;  1163          1155  2 !
;  1164          1156  2 ! ROUTINE VALUE:
;  1165          1157  2 !
;  1166          1158  2 !       status or write operation
;  1167          1159  2 !
;  1168          1160  2 ! SIDE EFFECTS:
;  1169          1161  2 !
;  1170          1162  2 !       none
;  1171          1163  2 !--
;  1172          1164  2
;  1173          1165  2
;  1174          1166  2 LOCAL
;  1175          1167  2     NAME_DESC   : STATDESC,
;  1176          1168  2     NAME_BUFF   : VECTOR [ KGB$S_NAME, BYTE ] ;
;  1177          1169  2
;  1178          1170  2 IDENT = 0 ;
;  1179          1171  2 FOUND_MATCH = TRUE;
;  1180          1172  2
;  1181          1173  2 !
;  1182          1174  2 ! First see if the identifier is there and get it's name.
;  1183          1175  2 ! If it doesn't exist then just return
;  1184          1176  2 !
;  1185          1177  2 UAF$FIND_UIC () ;                          ! Form an IDENT from the UIC in RECBUF
;  1186          1178  2 NAME_DESC[POINTER] = NAME_BUFF ;
;  1187          1179  2 NAME_DESC[LENGTH]  = KGB$S_NAME ;
;  1188        P 1180  2 STATUS = $IDTOASC ( ID     = .IDENT,
;  1189        P 1181  2                     NAMLEN = NAME_DESC[LENGTH],
;  1190          1182  2                     NAMBUF = NAME_DESC ) ;
```

```
;  1191          1183  2 IF NOT .STATUS
;  1192          1184  2     THEN RETURN TRUE ;
;  1193          1185  2
;  1194          1186  2 !
;  1195          1187  2 ! Now call the display routine
;  1196          1188  2 !
;  1197          1189  2 STATUS = UAF$WRITE_IDENT ( .IDENT, .SHOW_ID_FULL, FALSE ) ;
;  1198          1190  2
;  1199          1191  2 RETURN .STATUS ;
;  1200          1192  2
;  1201          1193  1 END ;                                   ! End of UAF$DISPLAY_IDENT_RECBUF


                                    000C 00000        .ENTRY  UAF$DISPLAY_IDENT_RECBUF, Save R2,R3    ; 1128
                          53 00000000'  00  9E 00002  MOVAB   IDENT, R3                               ;
                          52 00000000'  00  9E 00009  MOVAB   STATUS, R2                              ;
                                5E      28  C2 00010  SUBL2   #40, SP                                 ;
                          20  AE 010E0000 8F  D0 00013  MOVL    #17694720, NAME_DESC                  ; 1167
                                  24  AE  D4 0001B  CLRL    NAME_DESC+4                             ;
                                      63  D4 0001E  CLRL    IDENT                                   ; 1170
                00000000G  00         01  D0 00020  MOVL    #1, FOUND_MATCH                         ; 1171
                00000000V  00         00  FB 00027  CALLS   #0, UAF$FIND_UIC                        ; 1177
                      24   AE         6E  9E 0002E  MOVAB   NAME_BUFF, NAME_DESC+4                  ; 1178
                      20   AE         20  B0 00032  MOVW    #32, NAME_DESC                          ; 1179
                                      7E  7C 00036  CLRQ    -(SP)                                   ; 1182
                                      7E  D4 00038  CLRL    -(SP)                                   ;
                                2C    AE  9F 0003A  PUSHAB  NAME_DESC                               ;
                                30    AE  9F 0003D  PUSHAB  NAME_DESC                               ;
                                      63  DD 00040  PUSHL   IDENT                                  ;
                          00000000G   00    FB 00042  CALLS   #6, SYS$IDTOASC                       ;
                                      62  50 00049  MOVL    R0, STATUS                              ;
                                      04  62 0004C  BLBS    STATUS, 1$                              ; 1183
                                      50  01 0004F  MOVL    #1, R0                                  ; 1184
                                          04 00052  RET                                             ;
                                      7E  D4 00053  1$:  CLRL    -(SP)                            ; 1189
                                7E    FC  A2 9A 00055  MOVZBL  SHOW_ID_FULL, -(SP)                  ;
                                      63  DD 00059  PUSHL   IDENT                                  ;
                          00000000V   00    FB 0005B  CALLS   #3, UAF$WRITE_IDENT                   ;
                                      62  50 00062  MOVL    R0, STATUS                              ;
                                          04 00065  RET                                             ; 1193

; Routine Size:  102 bytes,    Routine Base:  $CODE$ + 07B7

; 1202          1194  1
```

```
;  1204          1195  1 %SBTTL ' UAF$DISPLAY_RIGHTS - Display rights held by user'
;  1205          1196  1
;  1206          1197  1 GLOBAL ROUTINE UAF$DISPLAY_RIGHTS : NOVALUE =
;  1207          1198  2 BEGIN
;  1208          1199  2 !++
;  1209          1200  2 !
;  1210          1201  2 ! FUNCTIONAL DESCRIPTION:
;  1211          1202  2 !
;  1212          1203  2 !       This is the common  outine used by the SHOW/RIGHTS and LIST/RIGHTS
;  1213          1204  2 !       command.  It will i terogate the rights data base and output the
;  1214          1205  2 !       desird information
;  1215          1206  2 !
;  1216          1207  2 ! INPUTS:
;  1217          1208  2 !
;  1218          1209  2 !       none
;  1219          1210  2 !
;  1220          1211  2 ! IMPLICIT INPUTS:
;  1221          1212  2 !
;  1222          1213  2 !       RABPTR must point to the output RAB
;  1223          1214  2 !       RDB_LIST_FLAG = 1 => list operation
;  1224          1215  2 !                     = 0 => show operation
;  1225          1216  2 !
;  1226          1217  2 ! OUTPUTS:
;  1227          1218  2 !
;  1228          1219  2 !       None
;  1229          1220  2 !
;  1230          1221  2 ! IMPLICIT OUTPUTS:
;  1231          1222  2 !
;  1232          1223  2 !       none
;  1233          1224  2 !
;  1234          1225  2 ! ROUTINE VALUE:
;  1235          1226  2 !
;  1236          1227  2 !       none
;  1237          1228  2 !
;  1238          1229  2 ! SIDE EFFECTS:
;  1239          1230  2 !
;  1240          1231  2 !       none
;  1241          1232  2 !--
;  1242          1233  2
;  1243          1234  2
;  1244          1235  2 BIND
;  1245          1236  2     HOLDER_VEC          = HOLDER : VECTOR [2,LONG] ;
;  1246          1237  2
;  1247          1238  2 !
;  1248          1239  2 ! If the user id parameter is not present then the /OWNER
;  1249          1240  2 ! qualifier must be.
;  1250          1241  2 !
;  1251          1242  3 IF NOT (CLI$PRESENT ( SD_TOKEN1 ) OR
;  1252          1243  3           CLI$PRESENT ( SD_USER ))
;  1253          1244  2     THEN
;  1254          1245  3         BEGIN
;  1255          1246  3         LIB$SIGNAL(UAF$_NOIDNAME);
;  1256          1247  3         RLSTFAB [FAB$V_DLT] = TRUE ;
;  1257          1248  3         RETURN ;
;  1258          1249  2         END ;
;  1259          1250  2
;  1260          1251  2 IF .RDB_LIST_FLAG
```

```
;  1261          1252  2      THEN LIB$SIGNAL(UAF$_LSTMSG1);
;  1262          1253  2 RDB_HEADER_FLAG = TRUE ;
;  1263          1254  2
;  1264          1255  2
;  1265          1256  2 IF CLI$GET_VALUE ( SD_TOKEN1, TOKENDSC )
;  1266          1257  2     THEN
;  1267          1258  3         BEGIN
;  1268          1259  3         HOLDER_VEC[1] = 0 ;
;  1269          1260  3
;  1270          1261  3         !
;  1271          1262  3         ! Convert the ID from ASCII to a longword format
;  1272          1263  3         !
;  1273        P 1264  3         STATUS = $ASCTOID ( NAME   = TOKENDSC,
;  1274        P 1265  3                             ID     = HOLDER_VEC[0],
;  1275          1266  3                             ATTRIB = 0 ) ;
;  1276          1267  3
;  1277          1268  3         IF NOT .STATUS
;  1278          1269  3             THEN
;  1279          1270  4                 BEGIN
;  1280          1271  4                 IF .RDB_LIST_FLAG
;  1281          1272  4                     THEN
;  1282          1273  5                         BEGIN
;  1283          1274  5                         LIB$SIGNAL(UAF$_LSTERR, 0, .STATUS);
;  1284          1275  5                         RLSTFAB [FAB$V_DLT] = TRUE ;
;  1285          1276  5                         END
;  1286          1277  4                     ELSE LIB$SIGNAL(UAF$_SHOWERR, 0, .STATUS);
;  1287          1278  4                 RETURN ;
;  1288          1279  3                 END ;
;  1289          1280  3
;  1290          1281  3
;  1291          1282  3         STATUS = UAF$WRITE_RIGHTS ( HOLDER ) ;
;  1292          1283  3         IF NOT .STATUS
;  1293          1284  3             THEN IF .RDB_LIST_FLAG
;  1294          1285  3                 THEN
;  1295          1286  4                     BEGIN
;  1296          1287  4                     LIB$SIGNAL(UAF$_LSTERR, 0, .STATUS);
;  1297          1288  4                     RLSTFAB [FAB$V_DLT] = TRUE ;
;  1298          1289  4                     RETURN ;
;  1299          1290  4                     END
;  1300          1291  3                 ELSE LIB$SIGNAL(UAF$_SHOWERR, 0, .STATUS);
;  1301          1292  2         END ;
;  1302          1293  2
;  1303          1294  2 IF CLI$PRESENT (SD_USER)
;  1304          1295  2     THEN
;  1305          1296  3         BEGIN
;  1306          1297  4         IF NOT (STATUS =  PARSE_WILD (SD_USER, TRUE ))
;  1307          1298  3             THEN IF .RDB_LIST_FLAG
;  1308          1299  3                 THEN
;  1309          1300  4                     BEGIN                                          ;
;  1310          1301  4                     LIB$SIGNAL(UAF$_LSTERR, 0, .STATUS);
;  1311          1302  4                     RLSTFAB [FAB$V_DLT] = TRUE ;                   ;
;  1312          1303  4                     RETURN ;
;  1313          1304  4                     END
;  1314          1305  3                 ELSE LIB$SIGNAL(UAF$_SHOWERR, 0, .STATUS);
;  1315          1306  3
;  1316          1307  3         FOUND_MATCH = FALSE ;
;  1317          1308  4         IF RMSBAD ( WILD_USER (UAF$DISPLAY_RIGHTS_RECBUF))
```

```
;  1318          1309  3              THEN
;  1319          1310  4                  BEGIN
;  1320          1311  4                  IF .RMSERR EQL RMS$_RNF
;  1321          1312  4                      THEN LIB$SIGNAL(UAF$_BADSPC)
;  1322          1313  4                      ELSE IF .RDB_LIST_FLAG
;  1323          1314  4                          THEN
;  1324          1315  5                              BEGIN
;  1325          1316  5                              LIB$SIGNAL(UAF$_LSTERR, 0, .STATUS);
;  1326          1317  5                              RLSTFAB [FAB$V_DLT] = TRUE ;
;  1327          1318  5                              RETURN ;
;  1328          1319  5                              END
;  1329          1320  4                          ELSE LIB$SIGNAL(UAF$_SHOWERR, 0, .STATUS);
;  1330          1321  3                  END ;
;  1331          1322  2          END ;
;  1332          1323  2
;  1333          1324  2 IF .RDB_LIST_FLAG
;  1334          1325  2     THEN LIB$SIGNAL(UAF$_RLSTMSG);
;  1335          1326  2 RETURN ;
;  1336          1327  2
;  1337          1328  1 END ;                                   !End of UAF$DISPLAY_RIGHTS


                                                    HOLDER_VEC=         HOLDER

                                    01FC 00000          .ENTRY  UAF$DISPLAY_RIGHTS, Save R2,R3,R4,R5,R6,R7,-; 1197
                                                                R8                                           ;
                            58 00000000G   00   9E 00002        MOVAB   RMSERR, R8                           ;
                            57 00000000G   00   9E 00009        MOVAB   TOKENDSC, R7                         ;
                            56 00000000G   00   9E 00010        MOVAB   CLI$PRESENT, R6                      ;
                            55 00000000G   8F   D0 00017        MOVL    #UAF$_SHOWERR, R5                    ;
                            54 00000000G   00   9E 0001E        MOVAB   LIB$SIGNAL, R4                       ;
                            53 00000000'   00   9E 00025        MOVAB   RDB_LIST_FLAG, R3                    ;
                            52 00000000'   00   9E 0002C        MOVAB   STATUS, R2                           ;
                                 0180      C2   9F 00033        PUSHAB  SD_TOKEN1                            ; 1242
                            66             01   FB 00037        CALLS   #1, CLI$PRESENT                      ;
                            16             50   E8 0003A        BLBS    R0, 1$                               ;
                                 0190      C2   9F 0003D        PUSHAB  SD_USER                              ; 1243
                            66             01   FB 00041        CALLS   #1, CLI$PRESENT                      ;
                            0C             50   E8 00044        BLBS    R0, 1$                               ;
                                00000000G  8F   DD 00047        PUSHL   #UAF$_NOIDNAME                       ; 1246
                            64             01   FB 0004D        CALLS   #1, LIB$SIGNAL                       ;
                                       00CC 31   00050        BRW     9$                                   ; 1247
                            09             63   E9 00053 1$:    BLBC    RDB_LIST_FLAG, 2$                    ; 1251
                                00000000G  8F   DD 00056        PUSHL   #UAF$_LSTMSG1                        ; 1252
                            64             01   FB 0005C        CALLS   #1, LIB$SIGNAL                       ;
                         FF A3             01   90 0005F 2$:    MOVB    #1, RDB_HEADER_FLAG                  ; 1253
                                           57   DD 00063        PUSHL   R7                                   ; 1256
                                 0180      C2   9F 00065        PUSHAB  SD_TOKEN1                            ;
                    00000000G    00        02   FB 00069        CALLS   #2, CLI$GET_VALUE                    ;
                            40             50   E9 00070        BLBC    R0, 4$                               ;
                                 0B        A3   D4 00073        CLRL    HOLDER_VEC+4                         ; 1259
                                           7E   D4 00076        CLRL    -(SP)                                ; 1266
                                 07        A3   9F 00078        PUSHAB  HOLDER_VEC                           ;
                                           57   DD 0007B        PUSHL   R7                                   ;
                    00000000G    00             03   FB 0007D        CALLS   #3, SYS$ASCTOID                      ;
```

```
                                62              50  DO 00084          MOVL    R0, STATUS                                      ;
                                0D              50  E8 00087          BLBS    R0, 3$                                          ; 1268
                                46              63  E8 0008A          BLBS    RDB_LIST_FLAG, 5$                               ; 1271
                                                50  DD 0008D          PUSHL   R0                                              ; 1277
                                                7E  D4 0008F          CLRL    -(SP)
                                                55  DD 00091          PUSHL   R5
                                64              03  FB 00093          CALLS   #3, LIB$SIGNAL
                                                04  00096          RET                                                     ; 1270
                                      07        A3  9F 00097 3$:       PUSHAB  HOLDER                                          ; 1282
                00000000V       00              01  FB 0009A          CALLS   #1, UAF$WRITE_RIGHTS
                                62              50  D0 000A1          MOVL    R0, STATUS
                                0C              50  E8 000A4          BLBS    R0, 4$                                          ; 1283
                                68              63  E8 000A7          BLBS    RDB_LIST_FLAG, 8$                               ; 1284
                                                50  DD 000AA          PUSHL   R0                                              ; 1291
                                                7E  D4 000AC          CLRL    -(SP)
                                                55  DD 000AE          PUSHL   R5
                                64              03  FB 000B0          CALLS   #3, LIB$SIGNAL
                                      0190      C2  9F 000B3 4$:       PUSHAB  SD_USER                                         ; 1294
                                66              01  FB 000B7          CALLS   #1, CLI$PRESENT
                                72              50  E9 000BA          BLBC    R0, 11$
                                                01  DD 000BD          PUSHL   #1                                              ; 1297
                                      0190      C2  9F 000BF          PUSHAB  SD_USER
                00000000G       00              02  FB 000C3          CALLS   #2, PARSE_WILD
                                62              50  D0 000CA          MOVL    R0, STATUS
                                0F              50  E8 000CD          BLBS    R0, 6$
                                50              62  D0 000D0          MOVL    STATUS, R0                                      ; 1301
                                3C              63  E8 000D3 5$:       BLBS    RDB_LIST_FLAG, 8$                               ; 1298
                                                50  DD 000D6          PUSHL   R0                                              ; 1305
                                                7E  D4 000D8          CLRL    -(SP)
                                                55  DD 000DA          PUSHL   R5
                                64              03  FB 000DC          CALLS   #3, LIB$SIGNAL
                      00000000G         00      D4 000DF 6$:       CLRL    FOUND_MATCH                                     ; 1307
                      00000000V         00      9F 000E5          PUSHAB  UAF$DISPLAY_RIGHTS_RECBUF                       ; 1308
                00000000G       00              01  FB 000EB          CALLS   #1, WILD_USER
                                68              50  D0 000F2          MOVL    R0, RMSERR
                                37              50  E8 000F5          BLBS    R0, 11$
                000182B2        8F              68  D1 000F8          CMPL    RMSERR, #98994                                  ; 1311
                                                0B  12 000FF          BNEQ    7$
                      00000000G         8F      DD 00101          PUSHL   #UAF$_BADSPC                                    ; 1312
                                64              01  FB 00107          CALLS   #1, LIB$SIGNAL
                                                23  11 0010A          BRB     11$
                                50              62  D0 0010C 7$:       MOVL    STATUS, R0                                      ; 1316
                                14              63  E9 0010F          BLBC    RDB_LIST_FLAG, 10$                              ; 1313
                                                50  DD 00112 8$:       PUSHL   R0                                              ; 1316
                                                7E  D4 00114          CLRL    -(SP)
                      00000000G         8F      DD 00116          PUSHL   #UAF$_LSTERR
                                64              03  FB 0011C          CALLS   #3, LIB$SIGNAL
                    00C1        C2        80    8F  88 0011F 9$:       BISB2   #128, RLSTFAB+5                                 ; 1317
                                                04  00125          RET                                                     ; 1315
                                                50  DD 00126 10$:      PUSHL   R0                                              ; 1320
                                                7E  D4 00128          CLRL    -(SP)
                                                55  DD 0012A          PUSHL   R5
                                64              03  FB 0012C          CALLS   #3, LIB$SIGNAL
                                09              63  E9 0012F 11$:      BLBC    RDB_LIST_FLAG, 12$                              ; 1324
                      00000000G         8F      DD 00132          PUSHL   #UAF$_RLSTMSG                                   ; 1325
                                64              01  FB 00138          CALLS   #1, LIB$SIGNAL
                                                04  0013B 12$:      RET                                                     ; 1328
```

; Routine Size: 316 bytes,   Routine Base: $CODE$ + 081D

; 1338   1329 1

```
;1340     1330  1 %SBTTL ' UAF$DISPLAY_RIGHTS_RECBUF - Display ids held'
;1341     1331  1
;1342     1332  1 GLOBAL ROUTINE UAF$DISPLAY_RIGHTS_RECBUF =
;1343     1333  2 BEGIN
;1344     1334  2 !++
;1345     1335  2 !
;1346     1336  2 ! FUNCTIONAL DESCRIPTION:
;1347     1337  2 !
;1348     1338  2 !       Display the identifiers which are held by the identifier
;1349     1339  2 !       which corresponds to the UIC in the current
;1350     1340  2 !       record in RECBUF
;1351     1341  2 !
;1352     1342  2 ! INPUTS:
;1353     1343  2 !
;1354     1344  2 !       none
;1355     1345  2 !
;1356     1346  2 ! IMPLICIT INPUTS:
;1357     1347  2 !
;1358     1348  2 !       RECBUF must contain a valid UAF record
;1359     1349  2 !       RABPTR must point to the output RAB
;1360     1350  2 !
;1361     1351  2 ! OUTPUTS:
;1362     1352  2 !
;1363     1353  2 !       None
;1364     1354  2 !
;1365     1355  2 ! IMPLICIT OUTPUTS:
;1366     1356  2 !
;1367     1357  2 !       none
;1368     1358  2 !
;1369     1359  2 ! ROUTINE VALUE:
;1370     1360  2 !
;1371     1361  2 !       none
;1372     1362  2 !
;1373     1363  2 ! SIDE EFFECTS:
;1374     1364  2 !
;1375     1365  2 !       none
;1376     1366  2 !--
;1377     1367  2
;1378     1368  2
;1379     1369  2 BIND
;1380     1370  2     HOLDER_VEC = HOLDER : VECTOR [2,LONG ] ;
;1381     1371  2
;1382     1372  2 FOUND_MATCH = TRUE;
;1383     1373  2
;1384     1374  2 !
;1385     1375  2 ! First see if the identifier is there and get it's name.
;1386     1376  2 ! If it doesn't exist then just return
;1387     1377  2 !
;1388     1378  2 UAF$BUILD_HOLDER() ;                    ! build a holder from the UIC in RECBUF
;1389     1379  2
;1390     1380  2 STATUS = $IDTOASC ( ID      = .HOLDER_VEC[0] ) ;
;1391     1381  2 IF NOT .STATUS
;1392     1382  2     THEN RETURN TRUE ;
;1393     1383  2
;1394     1384  2 !
;1395     1385  2 ! Now call the display routine
;1396     1386  2 !
```

```
;  1397          1387  2 STATUS = UAF$WRITE_RIGHTS ( HOLDER ) ;
;  1398          1388  2
;  1399          1389  2 RETURN .STATUS ;
;  1400          1390  2
;  1401          1391  1 END ;                                    ! End of UAF$DISPLAY_RIGHTS_RECBUF


                                                                 HOLDER_VEC=         HOLDER

                                                    000C 00000          .ENTRY  UAF$DISPLAY_RIGHTS_RECBUF, Save R2,R3      ; 1332
                                  53 00000000'  00   9E 00002          MOVAB   HOLDER_VEC, R3                             ;
                                  52 00000000'  00   9E 00009          MOVAB   STATUS, R2                                 ;
                        00000000G 00            01   D0 00010          MOVL    #1, FOUND_MATCH                            ; 1372
                            FAF0  CF            00   FB 00017          CALLS   #0, UAF$BUILD_HOLDER                       ; 1378
                                                7E   7C 0001C          CLRQ    -(SP)                                      ; 1380
                                                7E   7C 0001E          CLRQ    -(SP)                                      ;
                                                7E   D4 00020          CLRL    -(SP)                                      ;
                                                63   DD 00022          PUSHL   HOLDER_VEC                                 ;
                        00000000G 00            06   FB 00024          CALLS   #6, SYS$IDTOASC                            ;
                                  62            50   D0 0002B          MOVL    R0, STATUS                                 ;
                                  04            62   E8 0002E          BLBS    STATUS, 1$                                 ; 1381
                                  50            01   D0 00031          MOVL    #1, R0                                     ; 1382
                                                      04 00034          RET                                                ;
                                                53   DD 00035 1$:       PUSHL   R3                                         ; 1387
                        00000000V 00            01   FB 00037          CALLS   #1, UAF$WRITE_RIGHTS                       ;
                                  62            50   D0 0003E          MOVL    R0, STATUS                                 ;
                                                      04 00041          RET                                                ; 1391

; Routine Size:  66 bytes,    Routine Base:  $CODE$ + 0959

;  1402          1392  1
```

```
;  1404          1393  1 %SBTTL ' UAF$FIND_UIC  -  get uic from UAF record'
;  1405          1394  1
;  1406          1395  1 GLOBAL ROUTINE UAF$FIND_UIC =
;  1407          1396  2 BEGIN
;  1408          1397  2 !++
;  1409          1398  2 !
;  1410          1399  2 ! FUNCTIONAL DESCRIPTION:
;  1411          1400  2 !
;  1412          1401  2 !       Build an identifier fron the UIC in the current UAF record
;  1413          1402  2 !
;  1414          1403  2 ! INPUTS:
;  1415          1404  2 !
;  1416          1405  2 !       none
;  1417          1406  2 !
;  1418          1407  2 ! IMPLICIT INPUTS:
;  1419          1408  2 !
;  1420          1409  2 !       RECBUF must contain a valid UAF record
;  1421          1410  2 !
;  1422          1411  2 ! OUTPUTS:
;  1423          1412  2 !
;  1424          1413  2 !       None
;  1425          1414  2 !
;  1426          1415  2 ! IMPLICIT OUTPUTS:
;  1427          1416  2 !
;  1428          1417  2 !       IDENT is initialized
;  1429          1418  2 !
;  1430          1419  2 ! ROUTINE VALUE:
;  1431          1420  2 !
;  1432          1421  2 !       Always true
;  1433          1422  2 !
;  1434          1423  2 ! SIDE EFFECTS:
;  1435          1424  2 !
;  1436          1425  2 !       none
;  1437          1426  2 !--
;  1438          1427  2
;  1439          1428  2 FOUND_MATCH = TRUE;
;  1440          1429  2
;  1441          1430  2 IDENT[UIC$V_FORMAT] = UIC$K_UIC_FORMAT ;
;  1442          1431  2 IDENT[UIC$V_GROUP]  = .RECBUF [UAF$W_GRP] ;
;  1443          1432  2 IDENT[UIC$V_MEMBER] = .RECBUF [UAF$W_MEM] ;
;  1444          1433  2
;  1445          1434  2 RETURN TRUE ;
;  1446          1435  2
;  1447          1436  1 END ;                                   ! End of UAF$FIND_UIC


                                        0004 00000      .ENTRY  UAF$FIND_UIC, Save R2                   ; 1395
                            52 00000000'  00   9E 00002 MOVAB   IDENT, R2                               ;
                 00000000G  00            01   D0 00009 MOVL    #1, FOUND_MATCH                        ; 1428
                        03  A2            8F   8A 00010 BICB2   #192, IDENT+3                          ; 1430
      02   A2          0E  00 00000000G  00   F0 00015 INSV    RECBUF+38, #0, #14, IDENT+2             ; 1431
                            62 00000000G  00   B0 0001F MOVW    RECBUF+36, IDENT                        ; 1432
                            50            01   D0 00026 MOVL    #1, R0                                  ; 1434
                                              04 00029 RET                                              ; 1436
```

```
; Routine Size:  42 bytes,    Routine Base:  $CODE$ + 099B

; 1448          1437  1
```

```
;  1450          1438  1 %SBTTL ' UAF$GRANT_IDENT  -  Add RDB holder record '
;  1451          1439  1
;  1452          1440  1 GLOBAL ROUTINE UAF$GRANT_IDENT: NUVALUE =
;  1453          1441  2 BEGIN
;  1454          1442  2 !++
;  1455          1443  2 !
;  1456          1444  2 ! FUNCTIONAL DESCRIPTION:
;  1457          1445  2 !
;  1458          1446  2 !       Add a holder record to the rights data base for the specified
;  1459          1447  2 !       identifier.  This routine is invoked via the GRANT/IDENT command.
;  1460          1448  2 !
;  1461          1449  2 !
;  1462          1450  2 ! INPUTS:
;  1463          1451  2 !
;  1464          1452  2 !       none
;  1465          1453  2 !
;  1466          1454  2 ! IMPLICIT INPUTS:
;  1467          1455  2 !
;  1468          1456  2 !       Command line
;  1469          1457  2 !
;  1470          1458  2 ! OUTPUTS:
;  1471          1459  2 !
;  1472          1460  2 !       None
;  1473          1461  2 !
;  1474          1462  2 ! IMPLICIT OUTPUTS:
;  1475          1463  2 !
;  1476          1464  2 !       none
;  1477          1465  2 !
;  1478          1466  2 ! ROUTINE VALUE:
;  1479          1467  2 !
;  1480          1468  2 !       none
;  1481          1469  2 !
;  1482          1470  2 ! SIDE EFFECTS:
;  1483          1471  2 !
;  1484          1472  2 !       rights data base is modified
;  1485          1473  2 !
;  1486          1474  2 !--
;  1487          1475  2
;  1488          1476  2
;  1489          1477  2 LOCAL
;  1490          1478  2     ID_NAME_DESC        : STATDESC,
;  1491          1479  2     ID_NAME_BUFF        : VECTOR [ KGB$S_NAME, BYTE ] ,
;  1492          1480  2     USER_NAME_DESC      : STATDESC,
;  1493          1481  2     USER_NAME_BUFF      : VECTOR [ KGB$S_NAME, BYTE ] ;
;  1494          1482  2
;  1495          1483  2 BIND
;  1496          1484  2     HOLDER_VEC = HOLDER : VECTOR [2,LONG] ;
;  1497          1485  2
;  1498          1486  2 !
;  1499          1487  2 ! First we will get the identifier name.
;  1500          1488  2 ! This is a required parameter so it is guaranteed present
;  1501          1489  2 ! We will use the rights data base to translate the name
;  1502          1490  2 !
;  1503          1491  2 CLI$GET_VALUE ( SD_TOKEN1, TOKENDSC ) ;
;  1504          1492  2
;  1505          1493  2 CH$COPY ( .TOKENLEN, .TOKENPTR, %C' ', KGB$S_NAME, ID_NAME_BUFF );
;  1506          1494  2 ID_NAME_DESC[LENGTH] = MIN ( .TOKENLEN , KGB$S_NAME ) ;
```

```
;  1507     1495  2 ID_NAME_DESC[POINTER] = ID_NAME_BUFF ;
;  1508     1496  2
;  1509     1497  2 !
;  1510     1498  2 ! The user id parameter is required so we are guaranteed
;  1511     1499  2 ! that it is present.  We will get the parameter before finishing
;  1512     1500  2 ! up with the ID name so that we have it in case we must signal an
;  1513     1501  2 ! error
;  1514     1502  2 !
;  1515     1503  2 PARSE_WILD( SD_TOKEN2, TRUE ) ;
;  1516     1504  2 CH$COPY ( .TOKENLEN, .TOKENPTR, %C' ', KGB$S_NAME, USER_NAME_BUFF );
;  1517     1505  2 USER_NAME_DESC[LENGTH]  = MIN ( .TOKENLEN , KGB$S_NAME ) ;
;  1518     1506  2 USER_NAME_DESC[POINTER] = USER_NAME_BUFF ;
;  1519     1507  2 HOLDER_VEC[1] = 0 ;
;  1520     1508  2
;  1521     1509  2 !
;  1522     1510  2 ! Now finish up with the ID name
;  1523     1511  2 ! Convert the ID from ASCII to a longword format
;  1524     1512  2 !
;  1525   P 1513  2 STATUS = $ASCTOID ( NAME    = ID_NAME_DESC,
;  1526   P 1514  2                     ID      = IDENT,
;  1527     1515  2                     ATTRIB = 0 ) ;
;  1528     1516  2 IF NOT .STATUS
;  1529     1517  2     THEN
;  1530     1518  3         BEGIN
;  1531     1519  3         LIB$SIGNAL(UAF$_GRANTERR, 2, ID_NAME_DESC, USER_NAME_DESC, .STATUS);
;  1532     1520  3         RETURN ;
;  1533     1521  2         END ;
;  1534     1522  2
;  1535     1523  2 IF .IDENT[UIC$V_FORMAT] NEQ UIC$K_ID_FORMAT
;  1536     1524  2    THEN
;  1537     1525  3         BEGIN
;  1538     1526  3         LIB$SIGNAL(UAF$_NOTIDFMT);
;  1539     1527  3         RETURN ;
;  1540     1528  2         END ;
;  1541     1529  2
;  1542     1530  2 IF (.IDENT LSSU UIC$K_FIRST_ID) OR
;  1543     1531  3    (.IDENT GTRU UIC$K_LAST_ID)
;  1544     1532  2     THEN
;  1545     1533  3         BEGIN
;  1546     1534  3         LIB$SIGNAL(UAF$_IDOUTRNG);
;  1547     1535  3         RETURN ;
;  1548     1536  2         END ;
;  1549     1537  2
;  1550     1538  2 !
;  1551     1539  2 ! Now finish parsing the user ID entered
;  1552     1540  2 ! We will use the rights data base to translate the name
;  1553     1541  2 ! unless it was given in UIC format.  If it is a UIC then
;  1554     1542  2 ! we will build a holder and make sure that it is in the rights
;  1555     1543  2 ! data base.
;  1556     1544  2 !
;  1557     1545  2 !
;  1558     1546  2 IF .UIC_FLAG
;  1559     1547  2     THEN
;  1560     1548  3         BEGIN
;  1561     1549  3         LOCAL
;  1562     1550  3             TEMP_NAME_DESC      : STATDESC,
;  1563     1551  3             TEMP_NAME_BUFF      : VECTOR [ KGB$S_NAME, BYTE ] ;
```

```
;  1564          1552  3          IF NOT PARSE_UIC ( HOLDER[UIC$V_GROUP],
;  1565          1553  3                             HOLDER[UIC$V_MEMBER],
;  1566          1554  3                             TRUE )
;  1567          1555  3              THEN
;  1568          1556  4                  BEGIN
;  1569          1557  4                  LIB$SIGNAL(UAF$_UICERR, 2, .TOKENLEN, .TOKENPTR);
;  1570          1558  4                  RETURN ;
;  1571          1559  3                  END ;
;  1572          1560  3          HOLDER[UIC$V_FORMAT] = UIC$K_UIC_FORMAT ;
;  1573          1561  3          !
;  1574          1562  3          ! Make sure that the holder exists and get it's ASCII name
;  1575          1563  3          !
;  1576          1564  3          TEMP_NAME_DESC[LENGTH]  = KGB$S_NAME ;
;  1577          1565  3          TEMP_NAME_DESC[POINTER] = TEMP_NAME_BUFF ;
;  1578        P 1566  3          STATUS = $IDTOASC ( ID     = .HOLDER_VEC[0],
;  1579        P 1567  3                              NAMLEN = TEMP_NAME_DESC[LENGTH],
;  1580          1568  3                              NAMBUF = TEMP_NAME_DESC ) ;
;  1581          1569  3          IF NOT .STATUS
;  1582          1570  3              THEN
;  1583          1571  4                  BEGIN
;  1584          1572  4                  LIB$SIGNAL(UAF$_GRANTERR, 2, ID_NAME_DESC, USER_NAME_DESC, .STATUS);
;  1585          1573  4                  RETURN ;
;  1586          1574  4                  END
;  1587          1575  3              ELSE
;  1588          1576  4                  BEGIN
;  1589          1577  4                  CH$COPY (.TEMP_NAME_DESC[LENGTH], .TEMP_NAME_DESC[POINTER],
;  1590          1578  4                           %C' ', .USER_NAME_DESC[LENGTH], .USER_NAME_DESC[POINTER] ) ;
;  1591          1579  4                  USER_NAME_DESC[LENGTH] = .TEMP_NAME_DESC[LENGTH] ;
;  1592          1580  3                  END ;
;  1593          1581  3          END
;  1594          1582  2      ELSE
;  1595          1583  3          BEGIN
;  1596          1584  3          IF .STR_WILD
;  1597          1585  3              THEN
;  1598          1586  4                  BEGIN
;  1599          1587  4                  LIB$SIGNAL(UAF$_WLDNOTALWD);
;  1600          1588  4                  RETURN ;
;  1601          1589  3                  END ;
;  1602          1590  3
;  1603          1591  3          !
;  1604          1592  3          ! Convert the user ID from ASCII to a longword format
;  1605          1593  3          !
;  1606        P 1594  3          STATUS = $ASCTOID ( NAME    = USER_NAME_DESC,
;  1607        P 1595  3                              ID      = HOLDER_VEC[0],
;  1608          1596  3                              ATTRIB = 0 ) ;
;  1609          1597  3          IF NOT .STATUS
;  1610          1598  3              THEN
;  1611          1599  4                  BEGIN
;  1612          1600  4                  LIB$SIGNAL(UAF$_GRANTERR, 2, ID_NAME_DESC, USER_NAME_DESC, .STATUS);
;  1613          1601  4                  RETURN ;
;  1614          1602  3                  END ;
;  1615          1603  3
;  1616          1604  3          IF .HOLDER[UIC$V_FORMAT] NEQ UIC$K_UIC_FORMAT
;  1617          1605  3             THEN
;  1618          1606  4                  BEGIN
;  1619          1607  4                  LIB$SIGNAL(UAF$_NOTUICFMT,
;  1620          1608  4                  RETURN ;
```

```
;  1621          1609  3                  END ;
;  1622          1610  3
;  1623          1611  2          END ;
;  1624          1612  2
;  1625          1613  2
;  1626          1614  2 !
;  1627          1615  2 ! Set the resource attribute if /RESOURCE was specified.
;  1628          1616  2 !
;  1629          1617  2 IF CLI$PRESENT (SD_ATTRIBRESOURCE)
;  1630          1618  2     THEN ATTRIBUTES = KGB$M_RESOURCE
;  1631          1619  2     ELSE ATTRIBUTES = 0 ;
;  1632          1620  2
;  1633          1621  2 !
;  1634          1622  2 ! Now call the service to perform the addition
;  1635          1623  2 !
;  1636        P 1624  2 STATUS = $ADD_HOLDER ( ID     = .IDENT,
;  1637        P 1625  2                        HOLDER = HOLDER ,
;  1638          1626  2                        ATTRIB = .ATTRIBUTES ) ;
;  1639          1627  2
;  1640          1628  2 IF NOT .STATUS
;  1641          1629  2     THEN LIB$SIGNAL(UAF$_GRANTERR, 2, ID_NAME_DESC, USER_NAME_DESC, .STATUS)
;  1642          1630  2     ELSE
;  1643          1631  3         BEGIN
;  1644          1632  3         RIGHTSLIST_MODIFIED = TRUE ;
;  1645          1633  3         LIB$SIGNAL(UAF$_GRANTMSG, 2, ID_NAME_DESC, USER_NAME_DESC);
;  1646          1634  2         END ;
;  1647          1635  2
;  1648          1636  2 RETURN ;
;  1649          1637  2
;  1650          1638  1 END ;                                   !End of UAF$GRANT_IDENT


                                                HOLDER_VEC=         HOLDER
                                                            .EXTRN  SYS$ADD_HOLDER

                                                    0FFC 00000      .ENTRY  UAF$GRANT_IDENT, Save R2,R3,R4,R5,R6,R7,R8,-  ;  1440
                                                                            R9,R10,R11
                                                5B 00000000G   00   9E 00002    MOVAB   LIB$SIGNAL, R11                    ;
                                                5A 00000000G   00   9E 00009    MOVAB   TOKENPTR, R10                      ;
                                                59 00000000G   00   9E 00010    MOVAB   TOKENLEN, R9                       ;
                                                58 00000000'   00   9E 00017    MOVAB   STATUS, R8                         ;
                                                57 00000000'   00   9E 0001E    MOVAB   HOLDER, R7                         ;
                                                5E       88    AE   9E 00025    MOVAB   -120(SP), SP                       ;
                                             70 AE 010E0000    8F   D0 00029    MOVL    #17694720, ID_NAME_DESC            ;  1478
                                                         74    AE   D4 00031    CLRL    ID_NAME_DESC+4                     ;
                                             48 AE 010E0000    8F   D0 00034    MOVL    #17694720, USER_NAME_DESC          ;  1480
                                                         4C    AE   D4 0003C    CLRL    USER_NAME_DESC+4                   ;
                                                00000000G   00   9F 0003F    PUSHAB  TOKENDSC                           ;  1491
                                                       0180  C8   9F 00045    PUSHAB  SD_TOKEN1                          ;
                                          00000000G 00             02   FB 00049    CALLS   #2, CLI$GET_VALUE          ;
                                                    56             69   3C 00050    MOVZWL  TOKENLEN, R6               ;  1493
                                                    50             6A   D0 00053    MOVL    TOKENPTR, R0               ;
                                       20       20  60             56   2C 00056    MOVC5   R6, (R0), #32, #32, ID_NAME_BUFF   ;
                                                          50       AE        0005B
                                                    50             56   D0 0005D    MOVL    R6, R0                     ;  1494
                                                    20             50   B1 00060    CMPW    R0, #32                    ;
```

```
                                03 1B 00063          BLEQU   1$
                        50      20 D0 00065          MOVL    #32, R0
                  70    AE      50 B0 00068 1$:      MOVW    R0, ID_NAME_DESC
                  74    AE  50  AE 9E 0006C          MOVAB   ID_NAME_BUFF, ID_NAME_DESC+4                  ; 1495
                                01 DD 00071          PUSHL   #1                                            ; 1503
                          0188  C8 9F 00073          PUSHAB  SD_TOKEN2
             00000000G  00      02 FB 00077          CALLS   #2, PARSE_WILD
                        56      69 3C 0007E          MOVZWL  TOKENLEN, R6                                  ; 1504
                        50      6A D0 00081          MOVL    TOKENPTR, R0
20          20          60      56 2C 00084          MOVC5   R6, (R0), #32, #32, USER_NAME_BUFF
                            28  AE    00089
                        50      56 D0 0008B          MOVL    R6, R0                                        ; 1505
                        20      50 B1 0008E          CMPW    R0, #32
                                03 1B 00091          BLEQU   2$
                        50      20 D0 00093          MOVL    #32, R0
                  48    AE      50 B0 00096 2$:      MOVW    R0, USER_NAME_DESC
                  4C    AE  28  AE 9E 0009A          MOVAB   USER_NAME_BUFF, USER_NAME_DESC+4              ; 1506
                            04  A7 D4 0009F          CLRL    HOLDER_VEC+4                                  ; 1507
                                7E D4 000A2          CLRL    -(SP)                                         ; 1515
                            08  A7 9F 000A4          PUSHAB  IDENT
                            78  AE 9F 000A7          PUSHAB  ID_NAME_DESC
             00000000G  00      03 FB 000AA          CALLS   #3, SYS$ASCTOID
                        68      50 D0 000B1          MOVL    R0, STATUS
                        03      50 E8 000B4          BLBS    R0, 3$                                        ; 1516
                          00F8  31    000B7          BRW     16$
02       0B     A7      02      06 ED 000BA 3$:      CMPZV   #6, #2, IDENT+3, #2                           ; 1523
                                08 13 000C0          BEQL    4$
                     00000000G  8F DD 000C2          PUSHL   #UAF$_NOTIDFMT                                ; 1526
                                1C 11 000C8          BRB     6$
                        50  08  A7 D0 000CA 4$:      MOVL    IDENT, R0                                     ; 1530
             80010000   8F      50 D1 000CE          CMPL    R0, #-2147418112
                                09 1F 000D5          BLSSU   5$
             8FFFFFFF   8F      50 D1 000D7          CMPL    R0, #-1879048193                              ; 1531
                                08 1B 000DE          BLEQU   7$
                     00000000G  8F DD 000E0 5$:      PUSHL   #UAF$_IDOUTRNG                                ; 1534
                                77 11 000E6 6$:      BRB     10$
                   63 00000000G 00 E9 000E8 7$:      BLBC    UIC_FLAG, 9$                                  ; 1546
                  20    AE 010E0000 8F D0 000EF      MOVL    #17694720, TEMP_NAME_DESC                     ; 1550
                            24  AE D4 000F7          CLRL    TEMP_NAME_DESC+4
                                01 DD 000FA          PUSHL   #1                                            ; 1553
                                57 DD 000FC          PUSHL   R7
                            02  A7 9F 000FE          PUSHAB  HOLDER+2                                      ; 1552
             00000000G  00      03 FB 00101          CALLS   #3, PARSE_UIC                                 ; 1553
                        10      50 E8 00108          BLBS    R0, 8$
                                6A DD 0010B          PUSHL   TOKENPTR                                      ; 1557
                        7E      69 3C 0010D          MOVZWL  TOKENLEN, -(SP)
                                02 DD 00110          PUSHL   #2
                     00000000G  8F DD 00112          PUSHL   #UAF$_UICERR
                          00C0  31    00118          BRW     18$
                  03    A7  C0  8F 8A 0011B 8$:      BICB2   #192, HOLDER+3                                ; 1560
                  20    AE      20 B0 00120          MOVW    #32, TEMP_NAME_DESC                           ; 1564
                  24    AE      6E 9E 00124          MOVAB   TEMP_NAME_BUFF, TEMP_NAME_DESC+4              ; 1565
                                7E 7C 00128          CLRQ    -(SP)                                         ; 1568
                                7E D4 0012A          CLRL    -(SP)
                            2C  AE 9F 0012C          PUSHAB  TEMP_NAME_DESC
                            30  AE 9F 0012F          PUSHAB  TEMP_NAME_DESC
                                67 DD 00132          PUSHL   HOLDER_VEC
```

```
                    00000000G  00              06  FB 00134          CALLS   #6, SYS$IDTOASC                                  ;
                               68              50  D0 0013B          MOVL    R0, STATUS                                      ;
                               71              50  E9 0013E          BLBC    R0, 16$                                         ; 1569
      48  AE          20   24   BE      20     AE  2C 00141          MOVC5   TEMP_NAME_DESC, @TEMP_NAME_DESC+4, #32, -       ; 1578
                                           4C  BE      00149                  USER_NAME_DESC, @USER_NAME_DESC+4               ;
                           48  AE      20     AE  B0 0014B          MOVW    TEMP_NAME_DESC, USER_NAME_DESC                  ; 1579
                                           34  11 00150          BRB     13$                                             ; 1546
                               08 00000000G   00  E9 00152 9$:      BLBC    STR_WILD, 11$                                   ; 1584
                                  00000000G   8F  DD 00159          PUSHL   #UAF$_WLDNOTALWD                                ; 1587
                                           21  11 0015F 10$:      BRB     12$                                             ;
                                           7E  D4 00161 11$:      CLRL    -(SP)                                           ; 1596
                                           57  DD 00163          PUSHL   R7                                              ;
                                      50   AE  9F 00165          PUSHAB  USER_NAME_DESC                                  ;
                    00000000G  00          03  FB 00168          CALLS   #3, SYS$ASCTOID                                 ;
                               68          50  D0 0016F          MOVL    R0, STATUS                                      ;
                               3D          50  E9 00172          BLBC    R0, 16$                                         ; 1597
                           C0  8F      03  A7  93 00175          BITB    HOLDER+3, #192                                  ; 1604
                                           0A  13 0017A          BEQL    13$                                             ;
                                  00000000G  8F  DD 0017C          PUSHL   #UAF$_NOTUICFMT                                 ; 1607
                               6B             01  FB 00182 12$:      CALLS   #1, LIB$SIGNAL                                  ;
                                              04  00185          RET                                                     ; 1606
                                    0150   C8  9F 00186 13$:      PUSHAB  SD_ATTRIBRESOURCE                               ; 1617
                    00000000G  00          01  FB 0018A          CALLS   #1, CLI$PRESENT                                 ;
                               06          50  E9 00191          BLBC    R0, 14$                                         ;
                           FC  A7          01  D0 00194          MOVL    #1, ATTRIBUTES                                  ; 1618
                                           03  11 00198          BRB     15$                                             ;
                                    FC     A7  D4 0019A 14$:      CLRL    ATTRIBUTES                                      ; 1619
                                    FC     A7  DD 0019D 15$:      PUSHL   ATTRIBUTES                                      ; 1626
                                           57  DD 001A0          PUSHL   R7                                              ;
                                    08     A7  DD 001A2          PUSHL   IDENT                                           ;
                    00000000G  00          03  FB 001A5          CALLS   #3, SYS$ADD_HOLDER                              ;
                               68          50  D0 001AC          MOVL    R0, STATUS                                      ;
                               14          50  E8 001AF          BLBS    R0, 17$                                         ; 1628
                                           50  DD 001B2 16$:      PUSHL   R0                                              ; 1629
                                    4C     AE  9F 001B4          PUSHAB  USER_NAME_DESC                                  ;
                                    78     AE  9F 001B7          PUSHAB  ID_NAME_DESC                                    ;
                                           02  DD 001BA          PUSHL   #2                                              ;
                                  00000000G  8F  DD 001BC          PUSHL   #UAF$_GRANTERR                                  ;
                               6B             05  FB 001C2          CALLS   #5, LIB$SIGNAL                                  ;
                                              04  001C5          RET                                                     ;
                    00000000G  00          01  90 001C6 17$:      MOVB    #1, RIGHTSLIST_MODIFIED                         ; 1632
                                    48     AE  9F 001CD          PUSHAB  USER_NAME_DESC                                  ; 1633
                                    74     AE  9F 001D0          PUSHAB  ID_NAME_DESC                                    ;
                                           02  DD 001D3          PUSHL   #2                                              ;
                                  00000000G  8F  DD 001D5          PUSHL   #UAF$_GRANTMSG                                  ;
                               6B             04  FB 001DB 18$:      CALLS   #4, LIB$SIGNAL                                  ;
                                              04  001DE          RET                                                     ; 1638

; Routine Size:  479 bytes,    Routine Base:  $CODE$ + 09C5

; 1651          1639  1
```

```
;  1653          1640  1 %SBTTL ' UAF$RECORD_FOUND - Flag that the group is not empty'
;  1654          1641  1
;  1655          1642  1 GLOBAL ROUTINE UAF$RECORD_FOUND =
;  1656          1643  2 BEGIN
;  1657          1644  2 !++
;  1658          1645  2 !
;  1659          1646  2 ! FUNCTIONAL DESCRIPTION:
;  1660          1647  2 !
;  1661          1648  2 !       Flag that a particular group is not empty.  This is used to
;  1662          1649  2 !       determine if a group identifier should be removed.
;  1663          1650  2 !
;  1664          1651  2 !
;  1665          1652  2 ! INPUTS:
;  1666          1653  2 !
;  1667          1654  2 !       none
;  1668          1655  2 !
;  1669          1656  2 ! IMPLICIT INPUTS:
;  1670          1657  2 !
;  1671          1658  2 !       none
;  1672          1659  2 !
;  1673          1660  2 ! OUTPUTS:
;  1674          1661  2 !
;  1675          1662  2 !       None
;  1676          1663  2 !
;  1677          1664  2 ! IMPLICIT OUTPUTS:
;  1678          1665  2 !
;  1679          1666  2 !       FOUND_MATCH flag is set TRUE
;  1680          1667  2 !
;  1681          1668  2 ! ROUTINE VALUE:
;  1682          1669  2 !
;  1683          1670  2 !       Always false so that the WILD_USER loop will be exited after first match
;  1684          1671  2 !
;  1685          1672  2 ! SIDE EFFECTS:
;  1686          1673  2 !
;  1687          1674  2 !       none
;  1688          1675  2 !--
;  1689          1676  2
;  1690          1677  2
;  1691          1678  2 FOUND_MATCH = TRUE;
;  1692          1679  2
;  1693          1680  2 RETURN FALSE ;
;  1694          1681  2
;  1695          1682  1 END ;                                   ! End of UAF$RECORD_FOUND



                                                  0000 00000        .ENTRY  UAF$RECORD_FOUND, Save nothing          ; 1642
                      00000000G  00             01  D0 00002        MOVL    #1, FOUND_MATCH                         ; 1678
                                                50  D4 00009        CLRL    R0                                      ; 1680
                                                    04 0000B        RET                                             ; 1682

; Routine Size:  12 bytes,    Routine Base:  $CODE$ + 0BA4

;  1696          1683  1
```

```
;1698    1684  1 %SBTTL ' UAF$LIST_IDENT - Display rights identifiers '
;1699    1685  1
;1700    1686  1 GLOBAL ROUTINE UAF$LIST_IDENT : NOVALUE =
;1701    1687  2 BEGIN
;1702    1688  2 !++
;1703    1689  2 !
;1704    1690  2 ! FUNCTIONAL DESCRIPTION:
;1705    1691  2 !
;1706    1692  2 !       Write all identifiers to a listing file.  This routine is invoked
;1707    1693  2 !       via the LIST/IDENT command
;1708    1694  2 !
;1709    1695  2 ! INPUTS:
;1710    1696  2 !
;1711    1697  2 !       none
;1712    1698  2 !
;1713    1699  2 ! IMPLICIT INPUTS:
;1714    1700  2 !
;1715    1701  2 !       command line
;1716    1702  2 !
;1717    1703  2 ! OUTPUTS:
;1718    1704  2 !
;1719    1705  2 !       None
;1720    1706  2 !
;1721    1707  2 ! IMPLICIT OUTPUTS:
;1722    1708  2 !                                                                       ;
;1723    1709  2 !       none
;1724    1710  2 !
;1725    1711  2 ! ROUTINE VALUE:                                                       ;
;1726    1712  2 !
;1727    1713  2 !       none
;1728    1714  2 !
;1729    1715  2 ! SIDE EFFECTS:
;1730    1716  2 !
;1731    1717  2 !       A listing file is created
;1732    1718  2 !
;1733    1719  2 !--
;1734    1720  2
;1735    1721  2
;1736    1722  2 !
;1737    1723  2 ! Create the listing file
;1738    1724  2 !
;1739    1725  2 RLSTFAB [FAB$V_DLT] = FALSE ;
;1740    1726  3 IF RMSBAD ($CREATE (FAB = RLSTFAB))
;1741    1727  2     THEN
;1742    1728  3         BEGIN
;1743    1729  3         LIB$SIGNAL(UAF$_LSTERR, 0, .RMSERR);
;1744    1730  3         RETURN ;
;1745    1731  2         END ;
;1746    1732  2
;1747    1733  3 IF RMSBAD ( $CONNECT ( RAB = RLSTRAB))
;1748    1734  2     THEN
;1749    1735  3         BEGIN
;1750    1736  3         LIB$SIGNAL(UAF$_LSTERR, 0, .RMSERR);
;1751    1737  3         RETURN ;
;1752    1738  2         END ;
;1753    1739  2
;1754    1740  2 !
```

```
;  17>5          1741  2 ! Set the correct RAB for the output and flag a list operation
;  1756          1742  2 !
;  1757          1743  2 RABPTR = RLSTRAB ;
;  1758          1744  2 RDB_LIST_FLAG = TRUE ;
;  1759          1745  2
;  1760          1746  2 !
;  1761          1747  2 ! Set the type of display.
;  1762          1748  2 !
;  1763          1749  3 IF CLI$PRESENT (SD_BRIEF) OR (NOT CLI$PRESENT (SD_FULL))
;  1764          1750  2     THEN SHOW_ID_FULL = FALSE
;  1765          1751  2     ELSE SHOW_ID_FULL = TRUE ;
;  1766          1752  2
;  1767          1753  2 !
;  1768          1754  2 ! Now call the common routine for show or list
;  1769          1755  2 !
;  1770          1756  2 UAF$DISPLAY_IDENT () ;
;  1771          1757  2
;  1772          1758  2 $DISCONNECT ( RAB = RLSTRAB ) ;
;  1773          1759  2 $CLOSE ( FAB = RLSTFAB ) ;
;  1774          1760  2
;  1775          1761  2 RETURN
;  1776          1762  2
;  1777          1763  1 END ;                                   ! End of UAF$LIST_IDENT


                                                        .EXTRN  SYS$CREATE, SYS$CONNECT
                                                        .EXTRN  SYS$DISCONNECT, SYS$CLOSE

                                  001C 00000            .ENTRY  UAF$LIST_IDENT, Save R2,R3,R4          ; 1686
                        54 00000000G   00   9E 00002    MOVAB   CLI$PRESENT, R4
                        53 00000000G   00   9E 00009    MOVAB   RMSERR, R3
                        52 00000000'   00   9E 00010    MOVAB   RLSTRAB, R2
                  B5    A2        80   8F   8A 00017    BICB2   #128, RLSTFAB+5                        ; 1725
                            B0         A2   9F 0001C    PUSHAB  RLSTFAB                                ; 1726
00000000G         00                   01   FB 0001F    CALLS   #1, SYS$CREATE
                  63                   50   D0 00026    MOVL    R0, RMSERR
                  0F                   50   E9 00029    BLBC    R0, 1$
                                       52   DD 0002C    PUSHL   R2                                     ; 1733
00000000G         00                   01   FB 0002E    CALLS   #1, SYS$CONNECT
                  63                   50   D0 00035    MOVL    R0, RMSERR
                  12                   50   E8 00038    BLBS    R0, 2$
                                       63   DD 0003B 1$: PUSHL   RMSERR                                 ; 1736
                                       7E   D4 0003D    CLRL    -(SP)
                        00000000G      8F   DD 0003F    PUSHL   #UAF$_LSTERR
00000000G         00                   03   FB 00045    CALLS   #3, LIB$SIGNAL
                                       04   0004C    RET                                            ; 1735
00000000G         00                   62   9E 0004D 2$: MOVAB   RLSTRAB, RABPTR                       ; 1743
00000000'         00                   01   90 00054    MOVB    #1, RDB_LIST_FLAG                      ; 1744
                            4C         A2   9F 0005B    PUSHAB  SD_BRIEF                               ; 1749
                  64                   01   FB 0005E    CALLS   #1, CLI$PRESENT
                  09                   50   E8 00061    BLBS    R0, 3$
                            54         A2   9F 00064    PUSHAB  SD_FULL
                  64                   01   FB 00067    CALLS   #1, CLI$PRESENT
                  06                   50   E8 0006A    BLBS    R0, 4$
                               FEF0    C2   94 0006D 3$: CLRB    SHOW_ID_FULL                           ; 1750
                                       05   11 00071    BRB     5$
```

```
                     FEF0   C2              01  90 00073 4$:      MOVB    #1, SHOW_ID_FULL                        ; 1751
                     F8EC   CF              00  FB 00078 5$:      CALLS   #0, UAF$DISPLAY_IDENT                   ; 1756
                                            52  DD 0007D          PUSHL   R2                                      ; 1758
                00000000G   00              01  FB 0007F          CALLS   #1, SYS$DISCONNECT                      ;
                                       B0   A2  9F 00086          PUSHAB  RLSTFAB                                 ; 1759
                00000000G   00              01  FB 00089          CALLS   #1, SYS$CLOSE                           ;
                                                04 00090          RET                                             ; 1763

; Routine Size:  145 bytes,    Routine Base:  $CODE$ + 0BB0

; 1778          1764  1
```

```
;1780      1765  1 %SBTTL ' UAF$LIST_RIGHTS - Display rights held by user'
;1781      1766  1
;1782      1767  1 GLOBAL ROUTINE UAF$LIST_RIGHTS : NOVALUE =
;1783      1768  2 BEGIN
;1784      1769  2 !++
;1785      1770  2 !
;1786      1771  2 ! FUNCTIONAL DESCRIPTION:
;1787      1772  2 !
;1788      1773  2 !       Write all identifiers held by a specified identifier
;1789      1774  2 !       to a listing file.  This routine is invoked
;1790      1775  2 !       via the LIST/RIGHTS command
;1791      1776  2 !
;1792      1777  2 ! INPUTS:
;1793      1778  2 !
;1794      1779  2 !       none
;1795      1780  2 !
;1796      1781  2 ! IMPLICIT INPUTS:
;1797      1782  2 !
;1798      1783  2 !       command line
;1799      1784  2 !
;1800      1785  2 ! OUTPUTS:
;1801      1786  2 !
;1802      1787  2 !       None
;1803      1788  2 !
;1804      1789  2 ! IMPLICIT OUTPUTS:
;1805      1790  2 !
;1806      1791  2 !       none
;1807      1792  2 !
;1808      1793  2 ! ROUTINE VALUE:
;1809      1794  2 !
;1810      1795  2 !       none
;1811      1796  2 !
;1812      1797  2 ! SIDE EFFECTS:
;1813      1798  2 !
;1814      1799  2 !       A listing file is created
;1815      1800  2 !
;1816      1801  2 !--
;1817      1802  2
;1818      1803  2
;1819      1804  2 !
;1820      1805  2 ! Create the listing file
;1821      1806  2 !
;1822      1807  2 RLSTFAB [FAB$V_DLT] = FALSE ;
;1823      1808  3 IF RMSBAD ($CREATE (FAB = RLSTFAB))
;1824      1809  2     THEN
;1825      1810  3         BEGIN
;1826      1811  3         LIB$SIGNAL(UAF$_LSTERR, 0, .RMSERR);
;1827      1812  3         RETURN ;
;1828      1813  2         END ;
;1829      1814  2
;1830      1815  3 IF RMSBAD ( $CONNECT ( RAB = RLSTRAB))
;1831      1816  2     THEN
;1832      1817  3         BEGIN
;1833      1818  3         LIB$SIGNAL(UAF$_LSTERR, 0, .RMSERR);
;1834      1819  3         RETURN ;
;1835      1820  2         END ;
;1836      1821  2
```

```
;  1837          1822  2 !
;  1838          1823  2 ! Select the output RAB and flag a list operation
;  1839          1824  2 !
;  1840          1825  2 RABPTR = RLSTRAB ;
;  1841          1826  2 RDB_LIST_FLAG = TRUE ;
;  1842          1827  2
;  1843          1828  2 !
;  1844          1829  2 ! Call common routine to perform show or list
;  1845          1830  2 !
;  1846          1831  2 UAF$DISPLAY_RIGHTS () ;
;  1847          1832  2
;  1848          1833  2 $DISCONNECT ( RAB = RLSTRAB ) ;
;  1849          1834  2 $CLOSE ( FAB = RLSTFAB ) ;
;  1850          1835  2
;  1851          1836  2 RETURN ;
;  1852          1837  2
;  1853          1838  1 END ;                                    !End of UAF$LIST_RIGHTS



                                                    000C 00000          .ENTRY  UAF$LIST_RIGHTS, Save R2,R3                     ; 1767
                                 53 00000000G   00  9E 00002          MOVAB   RMSERR, R3                                      ;
                                 52 00000000'   00  9E 00009          MOVAB   RLSTRAB, R2                                     ;
                           B5    A2        80    8F  8A 00010          BICB2   #128, RLSTFAB+5                                 ; 1807
                                 B0    A2    9F  9F 00015          PUSHAB  RLSTFAB                                         ; 1808
                    00000000G    00          01  FB 00018          CALLS   #1, SYS$CREATE                                  ;
                                 63          50  D0 0001F          MOVL    R0, RMSERR                                      ;
                                 0F          50  E9 00022          BLBC    R0, 1$                                          ;
                                             52  DD 00025          PUSHL   R2                                              ; 1815
                    00000000G    00          01  FB 00027          CALLS   #1, SYS$CONNECT                                 ;
                                 63          50  D0 0002E          MOVL    R0, RMSERR                                      ;
                                 12          50  E8 00031          BLBS    R0, 2$                                          ;
                                             63  DD 00034  1$:      PUSHL   RMSERR                                          ; 1818
                                             7E  D4 00036          CLRL    -(SP)                                           ;
                                 00000000G   8F  DD 00038          PUSHL   #UAF$_LSTERR                                    ;
                    00000000G    00          03  FB 0003E          CALLS   #3, LIB$SIGNAL                                  ;
                                             04  00045          RET                                                     ; 1817
                    00000000G    00          62  9E 00046  2$:      MOVAB   RLSTRAB, RABPTR                                 ; 1825
                    00000000'    00          01  90 0004D          MOVB    #1, RDB_LIST_FLAG                               ; 1826
                         FB83    CF          00  FB 00054          CALLS   #0, UAF$DISPLAY_RIGHTS                          ; 1831
                                             52  DD 00059          PUSHL   R2                                              ; 1833
                    00000000G    00          01  FB 0005B          CALLS   #1, SYS$DISCONNECT                              ;
                                       B0    A2  9F 00062          PUSHAB  RLSTFAB                                         ; 1834
                    00000000G    00          01  FB 00065          CALLS   #1, SYS$CLOSE                                   ;
                                             04  0006C          RET                                                     ; 1838

; Routine Size:  109 bytes,    Routine Base:  $CODE$ + 0C41

;  1854          1839  1
```

```
;  1856          1840  1 %SBTTL ' UAF$MODIFY_IDENT  -  Modify RDB record'
;  1857          1841  1
;  1858          1842  1 GLOBAL ROUTINE UAF$MODIFY_IDENT : NOVALUE =
;  1859          1843  2 BEGIN
;  1860          1844  2 !++
;  1861          1845  2 !
;  1862          1846  2 ! FUNCTIONAL DESCRIPTION:
;  1863          1847  2 !
;  1864          1848  2 !       Modify a rights data base record.  This routine is invoked
;  1865          1849  2 !       via the MODIFY/IDENT command.
;  1866          1850  2 !
;  1867          1851  2 ! INPUTS:
;  1868          1852  2 !
;  1869          1853  2 !       none
;  1870          1854  2 !
;  1871          1855  2 ! IMPLICIT INPUTS:
;  1872          1856  2 !
;  1873          1857  2 !       command line
;  1874          1858  2 !
;  1875          1859  2 ! OUTPUTS:
;  1876          1860  2 !
;  1877          1861  2 !       None
;  1878          1862  2 !
;  1879          1863  2 ! IMPLICIT OUTPUTS:
;  1880          1864  2 !
;  1881          1865  2 !       none
;  1882          1866  2 !
;  1883          1867  2 ! ROUTINE VALUE:
;  1884          1868  2 !
;  1885          1869  2 !       none
;  1886          1870  2 !
;  1887          1871  2 ! SIDE EFFECTS:
;  1888          1872  2 !
;  1889          1873  2 !       Rights data base is modified.
;  1890          1874  2 !
;  1891          1875  2 !--
;  1892          1876  2
;  1893          1877  2
;  1894          1878  2 LOCAL
;  1895          1879  2     NAME_DESC           : STATDESC,
;  1896          1880  2     NAME_BUFF           : VECTOR [ KGB$S_NAME, BYTE ] ,
;  1897          1881  2     NEW_VALUE           : $BBLOCK [4],
;  1898          1882  2     NEW_NAME_DESC       : STATDESC,
;  1899          1883  2     NEW_NAME_BUFF       : VECTOR [ KGB$S_NAME, BYTE ] ,
;  1900          1884  2     NEW_NAME_PTR        : LONG ,
;  1901          1885  2     OLD_IDENT           : $BBLOCK [4],
;  1902          1886  2     CLR_ATTRIBUTES      : LONG INITIAL (0) ,
;  1903          1887  2     SET_ATTRIBUTES      : LONG INITIAL (0) ;
;  1904          1888  2
;  1905          1889  2
;  1906          1890  2 !
;  1907          1891  2 ! First we will get the identifier name.
;  1908          1892  2 ! This is a required parameter so it is guaranteed present
;  1909          1893  2 !
;  1910          1894  2 CLI$GET_VALUE ( SD_TOKEN1, TOKENDSC ) ;
;  1911          1895  2
;  1912          1896  2 CH$COPY ( .TOKENLEN, .TOKENPTR, %C' ', KGB$S_NAME,  NAME_BUFF );
```

```
;1913     1897  2 NAME_DESC[LENGTH]  = MIN ( .TOKENLEN , KGB$S_NAME ) ;
;1914     1898  2 NAME_DESC[POINTER] = NAME_BUFF ;
;1915     1899  2
;1916     1900  2 !
;1917     1901  2 ! Convert the ID from ASCII to a longword format
;1918     1902  2 !
;1919   P 1903  2 STATUS = $ASCTOID ( NAME    = NAME_DESC,
;1920   P 1904  2                     ID      = OLD_IDENT,
;1921     1905  2                     ATTRIB = 0 ) ;
;1922     1906  2 IF NOT .STATUS
;1923     1907  2     THEN
;1924     1908  3         BEGIN
;1925     1909  3         LIB$SIGNAL(UAF$_RDBMDFYERR, 1, NAME_DESC, .STATUS);
;1926     1910  3         RETURN ;
;1927     1911  2         END ;
;1928     1912  2
;1929     1913  2 !
;1930     1914  2 ! Now find out what to do with the status attribute.
;1931     1915  2 ! If it is explicitly negated in the command line then
;1932     1916  2 ! clear it.  If it is present but not negated then set it.
;1933     1917  2 ! If it is not specified then do nothing
;1934     1918  2 !
;1935     1919  2 STATUS = CLI$PRESENT ( SD_ATTRIBRESOURCE ) ;
;1936     1920  2 IF .STATUS EQL CLI$_NEGATED
;1937     1921  2     THEN CLR_ATTRIBUTES = KGB$M_RESOURCE
;1938     1922  2 ELSE IF .STATUS
;1939     1923  2     THEN SET_ATTRIBUTES = KGB$M_RESOURCE ;
;1940     1924  2
;1941     1925  2
;1942     1926  2 !
;1943     1927  2 ! Now get the new name if one was specified
;1944     1928  2 !
;1945     1929  2 IF CLI$GET_VALUE ( SD_NAME, TOKENDSC )
;1946     1930  2     THEN
;1947     1931  3         BEGIN
;1948     1932  3         NEW_NAME_PTR = NEW_NAME_DESC ;
;1949     1933  3         CH$COPY ( .TOKENLEN, .TOKENPTR, %C' ', KGB$S_NAME,  NEW_NAME_BUFF );
;1950     1934  3         NEW_NAME_DESC[LENGTH]  = MIN ( .TOKENLEN , KGB$S_NAME ) ;
;1951     1935  3         NEW_NAME_DESC[POINTER] = NEW_NAME_BUFF ;
;1952     1936  3         END
;1953     1937  2     ELSE NEW_NAME_PTR = 0 ;
;1954     1938  2
;1955     1939  2 !
;1956     1940  2 ! If a new ID was given then get it's value
;1957     1941  2 !
;1958     1942  2 IF NOT CLI$PRESENT ( SD_VALUE )
;1959     1943  2     THEN
;1960     1944  2         !
;1961     1945  2         ! No new identifier value was specified
;1962     1946  2         !
;1963     1947  2         NEW_VALUE = 0
;1964     1948  2
;1965     1949  2     ELSE
;1966     1950  3         BEGIN
;1967     1951  3         !
;1968     1952  3         ! /VALUE is there now determine what keyword was used
;1969     1953  3         ! and interpret the data as appropriate.
```

```
;1970    1954  3          ! First check for IDENTIFIER:
;1971    1955  3          IF CLI$GET_VALUE ( SD_VALUEIDENTIFIER, TOKENDSC )
;1972    1956  3              THEN
;1973    1957  4                  BEGIN
;1974    1958  4                  LOCAL
;1975    1959  4                      TMP_IDENT : LONG INITIAL (0) ;
;1976    1960  4
;1977    1961  4                  STATUS = GETVAL ( TMP_IDENT, LONGWORD_LENGTH );
;1978    1962  4                  IF NOT .STATUS
;1979    1963  4                      THEN
;1980    1964  5                          BEGIN
;1981    1965  5                          LIB$SIGNAL(UAF$_RDBMDFYERR, 1, NAME_DESC, .STATUS);
;1982    1966  5                          RETURN ;
;1983    1967  4                          END ;
;1984    1968  4                  NEW_VALUE [UIC$V_FORMAT]  = UIC$K_ID_FORMAT ;
;1985    1969  4                  NEW_VALUE [UIC$V_ID_CODE] = .TMP_IDENT ;
;1986    1970  4                  IF (.NEW_VALUE LSSU %X'80008000') OR
;1987    1971  5                     (.NEW_VALUE GTRU UIC$K_LAST_ID)
;1988    1972  4                      THEN
;1989    1973  5                          BEGIN
;1990    1974  5                          LIB$SIGNAL(UAF$_IDOUTRNG);
;1991    1975  5                          RETURN ;
;1992    1976  4                          END ;
;1993    1977  4                      END
;1994    1978  3              ELSE
;1995    1979  4                  BEGIN
;1996    1980  4                  !
;1997    1981  4                  ! Now check for UIC
;1998    1982  4                  ! These two keywords are synonyms
;1999    1983  4                  !
;2000    1984  4                  IF CLI$PRESENT (SD_VALUEUIC)
;2001    1985  4                      THEN IF NOT PARSE_WILD (SD_VALUEUIC, TRUE )
;2002    1986  4                          THEN
;2003    1987  5                              BEGIN
;2004    1988  5                              LIB$SIGNAL(UAF$_RDBMDFYERR, 1, NAME_DESC, .STATUS);
;2005    1989  5                              RETURN ;
;2006    1990  4                              END ;
;2007    1991  4                  IF .UIC_FLAG
;2008    1992  4                      THEN
;2009    1993  5                          BEGIN
;2010    1994  5                          IF .GRP_WILD AND NOT .MEM_WILD
;2011    1995  5                              THEN
;2012    1996  6                                  BEGIN
;2013    1997  6                                  !
;2014    1998  6                                  ! Wild cards are not allowed for group numbers
;2015    1999  6                                  !
;2016    2000  6                                  LIB$SIGNAL(UAF$_NOGRPWILD);
;2017    2001  6                                  RETURN ;
;2018    2002  5                                  END ;
;2019    2003  5                          !
;2020    2004  5                          ! The UIC format was used,  parse it to fine the
;2021    2005  5                          ! group and member number.
;2022    2006  5                          ! Wild card group and member numbers are allowed
;2023    2007  5                          !
;2024    2008  5                          IF NOT PARSE_UIC ( NEW_VALUE[UIC$V_GROUP],
;2025    2009  5                                             NEW_VALUE[UIC$V_MEMBER],
;2026    2010  5                                             TRUE )
```

```
;  2027     2011  5                        THEN
;  2028     2012  6                            BEGIN
;  2029     2013  6                            LIB$SIGNAL(UAF$_UICERR, 2, .TOKENLEN,
;  2030     2014  6                                                .TOKENPTR);
;  2031     2015  6                            RETURN ;
;  2032     2016  5                            END ;
;  2033     2017  5                        NEW_VALUE[UIC$V_FORMAT] = UIC$K_UIC_FORMAT ;
;  2034     2018  5                        END
;  2035     2019  4                    ELSE                                                    ;
;  2036     2020  5                        BEGIN
;  2037     2021  5                        !
;  2038     2022  5                        ! A user name was specified.  Get the UAF record                        ;
;  2039     2023  5                        ! for this user and pull the UIC out.
;  2040     2024  5                        !
;  2041     2025  5                        IF .STR_WILD
;  2042     2026  5                            THEN
;  2043     2027  6                                BEGIN
;  2044     2028  6                                !
;  2045     2029  6                                ! Wild cards are not allowed for user specs
;  2046     2030  6                                !
;  2047     2031  6                                LIB$SIGNAL(UAF$_WLDNOTALWD);
;  2048     2032  6                                RETURN ;
;  2049     2033  5                                END ;
;  2050     2034  5                        FOUND_MATCH = FALSE ;
;  2051     2035  6                        IF RMSBAD ( WILD_USER (UAF$FIND_UIC))
;  2052     2036  5                            THEN
;  2053     2037  6                                BEGIN
;  2054     2038  6                                IF .RMSERR EQL RMS$_RNF
;  2055     2039  6                                    THEN LIB$SIGNAL(UAF$_BADSPC)
;  2056     2040  6                                    ELSE LIB$SIGNAL(UAF$_RDBMDFYERR, 1,
;  2057     2041  6                                                    NAME_DESC, .RMSERR) ;
;  2058     2042  6                                RETURN ;
;  2059     2043  5                                END ;
;  2060     2044  5                        !
;  2061     2045  5                        ! UAF$FIND_UIC fills in IDENT
;  2062     2046  5                        !
;  2063     2047  5                        NEW_VALUE = .IDENT ;
;  2064     2048  4                        END ;
;  2065     2049  4
;  2066     2050  3                    END;
;  2067     2051  3
;  2068     2052  2            END ;
;  2069     2053  2
;  2070     2054  2 !
;  2071     2055  2 ! Finally find out if /HOLDER was specified.  If it was
;  2072     2056  2 ! then get the owners UAF record and pull the UIC out of it.
;  2073     2057  2 ! ALSO, if the qualifier was specified we will call the $MOD_HOLDER
;  2074     2058  2 ! service instead of the $MOD_IDENT service
;  2075     2059  2 !
;  2076     2060  2 IF CLI$PRESENT ( SD_HOLDER )
;  2077     2061  2     THEN
;  2078     2062  3         BEGIN
;  2079     2063  3         IF NOT PARSE_WILD (SD_HOLDER, FALSE )
;  2080     2064  3             THEN RETURN ;
;  2081     2065  3         IF .UIC_FLAG
;  2082     2066  3             THEN
;  2083     2067  4                 BEGIN
```

```
;  2084          2068  4                 IF .GRP_WILD AND NOT .MEM_WILD
;  2085          2069  4                     THEN
;  2086          2070  5                         BEGIN
;  2087          2071  5                         !
;  2088          2072  5                         ! Wild cards are not allowed for group numbers
;  2089          2073  5                         !
;  2090          2074  5                         LIB$SIGNAL(UAF$_NOGRPWILD);
;  2091          2075  5                         RETURN ;
;  2092          2076  4                         END ;
;  2093          2077  4                 !
;  2094          2078  4                 ! The UIC format was used,  parse it to fine the
;  2095          2079  4                 ! group and member number.
;  2096          2080  4                 ! Wild card group and member numbers are allowed
;  2097          2081  4                 !
;  2098          2082  4                 IF NOT PARSE_UIC ( IDENT[UIC$V_GROUP],
;  2099          2083  4                                    IDENT[UIC$V_MEMBER],
;  2100          2084  4                                    TRUE )
;  2101          2085  4                     THEN
;  2102          2086  5                         BEGIN
;  2103          2087  5                         LIB$SIGNAL(UAF$_UICERR, 2, .TOKENLEN, .TOKENPTR);
;  2104          2088  5                         RETURN ;
;  2105          2089  4                         END ;
;  2106          2090  4                 IDENT[UIC$V_FORMAT] = UIC$K_UIC_FORMAT ;
;  2107          2091  4                 END
;  2108          2092  3             ELSE
;  2109          2093  4                 BEGIN
;  2110          2094  4                 !
;  2111          2095  4                 ! A user name was specified.  Get the UAF record
;  2112          2096  4                 ! for this user and pull the UIC out.
;  2113          2097  4                 !
;  2114          2098  4                 IF .STR_WILD
;  2115          2099  4                     THEN
;  2116          2100  5                         BEGIN
;  2117          2101  5                         !
;  2118          2102  5                         ! Wild cards are not allowed for user specs
;  2119          2103  5                         !
;  2120          2104  5                         LIB$SIGNAL(UAF$_WLDNOTALWD);
;  2121          2105  5                         RETURN ;
;  2122          2106  4                         END ;
;  2123          2107  4                 FOUND_MATCH = FALSE ;
;  2124          2108  5                 IF RMSBAD ( WILD_USER (UAF$BUILD_HOLDER))
;  2125          2109  4                     THEN
;  2126          2110  5                         BEGIN
;  2127          2111  5                         IF .RMSERR EQL RMS$_RNF
;  2128          2112  5                             THEN LIB$SIGNAL(UAF$_BADSPC)
;  2129          2113  5                             ELSE LIB$SIGNAL(UAF$_RDBMDFYERR, 1, NAME_DESC,
;  2130          2114  5                                                 .RMSERR);
;  2131          2115  5                         RETURN ;
;  2132          2116  4                         END ;
;  2133          2117  3                 END ;
;  2134        P 2118  3             STATUS = $MOD_HOLDER ( ID          = .IDENT,
;  2135        P 2119  3                                    HOLDER      = HOLDER,
;  2136        P 2120  3                                    SET_ATTRIB = .SET_ATTRIBUTES,
;  2137          2121  3                                    CLR_ATTRIB = .CLR_ATTRIBUTES ) ;
;  2138          2122  3             END
;  2139          2123  2         ELSE
;  2140        P 2124  2             STATUS = $MOD_IDENT ( ID          = .OLD_IDENT,
```

```
;  2141        P 2125  2                                SET_ATTRIB = .SET_ATTRIBUTES,
;  2142        P 2126  2                                CLR_ATTRIB = .CLR_ATTRIBUTES,
;  2143        P 2127  2                                NEW_NAME   = .NEW_NAME_PTR,
;  2144          2128  2                                NEW_VALUE  = .NEW_VALUE ) ;
;  2145          2129  2
;  2146          2130  2 IF NOT .STATUS
;  2147          2131  2     THEN LIB$SIGNAL(UAF$_RDBMDFYERR, 1, NAME_DESC, .STATUS)
;  2148          2132  2     ELSE
;  2149          2133  3         BEGIN
;  2150          2134  3         RIGHTSLIST_MODIFIED = TRUE ;
;  2151          2135  3         LIB$SIGNAL(UAF$_RDBMDFYMSG, 1, NAME_DESC);
;  2152          2136  2         END ;
;  2153          2137  2
;  2154          2138  2 RETURN ;
;  2155          2139  1 END ;                                ! End of UAF$MODIFY_IDENT


                                                            .EXTRN  SYS$MOD_HOLDER, SYS$MOD_IDENT

                                                 0FFC 00000 .ENTRY  UAF$MODIFY_IDENT, Save R2,R3,R4,R5,R6,R7,-  ; 1842
                                                            R8,R9,R10,R11
                                5B 00000000'  00  9E 00002 MOVAB  IDENT, R11                                    ;
                                5A 00000000'  00  9E 00009 MOVAB  STATUS, R10                                   ;
                                5E        A4  AE  9E 00010 MOVAB  -92(SP), SP                                   ;
                             54 AE 010E0000   8F  D0 00014 MOVL   #17694720, NAME_DESC                          ; 1879
                                       58    AE  D4 0001C CLRL   NAME_DESC+4                                     ;
                             2C AE 010E0000   8F  D0 0001F MOVL   #17694720, NEW_NAME_DESC                      ; 1882
                                       30    AE  D4 00027 CLRL   NEW_NAME_DESC+4                                 ;
                                             58  7C 0002A CLRQ   SET_ATTRIBUTES                                 ;
                                00000000G    00  9F 0002C PUSHAB TOKENDSC                                       ; 1894
                                        0180 CA  9F 00032 PUSHAB SD_TOKEN1                                      ;
                          00000000G     00       02  FB 00036 CALLS  #2, CLI$GET_VALUE                         ;
                                        56 00000000G  00  3C 0003D MOVZWL TOKENLEN, R6                           ; 1896
                                        50 00000000G  00  D0 00044 MOVL   TOKENPTR, R0                          ;
                20                      60            56  2C 0004B MOVC5  R6, (R0), #32, #32, NAME_BUFF        ;
                                              34      AE      00050                                             ;
                                        50            56  D0 00052 MOVL   R6, R0                              ; 1897
                                        20            50  B1 00055 CMPW   R0, #32                             ;
                                                      03  1B 00058 BLEQU  1$                                   ;
                                        50            20  D0 0005A MOVL   #32, R0                             ;
                                   54   AE            50  B0 0005D 1$:  MOVW   R0, NAME_DESC                  ;
                                   58   AE       34   AE  9E 00061 MOVAB  NAME_BUFF, NAME_DESC+4               ; 1898
                                                      7E  D4 00066 CLRL   -(SP)                               ; 1905
                                              04   AE  9F 00068 PUSHAB OLD_IDENT                              ;
                                              5C   AE  9F 0006B PUSHAB NAME_DESC                              ;
                          00000000G     00       03  FB 0006E CALLS  #3, SYS$ASCTOID                         ;
                                        6A            50  D0 00075 MOVL   R0, STATUS                          ;
                                        03            50  E8 00078 BLBS   R0, 2$                               ; 1906
                                                  0234 31  0007B BRW    35$                                     ;
                                           0150   CA  9F 0007E 2$:  PUSHAB SD_ATTRIBRESOURCE                   ; 1919
                          00000000G     00       01  FB 00082 CALLS  #1, CLI$PRESENT                          ;
                                        6A            50  D0 00089 MOVL   R0, STATUS                          ;
                          00000000G     8F            50  D1 0008C CMPL   R0, #CLI$_NEGATED                  ; 1920
                                                      05  12 00093 BNEQ   3$                                    ;
                                        59            01  D0 00095 MOVL   #1, CLR_ATTRIBUTES                  ; 1921
                                                      06  11 00098 BRB    4$                                    ;
```

```
                                    03              50  E9 0009A 3$:      BLBC    R0, 4$                                  ;1922
                                    58              01  D0 0009D          MOVL    #1, SET_ATTRIBUTES                      ;1923
                                       000000000G   00  9F 000A0 4$:      PUSHAB  TOKENDSC                                ;1929
                                           0170     CA  9F 000A6          PUSHAB  SD_NAME                                 ;
                                00000000G  00          02  FB 000AA          CALLS   #2, CLI$GET_VALUE                   ;
                                    2F              50  E9 000B1          BLBC    R0, 6$                                  ;
                                    57          2C  AE  9E 000B4          MOVAB   NEW_NAME_DESC, NEW_NAME_PTR             ;1932
                                    56 00000000G    00  3C 000B8          MOVZWL  TOKENLEN, R6                            ;1933
                                    50 00000000G    00  D0 000BF          MOVL    TOKENPTR, R0                            ;
                20                  60              56  2C 000C6          MOVC5   R6, (R0), #32, #32, NEW_NAME_BUFF       ;
                                        0C  AE          000CB                                                           ;
                                    50              56  D0 000CD          MOVL    R6, R0                                  ;1934
                                    20              50  B1 000D0          CMPW    R0, #32                                 ;
                                                    03  1B 000D3          BLEQU   5$                                      ;
                                    50              20  D0 000D5          MOVL    #32, R0                                 ;
                                    2C  AE          50  B0 000D8 5$:      MOVW    R0, NEW_NAME_DESC                       ;
                                    30  AE      0C  AE  9E 000DC          MOVAB   NEW_NAME_BUFF, NEW_NAME_DESC+4          ;1935
                                                    02  11 000E1          BRB     7$                                      ;1929
                                                    57  D4 000E3 6$:      CLRL    NEW_NAME_PTR                            ;1937
                                        0198    CA  9F 000E5 7$:      PUSHAB  SD_VALUE                                ;1942
                                00000000G  00          01  FB 000E9          CALLS   #1, CLI$PRESENT                     ;
                                    06              50  E8 000F0          BLBS    R0, 9$                                  ;
                                        08  AE      D4 000F3          CLRL    NEW_VALUE                               ;1947
                                           00F8     31 000F6 8$:      BRW     23$                                     ;
                                       000000000G   00  9F 000F9 9$:      PUSHAB  TOKENDSC                                ;1955
                                           01A0     CA  9F 000FF          PUSHAB  SD_VALUEIDENTIFIER                      ;
                                00000000G  00          02  FB 00103          CALLS   #2, CLI$GET_VALUE                   ;
                                    41              50  E9 0010A          BLBC    R0, 12$                                 ;
                                        04  AE      D4 0010D          CLRL    TMP_IDENT                               ;1957
                                                    20  DD 00110          PUSHL   #32                                     ;1961
                                        08  AE      9F 00112          PUSHAB  TMP_IDENT                               ;
                                00000000G  00          02  FB 00115          CALLS   #2, GETVAL                          ;
                                    6A              50  D0 0011C          MOVL    R0, STATUS                              ;
                                    03              50  E8 0011F          BLBS    R0, 10$                                 ;1962
                                                   018D 31 00122          BRW     35$                                     ;
08  AE          02                  06              02  F0 00125 10$:      INSV    #2, #6, #2, NEW_VALUE+3                 ;1968
08  AE          1C                  00      04  AE  F0 0012B          INSV    TMP_IDENT, #0, #28, NEW_VALUE           ;1969
                80008000    8F      08  AE  D1 00132          CMPL    NEW_VALUE, #-2147450880                         ;1970
                                                    0A  1F 0013A          BLSSU   11$                                     ;
                8FFFFFFF    8F      08  AE  D1 0013C          CMPL    NEW_VALUE, #-1879048193                         ;1971
                                                    B0  1B 00144          BLEQU   8$                                      ;
                                       000000000G   8F  DD 00146 11$:      PUSHL   #UAF$_IDOUTRNG                          ;1974
                                                    7E  11 0014C          BRB     20$                                     ;
                                        01A8    CA  9F 0014E 12$:      PUSHAB  SD_VALUEUIC                             ;1984
                                00000000G  00          01  FB 00152          CALLS   #1, CLI$PRESENT                     ;
                                    15              50  E9 00159          BLBC    R0, 13$                                 ;
                                                    01  DD 0015C          PUSHL   #1                                      ;1985
                                        01A8    CA  9F 0015E          PUSHAB  SD_VALUEUIC                             ;
                                00000000G  00          02  FB 00162          CALLS   #2, PARSE_WILD                      ;
                                    05              50  E8 00169          BLBS    R0, 13$                                 ;
                                                    6A  DD 0016C          PUSHL   STATUS                                  ;1988
                                                   0143 31 0016E          BRW     36$                                     ;
                                    47 00000000G    00  E9 00171 13$:      BLBC    UIC_FLAG, 18$                           ;1991
                                    0F 00000000G    00  E9 00178          BLBC    GRP_WILD, 15$                           ;1994
                                    08 00000000G    00  E8 0017F          BLBS    MEM_WILD, 15$                           ;
                                       000000000G   8F  DD 00186 14$:      PUSHL   #UAF$_NOGRPWILD                         ;2000
                                                    3E  11 0018C          BRB     20$                                     ;
```

```
                                    01 DD 0018E 15$:    PUSHL   #1                                      ; 2009
                                 0C AE 9F 00190         PUSHAB  NEW_VALUE
                                 12 AE 9F 00193         PUSHAB  NEW_VALUE+2                             ; 2008
                00000000G 00        03 FB 00196         CALLS   #3, PARSE_UIC                           ; 2009
                          18        50 E8 0019D         BLBS    R0, 17$
                    00000000G 00    DD 001A0 16$:       PUSHL   TOKENPTR                                ; 2014
                          7E 00000000G 00 3C 001A6      MOVZWL  TOKENLEN, -(SP)                         ; 2013
                                    02 DD 001AD         PUSHL   #2
                    00000000G 8F    DD 001AF            PUSHL   #UAF$_UICERR
                              0107  31 001B5            BRW     37$
                       0B AE    C0 8F 8A 001B8 17$:     BICB2   #192, NEW_VALUE+3                      ; 2017
                                    32 11 001BD         BRB     23$                                     ; 1991
                          09 00000000G 00 E9 001BF 18$: BLBC    STR_WILD, 21$                           ; 2025
                    00000000G 8F    DD 001C6 19$:       PUSHL   #UAF$_WLDNOTALWD                        ; 2031
                              00B2  31 001CC 20$:       BRW     31$
                    00000000G 00    D4 001CF 21$:       CLRL    FOUND_MATCH                             ; 2034
                        FB14 CF     9F 001D5            PUSHAB  UAF$FIND_UIC                            ; 2035
00000000G 00                        01 FB 001D9         CALLS   #1, WILD_USER
00000000G 00                        50 D0 001E0         MOVL    R0, RMSERR
          03                        50 E8 001E7         BLBS    R0, 22$
                              007E  31 001EA            BRW     30$
                    08 AE           6B D0 001ED 22$:    MOVL    IDENT, NEW_VALUE                        ; 2047
                              0168  CA 9F 001F1 23$:    PUSHAB  SD_HOLDER                               ; 2060
00000000G 00                        01 FB 001F5         CALLS   #1, CLI$PRESENT
          03                        50 E8 001FC         BLBS    R0, 24$
                              0098  31 001FF            BRW     33$
                                 7E D4 00202 24$:       CLRL    -(SP)                                   ; 2063
                              0168  CA 9F 00204         PUSHAB  SD_HOLDER
00000000G 00                        02 FB 00208         CALLS   #2, PARSE_WILD
          01                        50 E8 0020F         BLBS    R0, 25$
                                    04 00212            RET
                          2C 00000000G 00 E9 00213 25$: BLBC    UIC_FLAG, 28$                           ; 2065
                          0A 00000000G 00 E9 0021A      BLBC    GRP_WILD, 26$                           ; 2068
                          03 00000000G 00 E8 00221      BLBS    MEM_WILD, 26$
                                 FF5B  31 00228         BRW     14$
                                    01 DD 0022B 26$:    PUSHL   #1                                      ; 2083
                                    5B DD 0022D         PUSHL   R11
                                 02 AB 9F 0022F         PUSHAB  IDENT+2                                 ; 2082
00000000G 00                        03 FB 00232         CALLS   #3, PARSE_UIC                           ; 2083
          03                        50 E8 00239         BLBS    R0, 27$
                                 FF61  31 0023C         BRW     16$
                       03 AB    C0 8F 8A 0023F 27$:     BICB2   #192, IDENT+3                          ; 2090
                                    43 11 00244         BRB     32$                                     ; 2065
                          03 00000000G 00 E9 00246 28$: BLBC    STR_WILD, 29$                           ; 2098
                                 FF76  31 0024D         BRW     19$
                    00000000G 00    D4 00250 29$:       CLRL    FOUND_MATCH                             ; 2107
                        F55D CF     9F 00256            PUSHAB  UAF$BUILD_HOLDER                        ; 2108
00000000G 00                        01 FB 0025A         CALLS   #1, WILD_USER
00000000G 00                        50 D0 00261         MOVL    R0, RMSERR
          1E                        50 E8 00268         BLBS    R0, 32$
          50 00000000G 00           D0 0026B 30$:       MOVL    RMSERR, R0                              ; 2111
000182B2  8F                        50 D1 00272         CMPL    R0, #98994
                                    37 12 00279         BNEQ    35$
                    00000000G 8F    DD 0027B            PUSHL   #UAF$_BADSPC                            ; 2112
00000000G 00                        01 FB 00281 31$:    CALLS   #1, LIB$SIGNAL
                                    04 00288            RET
          7E                        58 7D 00289 32$:    MOVQ    SET_ATTRIBUTES, -(SP)                   ; 2121
```

```
                                    F8   AB  9F 0028C          PUSHAB  HOLDER                                    ;
                                         6B  DD 0028F          PUSHL   IDENT                                     ;
                    00000000G  00        04  FB 00291          CALLS   #4, SYS$MOD_HOLDER                        ;
                                         12  11 00298          BRB     34$                                       ;
                                    08   AE  DD 0029A 33$:     PUSHL   NEW_VALUE                                 ; 2128
                                         57  DD 0029D          PUSHL   NEW_NAME_PTR                              ;
                               7E        58  7D 0029F          MOVQ    SET_ATTRIBUTES, -(SP)                     ;
                                    10   AE  DD 002A2          PUSHL   OLD_IDENT                                 ;
                    00000000G  00        05  FB 002A5          CALLS   #5, SYS$MOD_IDENT                         ;
                               6A        50  D0 002AC 34$:     MOVL    R0, STATUS                                ;
                               15        50  E8 002AF          BLBS    R0, 38$                                   ; 2130
                                         50  DD 002B2 35$:     PUSHL   R0                                        ; 2131
                                    58   AE  9F 002B4 36$:     PUSHAB  NAME_DESC                                 ;
                                         01  DD 002B7          PUSHL   #1                                        ;
                         00000000G       8F  DD 002B9          PUSHL   #UAF$_RDBMDFYERR                          ;
                    00000000G  00        04  FB 002BF 37$:     CALLS   #4, LIB$SIGNAL                            ;
                                             04 002C6          RET                                               ;
                    00000000G  00        01  90 002C7 38$:     MOVB    #1, RIGHTSLIST_MODIFIED                   ; 2134
                                    54   AE  9F 002CE          PUSHAB  NAME_DESC                                 ; 2135
                                         01  DD 002D1          PUSHL   #1                                        ;
                         00000000G       8F  DD 002D3          PUSHL   #UAF$_RDBMDFYMSG                          ;
                    00000000G  00        03  FB 002D9          CALLS   #3, LIB$SIGNAL                            ;
                                             04 002E0          RET                                               ; 2139

; Routine Size:  737 bytes,    Routine Base:  $CODE$ + 0CAE

; 2156          2140  1
```

```
;  2158          2141  1 %SBTTL ' UAF$REMOVE_GRP_IDENT - Remove group identifier from RDB'
;  2159          2142  1
;  2160          2143  1 GLOBAL ROUTINE UAF$REMOVE_GRP_IDENT : NOVALUE =
;  2161          2144  2 BEGIN
;  2162          2145  2 !++
;  2163          2146  2 !
;  2164          2147  2 ! FUNCTIONAL DESCRIPTION:
;  2165          2148  2 !
;  2166          2149  2 !       This routine is used to remove a group identifier from the rights data
;  2167          2150  2 !       base if there is no longer any in that group in the UAF.
;  2168          2151  2 !
;  2169          2152  2 !
;  2170          2153  2 ! INPUTS:
;  2171          2154  2 !
;  2172          2155  2 !       none
;  2173          2156  2 !
;  2174          2157  2 ! IMPLICIT INPUTS:
;  2175          2158  2 !
;  2176          2159  2 !       RECBUF must contain the UAF record which was just deleted.
;  2177          2160  2 !
;  2178          2161  2 ! OUTPUTS:
;  2179          2162  2 !
;  2180          2163  2 !       None
;  2181          2164  2 !
;  2182          2165  2 ! IMPLICIT OUTPUTS:
;  2183          2166  2 !
;  2184          2167  2 !       none
;  2185          2168  2 !
;  2186          2169  2 ! ROUTINE VALUE:
;  2187          2170  2 !
;  2188          2171  2 !       none
;  2189          2172  2 !
;  2190          2173  2 ! SIDE EFFECTS:
;  2191          2174  2 !
;  2192          2175  2 !       rights data base is modified
;  2193          2176  2 !
;  2194          2177  2 !--
;  2195          2178  2
;  2196          2179  2
;  2197          2180  2 BIND
;  2198          2181  2     UICMEMWLD    = CSTRING ('[!OW,*]') ;
;  2199          2182  2
;  2200          2183  2 LITERAL
;  2201          2184  2     STR_BUFFLEN = 20 ;
;  2202          2185  2
;  2203          2186  2 LOCAL
;  2204          2187  2     FAO_DESC     : STATDESC ,
;  2205          2188  2     NAME_DESC    : STATDESC,
;  2206          2189  2     NAME_BUFF    : VECTOR [ KGB$S_NAME, BYTE ] ,
;  2207          2190  2     STR_DESC     : STATDESC ,
;  2208          2191  2     STR_BUFF     : VECTOR [STR_BUFFLEN, BYTE ] ,
;  2209          2192  2     WILD_IDENT   : $BBLOCK [4] ;
;  2210          2193  2
;  2211          2194  2 !
;  2212          2195  2 ! Get the uic from the recbuf and build an identifier with it
;  2213          2196  2 ! Also build a wild card member version.
;  2214          2197  2 !
```

```
;  2215     2198  2 UAF$FIND_UIC ();
;  2216     2199  2 WILD_IDENT[UIC$V_FORMAT] = .IDENT[UIC$V_FORMAT] ;
;  2217     2200  2 WILD_IDENT[UIC$V_GROUP]  = .IDENT[UIC$V_GROUP] ;
;  2218     2201  2 WILD_IDENT[UIC$V_MEMBER] = UIC$K_WILD_MEMBER ;
;  2219     2202  2
;  2220     2203  2 !
;  2221     2204  2 ! Before we go any further see if the identifier even exists and get
;  2222     2205  2 ! it's name
;  2223     2206  2 !
;  2224     2207  2 NAME_DESC[LENGTH]  = KGB$S_NAME ;
;  2225     2208  2 NAME_DESC[POINTER] = NAME_BUFF ;
;  2226   P 2209  3 IF NOT ( $IDTOASC ( ID     = .WILD_IDENT,
;  2227   P 2210  3                      NAMLEN = NAME_DESC[LENGTH] ,
;  2228     2211  3                      NAMBUF = NAME_DESC ) )
;  2229     2212  2     THEN RETURN ;
;  2230     2213  2
;  2231     2214  2 !
;  2232     2215  2 ! Now build an ascii UIC string that we can pass to PARSE_WILD_UIC
;  2233     2216  2 ! After using this routine to parse the string we can use the standard
;  2234     2217  2 ! method of reading the UAF (ie. WILD_USER )
;  2235     2218  2 !
;  2236     2219  2 FAO_DESC[LENGTH]  = .UICMEMWLD<0,8> ;
;  2237     2220  2 FAO_DESC[POINTER] = UICMEMWLD + 1 ;
;  2238     2221  2 STR_DESC[LENGTH]  = STR_BUFFLEN ;
;  2239     2222  2 STR_DESC[POINTER] = STR_BUFF ;
;  2240     2223  2 $FAO ( FAO_DESC, STR_DESC[LENGTH], STR_DESC, .WILD_IDENT[UIC$V_GROUP] ) ;
;  2241     2224  2
;  2242     2225  2 !
;  2243     2226  2 ! Now set a global flag stating that the group is empty.  If WILD_USER
;  2244     2227  2 ! finds any records it will call UAF$RECORD_FOUND which will change
;  2245     2228  2 ! the state of the group empty flag and return a failure to WILD_USER.
;  2246     2229  2 ! The failure code will cause WILD_USER to exit it's loop there by
;  2247     2230  2 ! stopping as soon as a single record is found.
;  2248     2231  2 !
;  2249     2232  2 PARSE_WILD_UIC ( .STR_DESC[LENGTH], .STR_DESC[POINTER] ) ;
;  2250     2233  2 FOUND_MATCH = FALSE ;
;  2251     2234  2 STATUS = WILD_USER ( UAF$RECORD_FOUND ) ;
;  2252     2235  3 IF RMSBAD ( .STATUS )
;  2253     2236  2     THEN IF (.RMSERR NEQ RMS$_RNF) AND
;  2254     2237  3             (.RMSERR NEQ FALSE )
;  2255     2238  2          THEN LIB$SIGNAL(UAF$_RDBREMERR, 1, NAME_DESC, .RMSERR);
;  2256     2239  2
;  2257     2240  2 !
;  2258     2241  2 ! Now if the flag is still set to false
;  2259     2242  2 ! we can remove the identifier .
;  2260     2243  2 !
;  2261     2244  2 IF NOT .FOUND_MATCH
;  2262     2245  2     THEN
;  2263     2246  3         BEGIN
;  2264     2247  3
;  2265     2248  3         STATUS = $REM_IDENT ( ID = .WILD_IDENT ) ;
;  2266     2249  3
;  2267     2250  3         IF NOT .STATUS
;  2268     2251  3             THEN
;  2269     2252  4                 BEGIN
;  2270     2253  4                 LIB$SIGNAL(UAF$_RDBREMERR, 1, NAME_DESC, .STATUS);
;  2271     2254  4                 RETURN ;
```

```
2255 4                  END
2256 3          ELSE
2257 4                  BEGIN
2258 4                  RIGHTSLIST_MODIFIED = TRUE ;
2259 4                  IF .WILD_IDENT[UIC$V_FORMAT] EQL UIC$K_UIC_FORMAT
2260 4                      THEN LIB$SIGNAL(UAF$_RDBREMMSGU, 3, NAME_DESC,
2261 4                                  .WILD_IDENT[UIC$V_GROUP],
2262 4                                  .WILD_IDENT[UIC$V_MEMBER])
2263 4                      ELSE LIB$SIGNAL(UAF$_RDBREMMSG, 2, NAME_DESC, .WILD_IDENT);
2264 3                  END ;
2265 2          END ;
2266 2
2267 2  RETURN ;
2268 2
2269 1  END ;                                   ! End of UAF$REMOVE_GRP_IDENT


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                     07  0006D P.AAO:   .BYTE   7                                       ;
                5D  2A  2C  57  4F  21  5B  0006E       .ASCII  \[!OW,*]\                               ;

                                        UICMEMWLD=              P.AAO
                                                        .EXTRN  SYS$FAO, SYS$REM_IDENT

                                                        .PSECT  $CODE$,NOWRT,2

                                   00FC 00000           .ENTRY  UAF$REMOVE_GRP_IDENT, Save R2,R3,R4,R5,R6,- ; 2143
                                                                R7                                      ;
                     57 00000000G  8F  D0 00002         MOVL    #UAF$_RDBREMERR, R7                     ;
                     56 00000000G  00  9E 00009         MOVAB   RMSERR, R6                              ;
                     55 00000000'  00  9E 00010         MOVAB   STATUS, R5                              ;
                     54 00000000G  00  9E 00017         MOVAB   FOUND_MATCH, R4                         ;
                     53 00000000G  00  9E 0001E         MOVAB   LIB$SIGNAL, R3                          ;
                     5E        B4  AE  9E 00025         MOVAB   -76(SP), SP                             ;
           44     AE 010E0000  8F  D0 00029             MOVL    #17694720, FAO_DESC                     ; 2187
                           48  AE  D4 00031             CLRL    FAO_DESC+4                              ;
           3C     AE 010E0000  8F  D0 00034             MOVL    #17694720, NAME_DESC                    ; 2188
                           40  AE  D4 0003C             CLRL    NAME_DESC+4                             ;
           14     AE 010E0000  8F  D0 0003F             MOVL    #17694720, STR_DESC                     ; 2190
                           18  AE  D4 00047             CLRL    STR_DESC+4                              ;
         F9BD     CF           00  FB 0004A             CALLS   #0, UAF$FIND_UIC                        ; 2198
50 00000000'   00  02          06  EF 0004F             EXTZV   #6, #2, IDENT+3, R0                     ; 2199
52             02  1E          50  F0 00058             INSV    R0, #30, #2, WILD_IDENT                 ;
52             0E  10 00000000' 00  F0 0005D            INSV    IDENT+2, #16, #14, WILD_IDENT           ; 2200
                   52          01  AE 00066             MNEGW   #1, WILD_IDENT                          ; 2201
           3C      AE          20  B0 00069             MOVW    #32, NAME_DESC                          ; 2207
           40      AE      1C  AE  9E 0006D             MOVAB   NAME_BUFF, NAME_DESC+4                  ; 2208
                               7E  7C 00072             CLRQ    -(SP)                                   ; 2211
                               7E  D4 00074             CLRL    -(SP)                                   ;
                           48  AE  9F 00076             PUSHAB  NAME_DESC                               ;
                           4C  AE  9F 00079             PUSHAB  NAME_DESC                               ;
                               52  DD 0007C             PUSHL   WILD_IDENT                              ;
        00000000G  00          06  FB 0007E             CALLS   #6, SYS$IDTOASC                         ;
                   01          50  E8 00085             BLBS    R0, 1$                                  ;
                                   04 00088             RET                                             ;
```

```
                                44   AE 00000000'   00  9B 00089 1$:     MOVZBW  UICMEMWLD, FAO_DESC                          ; 2219
                                48   AE 00000000'   00  9E 00091         MOVAB   UICMEMWLD+1, FAO_DESC+4                      ; 2220
                                14   AE             14  B0 00099         MOVW    #20, STR_DESC                                ; 2221
                                18   AE             6E  9E 0009D         MOVAB   STR_BUFF, STR_DESC+4                         ; 2222
                7E        52          0E             10  EF 000A1         EXTZV   #16, #14, WILD_IDENT, -(SP)                  ; 2223
                                               18   AE  9F 000A6         PUSHAB  STR_DESC                                     ;
                                               1C   AE  9F 000A9         PUSHAB  STR_DESC                                     ;
                                               50   AE  9F 000AC         PUSHAB  FAO_DESC                                     ;
                                   00000000G   00      04  FB 000AF         CALLS   #4, SYS$FAO                                  ;
                                               18   AE  DD 000B6         PUSHL   STR_DESC+4                                   ; 2232
                                   7E          18   AE  3C 000B9         MOVZWL  STR_DESC, -(SP)                              ;
                                   00000000G   00      02  FB 000BD         CALLS   #2, PARSE_WILD_UIC                           ;
                                                       64  D4 000C4         CLRL    FOUND_MATCH                                  ; 2233
                                               FB4B    CF  9F 000C6         PUSHAB  UAF$RECORD_FOUND                             ; 2234
                                   00000000G   00      01  FB 000CA         CALLS   #1, WILD_USER                                ;
                                               65      50  D0 000D1         MOVL    R0, STATUS                                   ;
                                               66      50  D0 000D4         MOVL    R0, RMSERR                                   ; 2235
                                               1C      50  E8 000D7         BLBS    R0, 2$                                       ;
                                               50          66  D0 000DA         MOVL    RMSERR, R0                               ; 2236
                                   000182B2    8F          50  D1 000DD         CMPL    R0, #98994                               ;
                                                           10  13 000E4         BEQL    2$                                       ;
                                                           50  D5 000E6         TSTL    R0                                       ; 2237
                                                           0C  13 000E8         BEQL    2$                                       ;
                                                           50  DD 000EA         PUSHL   R0                                       ; 2238
                                               40      AE  9F 000EC         PUSHAB  NAME_DESC                                    ;
                                                       01  DD 000EF         PUSHL   #1                                           ;
                                                       57  DD 000F1         PUSHL   R7                                           ;
                                               63          04  FB 000F3         CALLS   #4, LIB$SIGNAL                           ;
                                               4F          64  E8 000F6 2$:     BLBS    FOUND_MATCH, 6$                          ; 2244
                                                           52  DD 000F9         PUSHL   WILD_IDENT                               ; 2248
                                   00000000G   00      01  FB 000FB         CALLS   #1, SYS$REM_IDENT                            ;
                                               65      50  D0 00102         MOVL    R0, STATUS                                   ;
                                               0B          50  E8 00105         BLBS    R0, 3$                                   ; 2250
                                                           50  DD 00108         PUSHL   R0                                       ; 2253
                                               40      AE  9F 0010A         PUSHAB  NAME_DESC                                    ;
                                                       01  DD 0010D         PUSHL   #1                                           ;
                                                       57  DD 0010F         PUSHL   R7                                           ;
                                                           32  11 00111         BRB     5$                                       ;
                      00000000G   00                   01  90 00113 3$:     MOVB    #1, RIGHTSLIST_MODIFIED                  ; 2258
        00        52          02                       1E  ED 0011A         CMPZV   #30, #2, WILD_IDENT, #0                      ; 2259
                                                       17  12 0011F         BNEQ    4$                                       ;
                                   7E                  52  3C 00121         MOVZWL  WILD_IDENT, -(SP)                            ; 2262
        7E        52          0E                       10  EF 00124         EXTZV   #16, #14, WILD_IDENT, -(SP)                  ; 2261
                                               44   AE  9F 00129         PUSHAB  NAME_DESC                                    ; 2260
                                                       03  DD 0012C         PUSHL   #3                                           ;
                                      00000000G    8F      DD 0012E         PUSHL   #UAF$_RDBREMMSGU                             ;
                                   63                  05  FB 00134         CALLS   #5, LIB$SIGNAL                               ;
                                                       04  00137         RET                                                  ;
                                                       52  DD 00138 4$:     PUSHL   WILD_IDENT                               ; 2263
                                               40      AE  9F 0013A         PUSHAB  NAME_DESC                                    ;
                                                       02  DD 0013D         PUSHL   #2                                           ;
                                      00000000G    8F      DD 0013F         PUSHL   #UAF$_RDBREMMSG                              ;
                                   63                  04  FB 00145 5$:     CALLS   #4, LIB$SIGNAL                               ;
                                                           04  00148 6$:     RET                                              ; 2269

; Routine Size:  329 bytes,    Routine Base:  $CODE$ + 0F8F
```

```
; 2287          2270  1
```

```
;  2289          2271  1 %SBTTL ' UAF$REMOVE_IDENT - Remove identifier from RDB'
;  2290          2272  1
;  2291          2273  1 GLOBAL ROUTINE UAF$REMOVE_IDENT : NOVALUE =
;  2292          2274  2 BEGIN
;  2293          2275  2 !++
;  2294          2276  2 !
;  2295          2277  2 ! FUNCTIONAL DESCRIPTION:
;  2296          2278  2 !
;  2297          2279  2 !       This routine is used to remove an identifier from the rights
;  2298          2280  2 !       data base.  It is invoked via the REMOVE/IDENT command.
;  2299          2281  2 !
;  2300          2282  2 ! INPUTS:
;  2301          2283  2 !
;  2302          2284  2 !       none
;  2303          2285  2 !
;  2304          2286  2 ! IMPLICIT INPUTS:
;  2305          2287  2 !
;  2306          2288  2 !       command line
;  2307          2289  2 !
;  2308          2290  2 ! OUTPUTS:
;  2309          2291  2 !
;  2310          2292  2 !       None
;  2311          2293  2 !
;  2312          2294  2 ! IMPLICIT OUTPUTS:
;  2313          2295  2 !
;  2314          2296  2 !       none
;  2315          2297  2 !
;  2316          2298  2 ! ROUTINE VALUE:
;  2317          2299  2 !
;  2318          2300  2 !       none
;  2319          2301  2 !
;  2320          2302  2 ! SIDE EFFECTS:
;  2321          2303  2 !
;  2322          2304  2 !       rights data base is modified
;  2323          2305  2 !
;  2324          2306  2 !--
;  2325          2307  2
;  2326          2308  2
;  2327          2309  2 !
;  2328          2310  2 ! Get the ID name.  This is a required parameter so we are guaranteed
;  2329          2311  2 ! that it will be present
;  2330          2312  2 !
;  2331          2313  2 CLI$GET_VALUE ( SD_TOKEN1, TOKENDSC ) ;
;  2332          2314  2
;  2333          2315  2 !
;  2334          2316  2 ! Convert the ID from ASCII to a longword format
;  2335          2317  2 !
;  2336        P 2318  2 STATUS = $ASCTOID ( NAME    = TOKENDSC,
;  2337        P 2319  2                     ID      = IDENT,
;  2338          2320  2                     ATTRIB = 0 ) ;
;  2339          2321  2
;  2340          2322  2 IF .STATUS
;  2341          2323  2     THEN
;  2342          2324  2          !
;  2343          2325  2          ! Now remove the entry
;  2344          2326  2          !
;  2345          2327  2          STATUS = $REM_IDENT ( ID = .IDENT ) ;
```

```
;  2346          2328  2
;  2347          2329  2 IF NOT .STATUS
;  2348          2330  2     THEN
;  2349          2331  3         BEGIN
;  2350          2332  3         LIB$SIGNAL(UAF$_RDBREMERR, 1, TOKENDSC, .STATUS);
;  2351          2333  3         RETURN ;
;  2352          2334  3         END
;  2353          2335  2     ELSE
;  2354          2336  3         BEGIN
;  2355          2337  3         RIGHTSLIST_MODIFIED = TRUE ;
;  2356          2338  3         IF .IDENT[UIC$V_FORMAT] EQL UIC$K_UIC_FORMAT
;  2357          2339  3             THEN LIB$SIGNAL(UAF$_RDBREMMSGU, 3, TOKENDSC,
;  2358          2340  3                             .IDENT[UIC$V_GROUP], .IDENT[UIC$V_MEMBER])
;  2359          2341  3             ELSE LIB$SIGNAL(UAF$_RDBREMMSG, 2, TOKENDSC, .IDENT);
;  2360          2342  2         END ;
;  2361          2343  2
;  2362          2344  2 RETURN ;
;  2363          2345  2
;  2364          2346  1 END ;                                   ! End of UAF$REMOVE_IDENT


                                    003C 00000         .ENTRY  UAF$REMOVE_IDENT, Save R2,R3,R4,R5      ; 2273
                          55 00000000G  00  9E 00002   MOVAB   LIB$SIGNAL, R5                          ;
                          54 00000000G  00  9E 00009   MOVAB   TOKENDSC, R4                            ;
                          53 00000000'  00  9E 00010   MOVAB   STATUS, R3                              ;
                          52 00000000'  00  9E 00017   MOVAB   IDENT, R2                              ;
                                        54  DD 0001E   PUSHL   R4                                     ; 2313
                                  0180  C3  9F 00020   PUSHAB  SD_TOKEN1                              ;
                    00000000G     00      02  FB 00024   CALLS   #2, CLI$GET_VALUE                    ;
                                          7E  D4 0002B   CLRL    -(SP)                                ; 2320
                                          52  DD 0002D   PUSHL   R2                                   ;
                                          54  DD 0002F   PUSHL   R4                                   ;
                    00000000G     00      03  FB 00031   CALLS   #3, SYS$ASCTOID                      ;
                                  63      50  D0 00038   MOVL    R0, STATUS                           ;
                                  0C      63  E9 0003B   BLBC    STATUS, 1$                           ; 2322
                                          62  DD 0003E   PUSHL   IDENT                               ; 2327
                    00000000G     00      01  FB 00040   CALLS   #1, SYS$REM_IDENT                   ;
                                  63      50  D0 00047   MOVL    R0, STATUS                           ;
                                  50      63  D0 0004A 1$: MOVL    STATUS, R0                           ; 2329
                                  0E      50  E8 0004D   BLBS    R0, 2$                               ;
                                          50  DD 00050   PUSHL   R0                                   ; 2332
                                          54  DD 00052   PUSHL   R4                                   ;
                                          01  DD 00054   PUSHL   #1                                   ;
                                00000000G     8F  DD 00056   PUSHL   #UAF$_RDBREMERR                   ;
                                              31  11 0005C   BRB     4$                               ;
                    00000000G     00          01  90 0005E 2$: MOVB    #1, RIGHTSLIST_MODIFIED          ; 2337
                          C0      8F    03    A2  93 00065   BITB    IDENT+3, #192                    ; 2338
                                              17  12 0006A   BNEQ    3$                               ;
                                          7E  62  3C 0006C   MOVZWL  IDENT, -(SP)                     ; 2340
7E        02    A2                      0E  00  EF 0006F   EXTZV   #0, #14, IDENT+2, -(SP)           ;
                                          54  DD 00075   PUSHL   R4                                   ; 2339
                                          03  DD 00077   PUSHL   #3                                   ;
                                00000000G     8F  DD 00079   PUSHL   #UAF$_RDBREMMSGU                  ;
                                      65          05  FB 0007F   CALLS   #5, LIB$SIGNAL               ;
```

```
                                    04 00082          RET                                                 ;
                                62  DD 00083 3$:      PUSHL   IDENT                                       ; 2341
                                54  DD 00085          PUSHL   R4                                          ;
                                02  DD 00087          PUSHL   #2                                          ;
                    00000000G   8F  DD 00089          PUSHL   #UAF$_RDBREMMSG                             ;
                            65      04  FB 0008F 4$:  CALLS   #4, LIB$SIGNAL                              ;
                                    04 00092          RET                                                 ; 2346

; Routine Size:  147 bytes,    Routine Base:  $CODE$ + 10D8

; 2365          2347  1
```

```
;  2367          2348  1 %SBTTL ' UAF$REMOVE_IDENT_RECBUF - Remove identifier from RDB'
;  2368          2349  1
;  2369          2350  1 GLOBAL ROUTINE UAF$REMOVE_IDENT_RECBUF =
;  2370          2351  2 BEGIN
;  2371          2352  2 !++
;  2372          2353  2 !
;  2373          2354  2 ! FUNCTIONAL DESCRIPTION:
;  2374          2355  2 !
;  2375          2356  2 !       This routine is used to remove an identifier from the rights data
;  2376          2357  2 !       base.  The identifier to be removed is determined by the UIC of
;  2377          2358  2 !       the UAF record in RECBUF
;  2378          2359  2 !
;  2379          2360  2 ! INPUTS:
;  2380          2361  2 !
;  2381          2362  2 !       none
;  2382          2363  2 !
;  2383          2364  2 ! IMPLICIT INPUTS:
;  2384          2365  2 !
;  2385          2366  2 !       RECBUF must contain a valid UAF record
;  2386          2367  2 !
;  2387          2368  2 ! OUTPUTS:
;  2388          2369  2 !
;  2389          2370  2 !       None
;  2390          2371  2 !
;  2391          2372  2 ! IMPLICIT OUTPUTS:
;  2392          2373  2 !
;  2393          2374  2 !       none
;  2394          2375  2 !
;  2395          2376  2 ! ROUTINE VALUE:
;  2396          2377  2 !
;  2397          2378  2 !       Always true
;  2398          2379  2 !
;  2399          2380  2 ! SIDE EFFECTS:
;  2400          2381  2 !
;  2401          2382  2 !       rights data base is modified
;  2402          2383  2 !--
;  2403          2384  2
;  2404          2385  2
;  2405          2386  2
;  2406          2387  2 BIND
;  2407          2388  2     UICSTRING   = CSTRING ('[!OW,!OW]') ;
;  2408          2389  2
;  2409          2390  2 LITERAL
;  2410          2391  2     STR_BUFFLEN = 20 ;
;  2411          2392  2
;  2412          2393  2 LOCAL
;  2413          2394  2     FAO_DESC    : STATDESC ,
;  2414          2395  2     NAME_DESC   : STATDESC,
;  2415          2396  2     NAME_BUFF   : VECTOR [ KGB$S_NAME, BYTE ] ,
;  2416          2397  2     STR_DESC    : STATDESC,
;  2417          2398  2     STR_BUFF    : VECTOR [STR_BUFFLEN, BYTE ] ;
;  2418          2399  2
;  2419          2400  2 !
;  2420          2401  2 ! Get the uic from the recbuf
;  2421          2402  2 !
;  2422          2403  2 UAF$FIND_UIC ();
;  2423          2404  2
```

```
;  2424          2405  2 !
;  2425          2406  2 ! Before we go any further see if the identifier even exists and get
;  2426          2407  2 ! it's name.  If it doesn't exist, skip right to the Group identifier
;  2427          2408  2 !
;  2428          2409  2 NAME_DESC[LENGTH]  = KGB$S_NAME ;
;  2429          2410  2 NAME_DESC[POINTER] = NAME_BUFF ;
;  2430        P 2411  3 IF ( $IDTOASC ( ID     = .IDENT,
;  2431        P 2412  3                  NAMLEN = NAME_DESC[LENGTH] ,
;  2432          2413  3                  NAMBUF = NAME_DESC ) )
;  2433          2414  2     THEN
;  2434          2415  3         BEGIN
;  2435          2416  3         !
;  2436          2417  3         ! Now build an ascii UIC string that we can pass to PARSE_WILD_UIC
;  2437          2418  3         ! After using this routine to parse the string we can use the standard
;  2438          2419  3         ! method of reading the UAF (ie. WILD_USER )
;  2439          2420  3         !
;  2440          2421  3         FAO_DESC[LENGTH]  = .UICSTRING<0,8> ;
;  2441          2422  3         FAO_DESC[POINTER] = UICSTRING + 1 ;
;  2442          2423  3         STR_DESC[LENGTH]  = STR_BUFFLEN ;
;  2443          2424  3         STR_DESC[POINTER] = STR_BUFF ;
;  2444        P 2425  3         $FAO ( FAO_DESC, STR_DESC[LENGTH], STR_DESC,
;  2445          2426  3              .IDENT[UIC$V_GROUP], .IDENT[UIC$V_MEMBER] ) ;
;  2446          2427  3
;  2447          2428  3         !
;  2448          2429  3         ! Now set a global flag stating that the record was not found.  If WILD_USER
;  2449          2430  3         ! finds any records it will call UAF$RECORD_FOUND which will change
;  2450          2431  3         ! the state of the flag and return a failure to WILD_USER.
;  2451          2432  3         ! The failure code will cause WILD_USER to exit it's loop there by
;  2452          2433  3         ! stopping as soon as a single record is found.
;  2453          2434  3         !
;  2454          2435  3         PARSE_WILD_UIC ( .STR_DESC[LENGTH], .STR_DESC[POINTER] ) ;
;  2455          2436  3         FOUND_MATCH = FALSE ;
;  2456          2437  3         STATUS = WILD_USER ( UAF$RECORD_FOUND ) ;
;  2457          2438  4         IF RMSBAD ( .STATUS )
;  2458          2439  3             THEN IF (.RMSERR NEQ RMS$_RNF) AND
;  2459          2440  4                     (.RMSERR NEQ FALSE )
;  2460          2441  3                 THEN LIB$SIGNAL(UAF$_RDBREMERR, 1, NAME_DESC, .RMSERR);
;  2461          2442  3
;  2462          2443  3         !
;  2463          2444  3         ! Now if the found_match flag is still set to false
;  2464          2445  3         ! we can remove the identifier .
;  2465          2446  3         !
;  2466          2447  3         IF NOT .FOUND_MATCH
;  2467          2448  3             THEN
;  2468          2449  4                 BEGIN
;  2469          2450  4
;  2470          2451  4                 STATUS = $REM_IDENT ( ID = .IDENT ) ;
;  2471          2452  4
;  2472          2453  4                 IF NOT .STATUS
;  2473          2454  4                     THEN
;  2474          2455  5                         BEGIN
;  2475          2456  5                         LIB$SIGNAL(UAF$_RDBREMERR, 1, NAME_DESC, .STATUS);
;  2476          2457  5                         RETURN TRUE ;
;  2477          2458  5                         END
;  2478          2459  4                     ELSE
;  2479          2460  5                         BEGIN
;  2480          2461  5                         RIGHTSLIST_MODIFIED = TRUE ;
```

```
;  2481          2462  5                          IF .IDENT[UIC$V_FORMAT] EQL UIC$K_UIC_FORMAT
;  2482          2463  5                              THEN LIB$SIGNAL(UAF$_RDBREMMSGU, 3, NAME_DESC,
;  2483          2464  5                                              .IDENT[UIC$V_GROUP],
;  2484          2465  5                                              .IDENT[UIC$V_MEMBER])
;  2485          2466  5                              ELSE LIB$SIGNAL(UAF$_RDBREMMSG, 2, NAME_DESC,
;  2486          2467  5                                              .IDENT);
;  2487          2468  4                          END ;
;  2488          2469  3                  END ;
;  2489          2470  2          END ;
;  2490          2471  2
;  2491          2472  2 !
;  2492          2473  2 ! Scan the UAF and see if there is anyone left in this group.
   2493          2474  2 ! If no then delete the group's identifier.
;  2494          2475  2 !
;  2495          2476  2 UAF$REMOVE_GRP_IDENT () ;
;  2496          2477  2
;  2497          2478  2 RETURN TRUE ;
;  2498          2479  2
;  2499          2480  1 END ;                                  ! End of UAF$REMOVE_IDENT_RECBUF


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                        09  00075 P.AAP:  .BYTE   9                                       ;
                                                5D  57  4F  21  2C  57  4F  21  5B  00076        .ASCII  \[!OW,!OW]\                             ;

                                                                UICSTRING=          P.AAP

                                                                .PSECT  $CODE$,NOWRT,2

                                                                .PSECT  $CODE$,NOWRT,2

                                                        01FC 00000      .ENTRY  UAF$REMOVE_IDENT_RECBUF, Save R2,R3,R4,R5,- ; 2350
                                                                        R6,R7,R8                                            ;
                                                    58 00000000G   8F   D0 00002        MOVL    #UAF$_RDBREMERR, R8         ;
                                                    57 00000000G   00   9E 00009        MOVAB   RMSERR, R7                  ;
                                                    56 00000000'   00   9E 00010        MOVAB   STATUS, R6                  ;
                                                    55 00000000G   00   9E 00017        MOVAB   FOUND_MATCH, R5             ;
                                                    54      FF0E   CF   9E 0001E        MOVAB   UAF$FIND_UIC, R4            ;
                                                    53 00000000G   00   9E 00023        MOVAB   LIB$SIGNAL, R3              ;
                                                    52 00000000'   00   9E 0002A        MOVAB   IDENT, R2                   ;
                                                    5E        B4   AE   9E 00031        MOVAB   -76(SP), SP                 ;
                                                44  AE 010E0000   8F   D0 00035        MOVL    #17694720, FAO_DESC         ; 2394
                                                          48  AE   D4 0003D        CLRL    FAO_DESC+4                  ;
                                                3C  AE 010E0000   8F   D0 00040        MOVL    #17694720, NAME_DESC        ; 2395
                                                          40  AE   D4 00048        CLRL    NAME_DESC+4                 ;
                                                14  AE 010E0000   8F   D0 0004B        MOVL    #17694720, STR_DESC         ; 2397
                                                          18  AE   D4 00053        CLRL    STR_DESC+4                  ;
                                                    64            FB 00056        CALLS   #0, UAF$FIND_UIC            ; 2403
                                                3C  AE            B0 00059        MOVW    #32, NAME_DESC              ; 2409
                                                40  AE        1C  AE   9E 0005D        MOVAB   NAME_BUFF, NAME_DESC+4      ; 2410
                                                                  7E   7C 00062        CLRQ    -(SP)                       ; 2413
                                                                  7E   D4 00064        CLRL    -(SP)                       ;
                                                          48  AE   9F 00066        PUSHAB  NAME_DESC                   ;
                                                          4C  AE   9F 00069        PUSHAB  NAME_DESC                   ;
                                                                  62   DD 0006C        PUSHL   IDENT                       ;
                                            00000000G   00        06   FB 0006E        CALLS   #6, SYS$IDTOASC             ;
```

```
                                 03            50  E8 00075          BLBS    R0, 1$                                  ;
                                             00C8  31 00078          BRW     5$                                      ;
                              44 AE 00000000'    00  9B 00078 1$:    MOVZBW  UICSTRING, FAO_DESC                     ; 2421
                              48 AE 00000000'    00  9E 00083        MOVAB   UICSTRING+1, FAO_DESC+4                 ; 2422
                              14 AE              14  B0 0008B        MOVW    #20, STR_DESC                           ; 2423
                              18 AE              6E  9E 0008F        MOVAB   STR_BUFF, STR_DESC+4                    ; 2424
                                 7E              62  3C 00093        MOVZWL  IDENT, -(SP)                            ; 2426
        7E      02  A2                  0E      00  EF 00096        EXTZV   #0, #14, IDENT+2, -(SP)                 ;
                                          1C     AE  9F 0009C        PUSHAB  STR_DESC                                ;
                                          20     AE  9F 0009F        PUSHAB  STR_DESC                                ;
                                          54     AE  9F 000A2        PUSHAB  FAO_DESC                                ;
                          00000000G       00         05  FB 000A5    CALLS   #5, SYS$FAO                             ;
                                          18     AE  DD 000AC        PUSHL   STR_DESC+4                              ; 2435
                                 7E       18     AE  3C 000AF        MOVZWL  STR_DESC, -(SP)                         ;
                          00000000G       00         02  FB 000B3    CALLS   #2, PARSE_WILD_UIC                      ;
                                             65      D4 000BA        CLRL    FOUND_MATCH                             ; 2436
                                        0209     C4  9F 000BC        PUSHAB  UAF$RECORD_FOUND                        ; 2437
                          00000000G       00         01  FB 000C0    CALLS   #1, WILD_USER                           ;
                                 66               50  D0 000C7        MOVL    R0, STATUS                              ;
                                 67               50  D0 000CA        MOVL    R0, RMSERR                              ; 2438
                                 1C               50  E8 000CD        BLBS    R0, 2$                                  ;
                                 50               67  D0 000D0        MOVL    RMSERR, R0                              ; 2439
                          000182B2  8F            50  D1 000D3        CMPL    R0, #98994                              ;
                                                  10  13 000DA        BEQL    2$                                      ;
                                                  50  D5 000DC        TSTL    R0                                      ; 2440
                                                  0C  13 000DE        BEQL    2$                                      ;
                                                  50  DD 000E0        PUSHL   R0                                      ; 2441
                                          40     AE  9F 000E2        PUSHAB  NAME_DESC                               ;
                                                 01  DD 000E5        PUSHL   #1                                      ;
                                                 58  DD 000E7        PUSHL   R8                                      ;
                                 63                  04  FB 000E9    CALLS   #4, LIB$SIGNAL                          ;
                                 54                  65  E8 000EC 2$: BLBS    FOUND_MATCH, 5$                         ; 2447
                                                     62  DD 000EF    PUSHL   IDENT                                   ; 2451
                          00000000G       00         01  FB 000F1    CALLS   #1, SYS$REM_IDENT                       ;
                                 66               50  D0 000F8        MOVL    R0, STATUS                              ;
                                 0E               50  E8 000FB        BLBS    R0, 3$                                  ; 2453
                                                  50  DD 000FE        PUSHL   R0                                      ; 2456
                                          40     AE  9F 00100        PUSHAB  NAME_DESC                               ;
                                                 01  DD 00103        PUSHL   #1                                      ;
                                                 58  DD 00105        PUSHL   R8                                      ;
                                 63                  04  FB 00107    CALLS   #4, LIB$SIGNAL                          ;
                                                     3C  11 0010A    BRB     6$                                      ; 2457
                          00000000G       00         01  90 0010C 3$: MOVB    #1, RIGHTSLIST_MODIFIED                 ; 2461
                                       C0 8F      03  A2  93 00113    BITB    IDENT+3, #192                           ; 2462
                                                      19  12 00118    BNEQ    4$                                      ;
                                 7E                   62  3C 0011A    MOVZWL  IDENT, -(SP)                            ; 2465
        7E      02  A2                  0E           00  EF 0011D    EXTZV   #0, #14, IDENT+2, -(SP)                 ; 2464
                                          44     AE  9F 00123        PUSHAB  NAME_DESC                               ; 2463
                                                 03  DD 00126        PUSHL   #3                                      ;
                                    00000000G    8F  DD 00128        PUSHL   #UAF$_RDBREMMSGU                        ;
                                 63                  05  FB 0012E    CALLS   #5, LIB$SIGNAL                          ;
                                                     10  11 00131    BRB     5$                                      ;
                                                     62  DD 00133 4$: PUSHL   IDENT                                   ; 2467
                                          40     AE  9F 00135        PUSHAB  NAME_DESC                               ; 2466
                                                 02  DD 00138        PUSHL   #2                                      ;
                                    00000000G    8F  DD 0013A        PUSHL   #UAF$_RDBREMMSG                         ;
                                 63                  04  FB 00140    CALLS   #4, LIB$SIGNAL                          ;
```

```
                     05F4   C4              00  FB 00143 5$:     CALLS   #0, UAF$REMOVE_GRP_IDENT                ; 2476
                            50              01  D0 00148 6$:     MOVL    #1, R0                                  ; 2478
                                                04 0014B         RET                                             ; 2480

; Routine Size:  332 bytes,    Routine Base:  $CODE$ + 116B

; 2500          2481  1
```

```
;  2502          2482  1 %SBTTL ' UAF$RENAME_IDENT  -  Modify RDB record'
;  2503          2483  1
;  2504          2484  1 GLOBAL ROUTINE UAF$RENAME_IDENT : NOVALUE =
;  2505          2485  2 BEGIN
;  2506          2486  2 !++
;  2507          2487  2 !
;  2508          2488  2 ! FUNCTIONAL DESCRIPTION:
;  2509          2489  2 !
;  2510          2490  2 !       This routine is used to change the name of an identifier in the
;  2511          2491  2 !       rights data base.  It is invoked via the RENAME/IDENT command.
;  2512          2492  2 !       This routine performs the same function as the MODIFY/IDENT/NAME
;  2513          2493  2 !       command
;  2514          2494  2 !
;  2515          2495  2 ! INPUTS:
;  2516          2496  2 !
;  2517          2497  2 !       none
;  2518          2498  2 !
;  2519          2499  2 ! IMPLICIT INPUTS:
;  2520          2500  2 !
;  2521          2501  2 !       command line
;  2522          2502  2 !
;  2523          2503  2 ! OUTPUTS:
;  2524          2504  2 !
;  2525          2505  2 !       None
;  2526          2506  2 !
;  2527          2507  2 ! IMPLICIT OUTPUTS:
;  2528          2508  2 !
;  2529          2509  2 !       none
;  2530          2510  2 !
;  2531          2511  2 ! ROUTINE VALUE:
;  2532          2512  2 !
;  2533          2513  2 !       none
;  2534          2514  2 !
;  2535          2515  2 ! SIDE EFFECTS:
;  2536          2516  2 !
;  2537          2517  2 !       rights data base is modified
;  2538          2518  2 !
;  2539          2519  2 !--
;  2540          2520  2
;  2541          2521  2
;  2542          2522  2 LOCAL
;  2543          2523  2     NAME_DESC           : STATDESC,
;  2544          2524  2     NAME_BUFF           : VECTOR [ KGB$S_NAME, BYTE ] ,
;  2545          2525  2     NEW_NAME_DESC       : STATDESC,
;  2546          2526  2     NEW_NAME_BUFF       : VECTOR [ KGB$S_NAME, BYTE ] ;
;  2547          2527  2
;  2548          2528  2 IDENT = 0 ;
;  2549          2529  2
;  2550          2530  2 !
;  2551          2531  2 ! First we will get the identifier name.
;  2552          2532  2 ! This is a required parameter so it is guaranteed present
;  2553          2533  2 !
;  2554          2534  2 CLI$GET_VALUE ( SD_TOKEN1, TOKENDSC ) ;
;  2555          2535  2
;  2556          2536  2 CH$COPY ( .TOKENLEN, .TOKENPTR, %C' ', KGB$S_NAME, NAME_BUFF );
;  2557          2537  2 NAME_DESC[LENGTH]  = MIN ( .TOKENLEN , KGB$S_NAME ) ;
;  2558          2538  2 NAME_DESC[POINTER] = NAME_BUFF ;
```

```
  2
  2 !
  2 ! Convert the ID from ASCII to a longword format
  2 !
P 2 STATUS = $ASCTOID ( NAME   = NAME_DESC,
P 2                     ID     = IDENT,
  2                     ATTRIB = 0 ) ;
  2 IF NOT .STATUS
  2     THEN
  3         BEGIN
  3         LIB$SIGNAL(UAF$_RDBMDFYERR, 1, NAME_DESC, .STATUS);
  3         RETURN ;
  2         END ;
  2
  2 !
  2 ! Now get the new name
  2 !
  2
  2 CLI$GET_VALUE ( SD_TOKEN2, TOKENDSC ) ;
  2 CH$COPY ( .TOKENLEN, .TOKENPTR, %C' ', KGB$S_NAME,  NEW_NAME_BUFF );
  2 NEW_NAME_DESC[LENGTH]  = MIN ( .TOKENLEN , KGB$S_NAME ) ;
  2 NEW_NAME_DESC[POINTER] = NEW_NAME_BUFF ;
  2
P 2 STATUS = $MOD_IDENT ( ID         = .IDENT,
  2                       NEW_NAME   = NEW_NAME_DESC ) ;
  2
  2 IF NOT .STATUS
  2     THEN LIB$SIGNAL(UAF$_RDBMDFYERR, 1, NAME_DESC, .STATUS)
  2     ELSE
  3         BEGIN
  3         RIGHTSLIST_MODIFIED = TRUE ;
  3         LIB$SIGNAL(UAF$_RDBMDFYMSG, 1, NAME_DESC);
  2         END ;
  2
  2 RETURN ;
  1 END ;                               ! End of UAF$RENAME_IDENT
```

```
                                 0FFC 00000     .ENTRY  UAF$RENAME_IDENT, Save R2,R3,R4,R5,R6,R7,-   ; 2484
                                                        R8,R9,R10,R11                                ;
           5B 00000000G    00   9E 00002        MOVAB   TOKENLEN, R11                                ;
           5A 00000000G    00   9E 00009        MOVAB   CLI$GET_VALUE, R10                           ;
           59 00000000G    00   9E 00010        MOVAB   TOKENDSC, R9                                 ;
           58 00000000'    00   9E 00017        MOVAB   IDENT, R8                                    ;
           57 00000000'    00   9E 0001E        MOVAB   STATUS, R7                                   ;
           5E        B0    AE   9E 00025        MOVAB   -80(SP), SP                                  ;
     48    AE 010E0000     8F   D0 00029        MOVL    #17694720, NAME_DESC                         ; 2523
                     4C    AE   D4 00031        CLRL    NAME_DESC+4                                  ;
     20    AE 010E0000     8F   D0 00034        MOVL    #17694720, NEW_NAME_DESC                     ; 2525
                     24    AE   D4 0003C        CLRL    NEW_NAME_DESC+4                              ;
                           68   D4 0003F        CLRL    IDENT                                        ; 2528
                           59   DD 00041        PUSHL   R9                                           ; 2534
                   0180    C7   9F 00043        PUSHAB  SD_TOKEN1                                    ;
           6A              02   FB 00047        CALLS   #2, CLI$GET_VALUE                            ;
```

```
                                56                  6B  3C 0004A          MOVZWL  TOKENLEN, R6                            ; 2536
                                50 00000000G        00  D0 0004D          MOVL    TOKENPTR, R0
            20                  20                  60  56  2C 00054      MOVC5   R6, (R0), #32, #32, NAME_BUFF           ;
                                                        28  AE   00059
                                50                      56  D0 0005B      MOVL    R6, R0                                  ; 2537
                                20                      50  B1 0005E      CMPW    R0, #32
                                                        03  1B 00061      BLEQU   1$                                      ;
                                50                      20  D0 00063      MOVL    #32, R0                                 ;
                          48    AE                      50  B0 00066 1$:  MOVW    R0, NAME_DESC                           ;
                          4C    AE              28      AE  9E 0006A      MOVAB   NAME_BUFF, NAME_DESC+4                  ; 2538
                                                        7E  D4 0006F      CLRL    -(SP)                                   ; 2545
                                                        58  DD 00071      PUSHL   R8                                      ;
                                                    50  AE  9F 00073      PUSHAB  NAME_DESC                               ;
                    00000000G   00                      03  FB 00076      CALLS   #3, SYS$ASCTOID                         ;
                                67                      50  D0 0007D      MOVL    R0, STATUS                              ;
                                42                      50  E9 00080      BLBC    R0, 3$                                  ; 2546
                                                        59  DD 00083      PUSHL   R9                                      ; 2557
                                                0188    C7  9F 00085      PUSHAB  SD_TOKEN2                               ;
                                6A                      02  FB 00089      CALLS   #2, CLI$GET_VALUE                       ;
                                56                      6B  3C 0008C      MOVZWL  TOKENLEN, R6                            ; 2558
                                50 00000000G            00  D0 0008F      MOVL    TOKENPTR, R0
            20                  20                  60  56  2C 00096      MOVC5   R6, (R0), #32, #32, NEW_NAME_BUFF       ;
                                                        6E   0009B
                                50                      56  D0 0009C      MOVL    R6, R0                                  ; 2559
                                20                      50  B1 0009F      CMPW    R0, #32
                                                        03  1B 000A2      BLEQU   2$                                      ;
                                50                      20  D0 000A4      MOVL    #32, R0                                 ;
                          20    AE                      50  B0 000A7 2$:  MOVW    R0, NEW_NAME_DESC                       ;
                          24    AE              6E      9E  9E 000AB      MOVAB   NEW_NAME_BUFF, NEW_NAME_DESC+4          ; 2560
                                                        7E  D4 000AF      CLRL    -(SP)                                   ; 2563
                                                    24  AE  9F 000B1      PUSHAB  NEW_NAME_DESC                           ;
                                                        7E  7C 000B4      CLRQ    -(SP)                                   ;
                                                        68  DD 000B6      PUSHL   IDENT                                   ;
                    00000000G   00                      05  FB 000B8      CALLS   #5, SYS$MOD_IDENT                       ;
                                67                      50  D0 000BF      MOVL    R0, STATUS                              ;
                                15                      50  E8 000C2      BLBS    R0, 4$                                  ; 2565
                                                        50  DD 000C5 3$:  PUSHL   R0                                      ; 2566
                                                    4C  AE  9F 000C7      PUSHAB  NAME_DESC                               ;
                                                        01  DD 000CA      PUSHL   #1                                      ;
                                            00000000G   8F  DD 000CC      PUSHL   #UAF$_RDBMDFYERR                        ;
                    00000000G   00                      04  FB 000D2      CALLS   #4, LIB$SIGNAL                          ;
                                                        04  000D9      RET
                    00000000G   00                      01  90 000DA 4$:  MOVB    #1, RIGHTSLIST_MODIFIED                 ; 2569
                                                    48  AE  9F 000E1      PUSHAB  NAME_DESC                               ; 2570
                                                        01  DD 000E4      PUSHL   #1                                      ;
                                            00000000G   8F  DD 000E6      PUSHL   #UAF$_RDBMDFYMSG                        ;
                    00000000G   00                      03  FB 000EC      CALLS   #3, LIB$SIGNAL                          ;
                                                        04  000F3      RET                                             ; 2574

; Routine Size:  244 bytes,    Routine Base:  $CODE$ + 12B7

; 2595          2575  1
; 2596          2576  1
```

```
;  2598          2577  1 %SBTTL ' UAF$REVOKE_IDENT  -  Delete RDB holder record '
;  2599          2578  1
;  2600          2579  1 GLOBAL ROUTINE UAF$REVOKE_IDENT: NOVALUE =
;  2601          2580  2 BEGIN
;  2602          2581  2 !++
;  2603          2582  2 !
;  2604          2583  2 ! FUNCTIONAL DESCRIPTION:
;  2605          2584  2 !
;  2606          2585  2 !       Remove a holder record from the rights data base for the specified
;  2607          2586  2 !       identifier.  This routine is invoked via the REVOKE/IDENT command.
;  2608          2587  2 !
;  2609          2588  2 ! INPUTS:
;  2610          2589  2 !
;  2611          2590  2 !       none
;  2612          2591  2 !
;  2613          2592  2 ! IMPLICIT INPUTS:
;  2614          2593  2 !
;  2615          2594  2 !       command line
;  2616          2595  2 !
;  2617          2596  2 ! OUTPUTS:
;  2618          2597  2 !
;  2619          2598  2 !       None
;  2620          2599  2 !
;  2621          2600  2 ! IMPLICIT OUTPUTS:
;  2622          2601  2 !
;  2623          2602  2 !       none
;  2624          2603  2 !
;  2625          2604  2 ! ROUTINE VALUE:
;  2626          2605  2 !
;  2627          2606  2 !       none
;  2628          2607  2 !
;  2629          2608  2 ! SIDE EFFECTS:
;  2630          2609  2 !
;  2631          2610  2 !       rights data base is modified
;  2632          2611  2 !
;  2633          2612  2 !--
;  2634          2613  2
;  2635          2614  2
;  2636          2615  2 BIND
;  2637          2616  2     HOLDER_VEC = HOLDER : VECTOR [2,LONG] ;
;  2638          2617  2
;  2639          2618  2 LOCAL
;  2640          2619  2     ID_NAME_DESC        : STATDESC,
;  2641          2620  2     ID_NAME_BUFF        : VECTOR [ KGB$S_NAME, BYTE ] ,
;  2642          2621  2     USER_NAME_DESC      : STATDESC,
;  2643          2622  2     USER_NAME_BUFF      : VECTOR [ KGB$S_NAME, BYTE ] ;
;  2644          2623  2
;  2645          2624  2
;  2646          2625  2 !
;  2647          2626  2 ! First we will get the identifier name.
;  2648          2627  2 ! This is a required parameter so it is guaranteed present
;  2649          2628  2 ! We will use the rights data base to translate the name
;  2650          2629  2 !
;  2651          2630  2 CLI$GET_VALUE ( SD_TOKEN1, TOKENDSC ) ;
;  2652          2631  2
;  2653          2632  2 CH$COPY ( .TOKENLEN, .TOKENPTR, %C' ', KGB$S_NAME, ID_NAME_BUFF );
;  2654          2633  2 ID_NAME_DESC[LENGTH] = MIN ( .TOKENLEN , KGB$S_NAME ) ;
```

```
;2655      2634  2 ID_NAME_DESC[POINTER] = ID_NAME_BUFF ;
;2656      2635  2
;2657      2636  2 !
;2658      2637  2 ! The user id parameter is required so we are guaranteed
;2659      2638  2 ! that it is present.  We will get the parameter before finishing
;2660      2639  2 ! up with the ID name so that we have it in case we must signal an
;2661      2640  2 ! error
;2662      2641  2 !
;2663      2642  2 PARSE_WILD( SD_TOKEN2, TRUE ) ;
;2664      2643  2 CH$COPY ( .TOKENLEN, .TOKENPTR, %C' ', KGB$S_NAME, USER_NAME_BUFF );
;2665      2644  2 USER_NAME_DESC[LENGTH]  = MIN ( .TOKENLEN , KGB$S_NAME ) ;
;2666      2645  2 USER_NAME_DESC[POINTER] = USER_NAME_BUFF ;
;2667      2646  2 HOLDER_VEC[1] = 0 ;
;2668      2647  2
;2669      2648  2 !
;2670      2649  2 ! Now finish up with the ID name
;2671      2650  2 ! Convert the ID from ASCII to a longword format
;2672      2651  2 !
;2673    P 2652  2 STATUS = $ASCTOID ( NAME    = ID_NAME_DESC,
;2674    P 2653  2                     ID      = IDENT,
;2675      2654  2                     ATTRIB = 0 ) ;
;2676      2655  2 IF NOT .STATUS
;2677      2656  2     THEN
;2678      2657  3         BEGIN
;2679      2658  3         LIB$SIGNAL(UAF$_REVOKEERR, 2, ID_NAME_DESC, USER_NAME_DESC, .STATUS);
;2680      2659  3         RETURN ;
;2681      2660  2         END ;
;2682      2661  2
;2683      2662  2 IF .IDENT[UIC$V_FORMAT] NEQ UIC$K_ID_FORMAT
;2684      2663  2    THEN
;2685      2664  3         BEGIN
;2686      2665  3         LIB$SIGNAL(UAF$_NOTIDFMT);
;2687      2666  3         RETURN ;
;2688      2667  2         END ;
;2689      2668  2
;2690      2669  2 IF (.IDENT LSSU UIC$K_FIRST_ID) OR
;2691      2670  3    (.IDENT GTRU UIC$K_LAST_ID)
;2692      2671  2     THEN
;2693      2672  3         BEGIN
;2694      2673  3         LIB$SIGNAL(UAF$_IDOUTRNG);
;2695      2674  3         RETURN ;
;2696      2675  2         END ;
;2697      2676  2
;2698      2677  2 !
;2699      2678  2 ! Now finish parsing the user ID entered
;2700      2679  2 ! We will use the rights data base to translate the name                    ;
;2701      2680  2 ! unless it was given in UIC format.  If it is a UIC then
;2702      2681  2 ! we will build a holder and make sure that it is in the rights
;2703      2682  2 ! data base.                                                               ;
;2704      2683  2 !
;2705      2684  2 !
;2706      2685  2 IF .UIC_FLAG
;2707      2686  2     THEN
;2708      2687  3         BEGIN
;2709      2688  3         LOCAL
;2710      2689  3             TEMP_NAME_DESC      : STATDESC,
;2711      2690  3             TEMP_NAME_BUFF      : VECTOR [ KGB$S_NAME, BYTE ] ;
```

```
;  2712          2691  3          IF NOT PARSE_UIC ( HOLDER[UIC$V_GROUP],
;  2713          2692  3                             HOLDER[UIC$V_MEMBER],
;  2714          2693  3                             TRUE )
;  2715          2694  3              THEN
;  2716          2695  4                  BEGIN
;  2717          2696  4                  LIB$SIGNAL(UAF$_UICERR, 2, .TOKENLEN, .TOKENPTR);
;  2718          2697  4                  RETURN ;
;  2719          2698  3                  END ;
;  2720          2699  3          HOLDER[UIC$V_FORMAT] = UIC$K_UIC_FORMAT ;
;  2721          2700  3          !
;  2722          2701  3          ! Make sure that the holder exists
;  2723          2702  3          !
;  2724          2703  3          TEMP_NAME_DESC[LENGTH]  = KGB$S_NAME ;
;  2725          2704  3          TEMP_NAME_DESC[POINTER] = TEMP_NAME_BUFF ;
;  2726        P 2705  3          STATUS = $IDTOASC ( ID     = .HOLDER_VEC[0],
;  2727        P 2706  3                              NAMLEN = TEMP_NAME_DESC[LENGTH],
;  2728          2707  3                              NAMBUF = TEMP_NAME_DESC ) ;
;  2729          2708  3          IF NOT .STATUS
;  2730          2709  3              THEN
;  2731          2710  4                  BEGIN
;  2732          2711  4                  LIB$SIGNAL(UAF$_REVOKEERR, 2, ID_NAME_DESC, USER_NAME_DESC,
;  2733          2712  4                                .STATUS);
;  2734          2713  4                  RETURN ;
;  2735          2714  4                  END
;  2736          2715  3              ELSE
;  2737          2716  4                  BEGIN
;  2738          2717  4                  CH$COPY (.TEMP_NAME_DESC[LENGTH], .TEMP_NAME_DESC[POINTER],
;  2739          2718  4                           %C' ', .USER_NAME_DESC[LENGTH], .USER_NAME_DESC[POINTER] ) ;
;  2740          2719  4                  USER_NAME_DESC[LENGTH] = .TEMP_NAME_DESC[LENGTH] ;
;  2741          2720  3                  END ;
;  2742          2721  3          END
;  2743          2722  2      ELSE
;  2744          2723  3          BEGIN
;  2745          2724  3          IF .STR_WILD
;  2746          2725  3              THEN
;  2747          2726  4                  BEGIN
;  2748          2727  4                  LIB$SIGNAL(UAF$_WLDNOTALWD);
;  2749          2728  4                  RETURN ;
;  2750          2729  3                  END ;
;  2751          2730  3
;  2752          2731  3          !
;  2753          2732  3          ! Convert the user ID from ASCII to a longword format
;  2754          2733  3          !
;  2755        P 2734  3          STATUS = $ASCTOID ( NAME    = USER_NAME_DESC,
;  2756        P 2735  3                              ID      = HOLDER_VEC[0],
;  2757          2736  3                              ATTRIB = 0 ) ;
;  2758          2737  3          IF NOT .STATUS
;  2759          2738  3              THEN
;  2760          2739  4                  BEGIN
;  2761          2740  4                  LIB$SIGNAL(UAF$_REVOKEERR, 2, ID_NAME_DESC, USER_NAME_DESC,
;  2762          2741  4                                .STATUS);
;  2763          2742  4                  RETURN ;
;  2764          2743  3                  END ;
;  2765          2744  3
;  2766          2745  3          IF .HOLDER[UIC$V_FORMAT] NEQ UIC$K_UIC_FORMAT
;  2767          2746  3             THEN
;  2768          2747  4                  BEGIN
```

```
;  2769          2748  4                 LIB$SIGNAL(UAF$_NOTUICFMT);
;  2770          2749  4                 RETURN ;
;  2771          2750  3                 END ;
;  2772          2751  3
;  2773          2752  2         END ;
;  2774          2753  2
;  2775          2754  2
;  2776          2755  2 !
;  2777          2756  2 ! Now call the service to perform the addition
;  2778          2757  2 !
;  2779        P 2758  2 STATUS = $REM_HOLDER ( ID     = .IDENT,
;  2780          2759  2                        HOLDER = HOLDER ) ;
;  2781          2760  2
;  2782          2761  2 IF NOT .STATUS
;  2783          2762  2     THEN LIB$SIGNAL(UAF$_REVOKEERR, 2, ID_NAME_DESC, USER_NAME_DESC, .STATUS)
;  2784          2763  2     ELSE
;  2785          2764  3         BEGIN
;  2786          2765  3         RIGHTSLIST_MODIFIED = TRUE ;
;  2787          2766  3         LIB$SIGNAL(UAF$_REVOKEMSG, 2, ID_NAME_DESC, USER_NAME_DESC);
;  2788          2767  2         END ;
;  2789          2768  2
;  2790          2769  2 RETURN ;
;  2791          2770  2
;  2792          2771  1 END ;                                   !End of UAF$REVOKE_IDENT


                                                        HOLDER_VEC=         HOLDER
                                                                .EXTRN  SYS$REM_HOLDER

                                            0FFC 00000          .ENTRY  UAF$REVOKE_IDENT, Save R2,R3,R4,R5,R6,R7,-  ; 2579
                                                                        R8,R9,R10,R11
                                      5B 00000000G   00  9E 00002       MOVAB   LIB$SIGNAL, R11
                                      5A 00000000G   00  9E 00009       MOVAB   TOKENPTR, R10
                                      59 00000000G   00  9E 00010       MOVAB   TOKENLEN, R9
                                      58 00000000'   00  9E 00017       MOVAB   STATUS, R8
                                      57 00000000'   00  9E 0001E       MOVAB   HOLDER, R7
                                      5E         88  AE  9E 00025       MOVAB   -120(SP), SP
                                70    AE 010E0000  8F  D0 00029       MOVL    #17694720, ID_NAME_DESC                 ; 2619
                                           74      AE  D4 00031       CLRL    ID_NAME_DESC+4
                                48    AE 010E0000  8F  D0 00034       MOVL    #17694720, USER_NAME_DESC               ; 2621
                                           4C      AE  D4 0003C       CLRL    USER_NAME_DESC+4
                                      000000000G   00  9F 0003F       PUSHAB  TOKENDSC                                ; 2630
                                           0180    C8  9F 00045       PUSHAB  SD_TOKEN1
                           000000000G   00                02  FB 00049       CALLS   #2, CLI$GET_VALUE
                                      56                   69  3C 00050       MOVZWL  TOKENLEN, R6                    ; 2632
                                      50                   6A  D0 00053       MOVL    TOKENPTR, R0
                20         20         60                   56  2C 00056       MOVC5   R6, (R0), #32, #32, ID_NAME_BUFF
                                           50      AE          00058
                                      50                   56  D0 0005D       MOVL    R6, R0                          ; 2633
                                      20                   50  B1 00060       CMPW    R0, #32
                                                           03  1B 00063       BLEQU   1$
                                      50                   20  D0 00065       MOVL    #32, R0
                                70    AE                   50  B0 00068 1$:   MOVW    R0, ID_NAME_DESC
                                74    AE             50    AE  9E 0006C       MOVAB   ID_NAME_BUFF, ID_NAME_DESC+4    ; 2634
                                                           01  DD 00071       PUSHL   #1                              ; 2642
                                           0188    C8  9F 00073       PUSHAB  SD_TOKEN2
```

```
                                00000000G  00              02  FB 00077          CALLS   #2, PARSE_WILD
                                           56              69  3C 0007E          MOVZWL  TOKENLEN, R6                                    ; 2643
                                           50              6A  D0 00081          MOVL    TOKENPTR, R0
           20                20            60              56  2C 00084          MOVC5   R6, (R0), #32, #32, USER_NAME_BUFF
                                                     28    AE      00089
                                           50              56  D0 0008B          MOVL    R6, R0                                          ; 2644
                                           20              50  B1 0008E          CMPW    R0, #32
                                                           03  1B 00091          BLEQU   2$
                                           50              20  D0 00093          MOVL    #32, R0
                                     48    AE              50  B0 00096 2$:      MOVW    R0, USER_NAME_DESC
                                     4C    AE        28    AE  9E 0009A          MOVAB   USER_NAME_BUFF, USER_NAME_DESC+4                ; 2645
                                                     04    A7  D4 0009F          CLRL    HOLDER_VEC+4                                    ; 2646
                                                           7E  D4 000A2          CLRL    -(SP)                                           ; 2654
                                                     08    A7  9F 000A4          PUSHAB  IDENT
                                                     78    AE  9F 000A7          PUSHAB  ID_NAME_DESC
                                00000000G  00              03  FB 000AA          CALLS   #3, SYS$ASCTOID
                                           68              50  D0 000B1          MOVL    R0, STATUS
                                           03              50  E8 000B4          BLBS    R0, 3$                                          ; 2655
                                                        00DE   31 000B7          BRW     14$
           02        0B    A7            02                 06  ED 000BA 3$:      CMPZV   #6, #2, IDENT+3, #2                             ; 2662
                                                           08  13 000C0          BEQL    4$
                                                00000000G  8F  DD 000C2          PUSHL   #UAF$_NOTIDFMT                                  ; 2665
                                                           1C  11 000C8          BRB     6$
                                           50        08    A7  D0 000CA 4$:      MOVL    IDENT, R0                                       ; 2669
                                80010000   8F              50  D1 000CE          CMPL    R0, #-2147418112
                                                           09  1F 000D5          BLSSU   5$
                                8FFFFFFF   8F              50  D1 000D7          CMPL    R0, #-1879048193                                ; 2670
                                                           08  1B 000DE          BLEQU   7$
                                                00000000G  8F  DD 000E0 5$:      PUSHL   #UAF$_IDOUTRNG                                  ; 2673
                                                           77  11 000E6 6$:      BRB     10$
                                           63 00000000G    00  E9 000E8 7$:      BLBC    UIC_FLAG, 9$                                    ; 2685
                                     20    AE 010E0000     8F  D0 000EF          MOVL    #17894720, TEMP_NAME_DESC                       ; 2689
                                                     24    AE  D4 000F7          CLRL    TEMP_NAME_DESC+4
                                                           01  DD 000FA          PUSHL   #1                                              ; 2692
                                                           57  DD 000FC          PUSHL   R7
                                                     02    A7  9F 000FE          PUSHAB  HOLDER+2                                        ; 2691
                                00000000G  00              03  FB 00101          CALLS   #3, PARSE_UIC                                   ; 2692
                                           10              50  E8 00108          BLBS    R0, 8$
                                                           6A  DD 0010B          PUSHL   TOKENPTR                                        ; 2696
                                           7E              69  3C 0010D          MOVZWL  TOKENLEN, -(SP)
                                                           02  DD 00110          PUSHL   #2
                                                00000000G  8F  DD 00112          PUSHL   #UAF$_UICERR
                                                        00A6   31 00118          BRW     16$
                                     03    A7        C0    8F  8A 0011B 8$:      BICB2   #192, HOLDER+3                                  ; 2699
                                     20    AE              20  B0 00120          MOVW    #32, TEMP_NAME_DESC                             ; 2703
                                     24    AE              6E  9E 00124          MOVAB   TEMP_NAME_BUFF, TEMP_NAME_DESC+4                ; 2704
                                                           7E  7C 00128          CLRQ    -(SP)                                           ; 2707
                                                           7E  D4 0012A          CLRL    -(SP)
                                                     2C    AE  9F 0012C          PUSHAB  TEMP_NAME_DESC
                                                     30    AE  9F 0012F          PUSHAB  TEMP_NAME_DESC
                                                           67  DD 00132          PUSHL   HOLDER_VEC
                                00000000G  00              06  FB 00134          CALLS   #6, SYS$IDTOASC
                                           68              50  D0 0013B          MOVL    R0, STATUS
                                           57              50  E9 0013E          BLBC    R0, 14$                                         ; 2708
  48   AE              20        24    BE              20  AE  2C 00141          MOVC5   TEMP_NAME_DESC, @TEMP_NAME_DESC+4, #32, -       ; 2718
                                                     4C    BE      00149                  USER_NAME_DESC, @USER_NAME_DESC+4
                                     48    AE        20  AE  B0 0014B          MOVW    TEMP_NAME_DESC, USER_NAME_DESC                  ; 2719
```

```
                                                34  11 00150          BRB     13$                                     ; 2685
                                             08 00000000G   00  E9 00152 9$:       BLBC    STR_WILD, 11$                           ; 2724
                                                00000000G   8F  DD 00159           PUSHL   #UAF$_WLDNOTALWD                        ; 2727
                                                21  11 0015F 10$:       BRB     12$
                                                7E  D4 00161 11$:       CLRL    -(SP)                                           ; 2736
                                                57  DD 00163           PUSHL   R7
                                                    50  AE  9F 00165           PUSHAB  USER_NAME_DESC
                                         00000000G  00      03  FB 00168           CALLS   #3, SYS$ASCTOID
                                                68          50  D0 0016F           MOVL    R0, STATUS
                                                23          50  E9 00172           BLBC    R0, 14$                                 ; 2737
                                              C0  8F    03  A7  93 00175           BITB    HOLDER+3, #192                          ; 2745
                                                0A  13 0017A           BEQL    13$
                                             00000000G  8F  DD 0017C           PUSHL   #UAF$_NOTUICFMT                         ; 2748
                                             6B             01  FB 00182 12$:       CALLS   #1, LIB$SIGNAL
                                                04 00185           RET                                                     ; 2747
                                                57  DD 00186 13$:       PUSHL   R7                                              ; 2759
                                                    08  A7  DD 00188           PUSHL   IDENT
                                         00000000G  00      02  FB 0018B           CALLS   #2, SYS$REM_HOLDER
                                                68          50  D0 00192           MOVL    R0, STATUS
                                                14          50  E8 00195           BLBS    R0, 15$                                 ; 2761
                                                50  DD 00198 14$:       PUSHL   R0                                              ; 2762
                                                    4C  AE  9F 0019A           PUSHAB  USER_NAME_DESC
                                                    78  AE  9F 0019D           PUSHAB  ID_NAME_DESC
                                                02  DD 001A0           PUSHL   #2
                                             00000000G  8F  DD 001A2           PUSHL   #UAF$_REVOKEERR
                                             6B             05  FB 001A8           CALLS   #5, LIB$SIGNAL
                                                04 001AB           RET
                                         00000000G  00      01  90 001AC 15$:       MOVB    #1, RIGHTSLIST_MODIFIED                 ; 2765
                                                    48  AE  9F 001B3           PUSHAB  USER_NAME_DESC                          ; 2766
                                                    74  AE  9F 001B6           PUSHAB  ID_NAME_DESC
                                                02  DD 001B9           PUSHL   #2
                                             00000000G  8F  DD 001BB           PUSHL   #UAF$_REVOKEMSG
                                             6B             04  FB 001C1 16$:       CALLS   #4, LIB$SIGNAL
                                                04 001C4           RET                                                     ; 2771

; Routine Size:  453 bytes,    Routine Base:  $CODE$ + 13AB

; 2793          2772  1
```

```
;  2795          2773  1 %SBTTL ' UAF$SHOW_IDENT  -  Display rights identifiers '
;  2796          2774  1
;  2797          2775  1 GLOBAL ROUTINE UAF$SHOW_IDENT : NOVALUE =
;  2798          2776  2 BEGIN
;  2799          2777  2 !++
;  2800          2778  2 !
;  2801          2779  2 ! FUNCTIONAL DESCRIPTION:
;  2802          2780  2 !
;  2803          2781  2 !       Write all identifiers to SYS$OUTPUT.  This routine is invoked
;  2804          2782  2 !       via the SHOW/IDENT command
;  2805          2783  2 !
;  2806          2784  2 ! INPUTS:
;  2807          2785  2 !
;  2808          2786  2 !       none
;  2809          2787  2 !
;  2810          2788  2 ! IMPLICIT INPUTS:
;  2811          2789  2 !
;  2812          2790  2 !       command line
;  2813          2791  2 !
;  2814          2792  2 ! OUTPUTS:
;  2815          2793  2 !
;  2816          2794  2 !       None
;  2817          2795  2 !
;  2818          2796  2 ! IMPLICIT OUTPUTS:
;  2819          2797  2 !
;  2820          2798  2 !       none
;  2821          2799  2 !
;  2822          2800  2 ! ROUTINE VALUE:
;  2823          2801  2 !
;  2824          2802  2 !       none
;  2825          2803  2 !
;  2826          2804  2 ! SIDE EFFECTS:
;  2827          2805  2 !
;  2828          2806  2 !       none
;  2829          2807  2 !--
;  2830          2808  2
;  2831          2809  2
;  2832          2810  2 !
;  2833          2811  2 ! Set the correct RAB for the output and flag a show operation
;  2834          2812  2 !
;  2835          2813  2 RABPTR = OUTRAB ;
;  2836          2814  2 RDB_LIST_FLAG = FALSE ;
;  2837          2815  2
;  2838          2816  2 !
;  2839          2817  2 ! Set the type of display.  This is the same no matter
;  2840          2818  2 ! what parameters or qualifiers are used
;  2841          2819  2 !
;  2842          2820  3 IF CLI$PRESENT (SD_FULL) OR (NOT CLI$PRESENT (SD_BRIEF))
;  2843          2821  2     THEN SHOW_ID_FULL = TRUE
;  2844          2822  2     ELSE SHOW_ID_FULL = FALSE ;
;  2845          2823  2
;  2846          2824  2 !
;  2847          2825  2 ! Now call the common routine for show or list
;  2848          2826  2 !
;  2849          2827  2 UAF$DISPLAY_IDENT () ;
;  2850          2828  2
;  2851          2829  2 RETURN
```

```
; 2852          2830  2
; 2853          2831  1 END ;                                   ! End of UAF$SHOW_IDENT


                                                 000C 00000          .ENTRY  UAF$SHOW_IDENT, Save R2,R3                  ; 2775
                                      53 00000000G   00   9E 00002   MOVAB   CLI$PRESENT, R3                             ;
                                      52 00000000'   00   9E 00009   MOVAB   SHOW_ID_FULL, R2                           ;
                           00000000G  00 00000000G   00   9E 00010   MOVAB   OUTRAB, RABPTR                              ; 2813
                                      00000000'      00   94 0001B   CLRB    RDB_LIST_FLAG                             ; 2814
                                           0164      C2   9F 00021   PUSHAB  SD_FULL                                   ; 2820
                                      63             01   FB 00025   CALLS   #1, CLI$PRESENT                          ;
                                      0A             50   E8 00028   BLBS    R0, 1$                                    ;
                                           015C      C2   9F 0002B   PUSHAB  SD_BRIEF                                  ;
                                      63             01   FB 0002F   CALLS   #1, CLI$PRESENT                          ;
                                      05             50   E8 00032   BLBS    R0, 2$                                    ;
                                      62             01   90 00035 1$:  MOVB    #1, SHOW_ID_FULL                          ; 2821
                                                     02   11 00038   BRB     3$                                        ;
                                                     62   94 0003A 2$:  CLRB    SHOW_ID_FULL                              ; 2822
                                EF68  CF             00   FB 0003C 3$:  CALLS   #0, UAF$DISPLAY_IDENT                  ; 2827
                                                        04 00041         RET                                           ; 2831

; Routine Size:  66 bytes,    Routine Base:  $CODE$ + 1570

; 2854          2832  1
```

```
;  2856          2833  1 %SBTTL ' UAF$SHOW_RIGHTS - Display rights held by user'
;  2857          2834  1
;  2858          2835  1 GLOBAL ROUTINE UAF$SHOW_RIGHTS : NOVALUE =
;  2859          2836  2 BEGIN
;  2860          2837  2 !++
;  2861          2838  2 !
;  2862          2839  2 ! FUNCTIONAL DESCRIPTION:
;  2863          2840  2 !
;  2864          2841  2 !       Write all identifiers held by a specified identifier
;  2865          2842  2 !       to SYS$OUTPUT.  This routine is invoked
;  2866          2843  2 !       via the SHOW/RIGHTS command
;  2867          2844  2 !
;  2868          2845  2 ! INPUTS:
;  2869          2846  2 !
;  2870          2847  2 !       none
;  2871          2848  2 !
;  2872          2849  2 ! IMPLICIT INPUTS:
;  2873          2850  2 !
;  2874          2851  2 !       command line
;  2875          2852  2 !
;  2876          2853  2 ! OUTPUTS:
;  2877          2854  2 !
;  2878          2855  2 !       None
;  2879          2856  2 !
;  2880          2857  2 ! IMPLICIT OUTPUTS:
;  2881          2858  2 !
;  2882          2859  2 !       none
;  2883          2860  2 !
;  2884          2861  2 ! ROUTINE VALUE:
;  2885          2862  2 !
;  2886          2863  2 !       none
;  2887          2864  2 !
;  2888          2865  2 ! SIDE EFFECTS:
;  2889          2866  2 !
;  2890          2867  2 !       none
;  2891          2868  2 !--
;  2892          2869  2
;  2893          2870  2
;  2894          2871  2 !
;  2895          2872  2 ! Select the output RAB and flag a show operation
;  2896          2873  2 !
;  2897          2874  2 RABPTR = OUTRAB ;
;  2898          2875  2 RDB_LIST_FLAG = FALSE ;
;  2899          2876  2
;  2900          2877  2 !
;  2901          2878  2 ! Call common routine to perform show or list
;  2902          2879  2 !
;  2903          2880  2 UAF$DISPLAY_RIGHTS () ;
;  2904          2881  2
;  2905          2882  2 RETURN ;
;  2906          2883  2
;  2907          2884  1 END ;                                   !End of UAF$SHOW_RIGHTS
```

```
                                    0000 00000         .ENTRY  UAF$SHOW_RIGHTS, Save nothing                ; 2835
                00000000G  00 00000000G  00  9E 00002  MOVAB   OUTRAB, RABPTR                               ; 2874
                           00000000'     00  94 0000D  CLRB    RDB_LIST_FLAG                               ; 2875
                    F253   CF             00  FB 00013  CALLS   #0, UAF$DISPLAY_RIGHTS                     ; 2880
                                               04 00018  RET                                               ; 2884

; Routine Size:  25 bytes,    Routine Base:  $CODE$ + 15B2

; 2908          2885  1
```

```
;2910   2886  1 %SBTTL ' UAF$WRITE_HOLDERS - Display holders of specified ident'
;2911   2887  1
;2912   2888  1 ROUTINE UAF$WRITE_HOLDERS ( IDENT ) =
;2913   2889  2 BEGIN
;2914   2890  2 !++
;2915   2891  2 !
;2916   2892  2 ! FUNCTIONAL DESCRIPTION:
;2917   2893  2 !
;2918   2894  2 !       Write all holders of the specified identifier to the output
;2919   2895  2 !       file.
;2920   2896  2 !
;2921   2897  2 ! INPUTS:
;2922   2898  2 !
;2923   2899  2 !       IDENT   value of identifier to find holders of
;2924   2900  2 !
;2925   2901  2 ! IMPLICIT INPUTS:
;2926   2902  2 !
;2927   2903  2 !       RABPTR must point to the output RAB
;2928   2904  2 !
;2929   2905  2 ! OUTPUTS:
;2930   2906  2 !
;2931   2907  2 !       None
;2932   2908  2 !
;2933   2909  2 ! IMPLICIT OUTPUTS:
;2934   2910  2 !
;2935   2911  2 !       none
;2936   2912  2 !
;2937   2913  2 ! ROUTINE VALUE:
;2938   2914  2 !
;2939   2915  2 !       status of rights data base and write operations
;2940   2916  2 !
;2941   2917  2 ! SIDE EFFECTS:
;2942   2918  2 !
;2943   2919  2 !       The output file is written to
;2944   2920  2 !
;2945   2921  2 !--
;2946   2922  2
;2947   2923  2
;2948   2924  2 $ASSUME (KGB$S_NAME, EQL, 32 ) ; ! If length changes HOLDERLINE must also be fixed
;2949   2925  2
;2950   2926  2 LITERAL
;2951   2927  2     ATTRIB_BUFFLEN      = 15 ;
;2952   2928  2
;2953   2929  2 BIND
;2954   2930  2     HOLDERHEAD          = CSTRING ('    !32AF !AS') ,
;2955   2931  2     HOLDERLINE          = CSTRING ('    !32AF !AS') ,
;2956   2932  2     HOLDER_VEC          = HOLDER : VECTOR [2,LONG] ;
;2957   2933  2
;2958   2934  2 LOCAL
;2959   2935  2     CONTINUE            : BYTE ,
;2960   2936  2     CONTEXT             : LONG INITIAL (0) ,
;2961   2937  2     ATTRIB_DESC         : STATDESC,
;2962   2938  2     ATTRIB_BUFF         : VECTOR [ ATTRIB_BUFFLEN, BYTE ] ,
;2963   2939  2     NAME_DESC           : STATDESC,
;2964   2940  2     NAME_BUFF           : VECTOR [ KGB$S_NAME, BYTE ] ,
;2965   2941  2     SD_ATTRIBUTES       : QUALSTR_DESC ('Attributes'),
;2966   2942  2     SD_HOLDER           : QUALSTR_DESC ('Holder'),
```

```
;  2967          2943  2     SD_RESOURCE          : QUALSTR_DESC ('RESOURCE'),
;  2968          2944  2     SD_NORESOURCE        : QUALSTR_DESC ('NORESOURCE') ;
;  2969          2945  2
;  2970        P 2946  2 STATUS = $FIND_HOLDER ( ID      = .IDENT,
;  2971        P 2947  2                         HOLDER  = HOLDER,
;  2972        P 2948  2                         ATTRIB  = ATTRIBUTES,
;  2973          2949  2                         CONTXT  = CONTEXT ) ;
;  2974          2950  2 IF NOT .STATUS
;  2975          2951  2     THEN
;  2976          2952  3         BEGIN
;  2977          2953  3         $FINISH_RDB ( CONTXT = CONTEXT ) ;
;  2978          2954  3         IF .STATUS EQL SS$_NOSUCHID
;  2979          2955  3             THEN RETURN SS$_NORMAL
;  2980          2956  3             ELSE RETURN .STATUS ;
;  2981          2957  3         END
;  2982          2958  2     ELSE
;  2983          2959  3         BEGIN
;  2984          2960  3         !
;  2985          2961  3         ! Print a header
;  2986          2962  3         !
;  2987        P 2963  3         FAOMAC ( HOLDERHEAD,
;  2988        P 2964  3                  .SD_HOLDER[LENGTH], .SD_HOLDER[POINTER],
;  2989          2965  3                  SD_ATTRIBUTES ) ;
;  2990          2966  2         END ;
;  2991          2967  2
;  2992          2968  2
;  2993          2969  2 !
;  2994          2970  2 ! Loop until we find all the identifiers held by this guy
;  2995          2971  2 !
;  2996          2972  2 CONTINUE = TRUE ;
;  2997          2973  2 DO
;  2998          2974  3     BEGIN
;  2999          2975  3     !
;  3000          2976  3     ! Init the name descriptor
;  3001          2977  3     ! and find the ASCII name of the identifier
;  3002          2978  3     !
;  3003          2979  3     NAME_DESC[POINTER] = NAME_BUFF ;
;  3004          2980  3     NAME_DESC[LENGTH]  = KGB$S_NAME ;
;  3005          2981  3     CH$FILL ( %C' ', .NAME_DESC[LENGTH], .NAME_DESC[POINTER] ) ;
;  3006        P 2982  3     STATUS = $IDTOASC ( ID     = .HOLDER_VEC[0],
;  3007        P 2983  3                         NAMLEN = NAME_DESC[LENGTH],
;  3008          2984  3                         NAMBUF = NAME_DESC ) ;
;  3009          2985  3
;  3010          2986  3     !
;  3011          2987  3     ! Build the attribute string
;  3012          2988  3     !
;  3013          2989  3     CH$FILL ( %C' ', ATTRIB_BUFFLEN, ATTRIB_BUFF ) ;
;  3014          2990  3     ATTRIB_DESC[LENGTH]  = 0 ;
;  3015          2991  3     ATTRIB_DESC[POINTER] = ATTRIB_BUFF ;
;  3016          2992  4     IF ( .ATTRIBUTES AND KGB$M_RESOURCE )
;  3017          2993  3         THEN
;  3018          2994  4             BEGIN
;  3019          2995  4             CH$MOVE ( .SD_RESOURCE[LENGTH], .SD_RESOURCE[POINTER],
;  3020          2996  4                       ATTRIB_BUFF[.ATTRIB_DESC[LENGTH]] ) ;
;  3021          2997  4             ATTRIB_DESC[LENGTH] =
;  3022          2998  4               .ATTRIB_DESC[LENGTH] + .SD_RESOURCE[LENGTH] + 1 ;
;  3023          2999  4             END
```

```
;  3024          3000  3          ELSE
;  3025          3001  4              BEGIN
;  3026          3002  4              CH$MOVE ( .SD_NORESOURCE[LENGTH], .SD_NORESOURCE[POINTER],
;  3027          3003  4                        ATTRIB_BUFF[.ATTRIB_DESC[LENGTH]] ) ;
;  3028          3004  4              ATTRIB_DESC[LENGTH] =
;  3029          3005  4                .ATTRIB_DESC[LENGTH] + .SD_NORESOURCE[LENGTH] + 1 ;
;  3030          3006  3              END ;
;  3031          3007  3
;  3032          3008  3      !
;  3033          3009  3      ! Output the line
;  3034          3010  3      !
;  3035        P 3011  3      FAOMAC ( HOLDERLINE,
;  3036        P 3012  3               .NAME_DESC[LENGTH], .NAME_DESC[POINTER],    ! Identifier name
;  3037          3013  3               ATTRIB_DESC ) ;                             ! Attributes
;  3038          3014  3
;  3039        P 3015  3      STATUS = $FIND_HOLDER ( ID      = .IDENT,
;  3040        P 3016  3                              HOLDER  = HOLDER,
;  3041        P 3017  3                              ATTRIB  = ATTRIBUTES,
;  3042          3018  3                              CONTXT  = CONTEXT ) ;
;  3043          3019  3      IF NOT .STATUS
;  3044          3020  3          THEN
;  3045          3021  4              BEGIN
;  3046          3022  4              CONTINUE = FALSE ;
;  3047          3023  4              IF .STATUS EQL SS$_NOSUCHID
;  3048          3024  4                  THEN STATUS = SS$_NORMAL ;
;  3049          3025  3              END ;
;  3050          3026  3
;  3051          3027  3      END
;  3052          3028  2 WHILE .CONTINUE;
;  3053          3029  2
;  3054          3030  2 $FINISH_RDB ( CONTXT = CONTEXT ) ;
;  3055          3031  2
;  3056          3032  2 RETURN .STATUS ;
;  3057          3033  2
;  3058          3034  1 END ;                                   ! End of UAF$WRITE_HOLDERS


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                        0D  0007F P.AAQ:  .BYTE   13
    53  41  21  20  46  41  32  33  21  20  20  20     20  00080         .ASCII  \    !32AF !AS\                    ;
                                                        0D  0008D P.AAR:  .BYTE   13
    53  41  21  20  46  41  32  33  21  20  20  20     20  0008E         .ASCII  \    !32AF !AS\                    ;
                73  65  74  75  62  69  72  74  74  41      0009B P.AAS:  .ASCII  \Attributes\                      ;
                            72  65  64  6C  6F  48          000A5 P.AAT:  .ASCII  \Holder\                          ;
                        45  43  52  55  4F  53  45  52      000AB P.AAU:  .ASCII  \RESOURCE\                        ;
                45  43  52  55  4F  53  45  52  4F  4E      000B3 P.AAV:  .ASCII  \NORESOURCE\                      ;

                                                        HOLDERHEAD=         P.AAQ
                                                        HOLDERLINE=         P.AAR
                                                        HOLDER_VEC=         HOLDER
                                                                .EXTRN  SYS$FIND_HOLDER
                                                                .EXTRN  SYS$FINISH_RDB, SYS$PUT

                                                                .PSECT  $CODE$,NOWRT,2
```

```
                                                0FFC 00000 UAF$WRITE_HOLDERS:
                                                                .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11            ; 2888
                                5B 00000000G    00  9E 00002        MOVAB   RABPTR, R11
                                5A 00000000G    00  9E 00009        MOVAB   FAODSC, R10
                                59 00000000'    00  9E 00010        MOVAB   ATTRIBUTES, R9
                                58 00000000'    00  9E 00017        MOVAB   STATUS, R8
                                57 00000000'    00  9E 0001E        MOVAB   P.AAS, R7
                                5E      A0      AE  9E 00025        MOVAB   -96(SP), SP
                                                7E  D4 00029        CLRL    CONTEXT                                 ; 2889
                             5C AE 010E0000     8F  D0 0002B        MOVL    #17694720, ATTRIB_DESC                  ; 2937
                                       60       AE  D4 00033        CLRL    ATTRIB_DESC+4
                             44 AE 010E0000     8F  D0 00036        MOVL    #17694720, NAME_DESC                    ; 2939
                                       48       AE  D4 0003E        CLRL    NAME_DESC+4
                             1C AE 010E000A     8F  D0 00041        MOVL    #17694730, SD_ATTRIBUTES                ; 2941
                             20 AE              67  9E 00049        MOVAB   P.AAS, SD_ATTRIBUTES+4
                             14 AE 010E0006     8F  D0 0004D        MOVL    #17694726, SD_HOLDER                    ; 2942
                             18 AE        0A    A7  9E 00055        MOVAB   P.AAT, SD_HOLDER+4
                             0C AE 010E0008     8F  D0 0005A        MOVL    #17694728, SD_RESOURCE                  ; 2943
                             10 AE        10    A7  9E 00062        MOVAB   P.AAU, SD_RESOURCE+4
                             04 AE 010E000A     8F  D0 00067        MOVL    #17694730, SD_NORESOURCE                ; 2944
                             08 AE        18    A7  9E 0006F        MOVAB   P.AAV, SD_NORESOURCE+4
                                      4200      8F  BB 00074        PUSHR   #^M<R9,SP>                              ; 2949
                                        04      A9  9F 00078        PUSHAB  HOLDER
                                        04      AC  DD 0007B        PUSHL   IDENT
                          00000000G     00          FB 0007E        CALLS   #4, SYS$FIND_HOLDER
                                        68          D0 00085        MOVL    R0, STATUS
                                        1A          E8 00088        BLBS    STATUS, 2$                              ; 2950
                                                    DD 0008B        PUSHL   SP                                      ; 2953
                          00000000G     00          FB 0008D        CALLS   #1, SYS$FINISH_RDB
                                        50          D0 00094        MOVL    STATUS, R0                              ; 2954
                             000021EC   8F          D1 00097        CMPL    R0, #8684
                                                    13 0009E        BEQL    1$
                                                    04 000A0        RET
                                        50          D0 000A1 1$:    MOVL    #1, R0                                  ; 2956
                                                    04 000A4        RET
                                        6A      E4  9B 000A5 2$:    MOVZBW  HOLDERHEAD, FAODSC                      ; 2965
                                   04   AA      E5  9E 000A9        MOVAB   HOLDERHEAD+1, FAODSC+4
                                        1C      AE  9F 000AE        PUSHAB  SD_ATTRIBUTES
                                        1C      AE  DD 000B1        PUSHL   SD_HOLDER+4
                                        7E      1C  3C 000B4        MOVZWL  SD_HOLDER, -(SP)
                                             00000000G  9F 000B8    PUSHAB  DISDSC
                                   7E           6B  C1 000BE        ADDL3   #34, RABPTR, -(SP)
                                                5A  DD 000C2        PUSHL   R10
                                00000000G       00  FB 000C4        CALLS   #6, SYS$FAO
                                                6B  DD 000CB        PUSHL   RABPTR
                                00000000G       00  FB 000CD        CALLS   #1, SYS$PUT
                                                56  90 000D4        MOVB    #1, CONTINUE                            ; 2972
                                   48   AE      24  9E 000D7 3$:    MOVAB   NAME_BUFF, NAME_DESC+4                  ; 2979
                                   44   AE          20  B0 000DC    MOVW    #32, NAME_DESC                          ; 2980
                          44   AE          20       6E  2C 000E0    MOVC5   #0, (SP), #32, NAME_DESC, @NAME_DESC+4  ; 2981
                                                48  BE  000E6
                                                    7E  7C 000E8    CLRQ    -(SP)                                   ; 2984
                                                    7E  D4 000EA    CLRL    -(SP)
                                                50  AE  9F 000EC    PUSHAB  NAME_DESC
                                                54  AE  9F 000EF    PUSHAB  NAME_DESC
                                                04  A9  DD 000F2    PUSHL   HOLDER_VEC
                                00000000G       00      06  FB 000F5    CALLS   #6, SYS$IDTOASC
```

```
                                68              50  D0 000FC          MOVL    R0, STATUS                                      ;
            0F                    20            6E                00  2C 000FF          MOVC5   #0, (SP), #32, #15, ATTRIB_BUFF         ; 2989
                                                          4C    AE      00104
                                                          5C    AE  B4  00106           CLRW    ATTRIB_DESC                             ; 2990
                                                    60    AE      4C  AE  9E  00109     MOVAB   ATTRIB_BUFF, ATTRIB_DESC+4              ; 2991
                                                          50      5C  AE  3C  0010E     MOVZWL  ATTRIB_DESC, R0                         ; 2996
                                                          51      4C  AE  9E  00112     MOVAB   ATTRIB_BUFF, R1
                                                          11          69  E9  00116     BLBC    ATTRIBUTES, 4$                          ; 2992
                                                6140      10    BE    0C  AE  28  00119 MOVC3   SD_RESOURCE, @SD_RESOURCE+4, (R1)[R0]    ; 2996
                                                          50      5C  AE  3C  00120     MOVZWL  ATTRIB_DESC, R0                         ; 2998
                                                          51      0C  AE  3C  00124     MOVZWL  SD_RESOURCE, R1
                                                                      0F  11  00128     BRB     5$
                                                6140      08    BE    04  AE  28  0012A 4$:   MOVC3   SD_NORESOURCE, @SD_NORESOURCE+4, (R1)[R0]  ; 3003
                                                          50      5C  AE  3C  00131     MOVZWL  ATTRIB_DESC, R0                         ; 3005
                                                          51      04  AE  3C  00135     MOVZWL  SD_NORESOURCE, R1
                                                          50          51  C0  00139 5$:   ADDL2   R1, R0                                  ;
                                              5C  AE      50          01  A1  0013C     ADDW3   #1, R0, ATTRIB_DESC                     ;
                                                          6A      F2  A7  9B  00141     MOVZBW  HOLDERLINE, FAODSC                      ; 3013
                                                    04    AA      F3  A7  9E  00145     MOVAB   HOLDERLINE+1, FAODSC+4                  ;
                                                          5C  AE      9F  00147     PUSHAB  ATTRIB_DESC                             ;
                                                          4C  AE      DD  0014D     PUSHL   NAME_DESC+4                             ;
                                                          7E  4C  AE  3C  00150     MOVZWL  NAME_DESC, -(SP)                        ;
                                                        00000000G     00  9F  00154 PUSHAB  DISDSC                                  ;
                                                7E                    22  C1  00157 ADDL3   #34, RABPTR, -(SP)                      ;
                                                          5A      DD  5A  DD  0015E     PUSHL   R10                                     ;
                                                00000000G 00          06  FB  00160     CALLS   #6, SYS$FAO                             ;
                                                          6B      DD  6B  DD  00167     PUSHL   RABPTR                                  ;
                                                00000000G 00          01  FB  00169     CALLS   #1, SYS$PUT                             ;
                                                        4200      8F  BB  00170         PUSHR   #^M<R9,SP>                              ; 3018
                                                          04      A9  9F  00174         PUSHAB  HOLDER                                  ;
                                                          04      AC  DD  00177         PUSHL   IDENT                                   ;
                                                00000000G 00          04  FB  0017A     CALLS   #4, SYS$FIND_HOLDER                     ;
                                                          68          50  D0  00181     MOVL    R0, STATUS                              ;
                                                          0E          50  E8  00184     BLBS    R0, 6$                                  ; 3019
                                                                      56  94  00187     CLRB    CONTINUE                                ; 3022
                                                000021EC  8F          50  D1  00189     CMPL    R0, #8684                               ; 3023
                                                                      03  12  00190     BNEQ    6$                                      ;
                                                          68          01  D0  00192     MOVL    #1, STATUS                              ; 3024
                                                          03          56  E9  00195 6$:   BLBC    CONTINUE, 7$                            ; 3028
                                                                    FF3C  31  00198     BRW     3$                                      ;
                                                                      5E  DD  0019B 7$:   PUSHL   SP                                      ; 3030
                                                00000000G 00          01  FB  0019D     CALLS   #1, SYS$FINISH_RDB                      ;
                                                          50          68  D0  001A4     MOVL    STATUS, R0                              ; 3032
                                                                          04  001A7     RET                                             ; 3034

; Routine Size:  424 bytes,    Routine Base:  $CODE$ + 15CB

; 3059          3035  1
```

```
;3061     3036  1 %SBTTL ' UAF$WRITE_IDENT - Display identifiers'
;3062     3037  1
;3063     3038  1 ROUTINE UAF$WRITE_IDENT ( START_IDENT, FULLFLAG, WILD_CARD ) =
;3064     3039  2 BEGIN
;3065     3040  2 !++
;3066     3041  2 !
;3067     3042  2 ! FUNCTIONAL DESCRIPTION:
;3068     3043  2 !
;3069     3044  2 !       Write specified identifier(s) to the oupput file
;3070     3045  2 !
;3071     3046  2 ! INPUTS:
;3072     3047  2 !
;3073     3048  2 !       START_IDENT     value of identifier to be displayed or
;3074     3049  2 !                       -1 if all identifiers are to be displayed.
;3075     3050  2 !       FULLFLAG        flag specifying if a full or brief listing
;3076     3051  2 !                       should be created
;3077     3052  2 !       WILD_CARD       This is a wild card display
;3078     3053  2 !
;3079     3054  2 ! IMPLICIT INPUTS:
;3080     3055  2 !
;3081     3056  2 !       RDB_HEADER_FLAG = true if header is to be output else false
;3082     3057  2 !
;3083     3058  2 ! OUTPUTS:
;3084     3059  2 !
;3085     3060  2 !       None
;3086     3061  2 !
;3087     3062  2 ! IMPLICIT OUTPUTS:
;3088     3063  2 !
;3089     3064  2 !       none
;3090     3065  2 !
;3091     3066  2 ! ROUTINE VALUE:
;3092     3067  2 !
;3093     3068  2 !       status of RDB and write operations
;3094     3069  2 !
;3095     3070  2 ! SIDE EFFECTS:
;3096     3071  2 !
;3097     3072  2 !       The output file is written to
;3098     3073  2 !
;3099     3074  2 !--
;3100     3075  2
;3101     3076  2
;3102     3077  2 $ASSUME (KGB$S_NAME, EQL, 32 ) ; ! If length changes IDENTLINE must also be fixed
;3103     3078  2
;3104     3079  2 LITERAL
;3105     3080  2     ATTRIB_BUFFLEN      = 15 ;
;3106     3081  2
;3107     3082  2 BIND
;3108     3083  2     IDENTHEAD           = CSTRING ('  !32AF !15AF !AS') ,
;3109     3084  2     IDENTLINE           = CSTRING ('  !32AF %X!XL       !AS') ,
;3110     3085  2     IDENTLINEU          = CSTRING ('  !32AF [!OW,!OW] !AS') ;
;3111     3086  2
;3112     3087  2 LOCAL
;3113     3088  2     CONTEXT             : LONG INITIAL (0) ,
;3114     3089  2     ATTRIB_DESC         : STATDESC,
;3115     3090  2     ATTRIB_BUFF         : VECTOR [ ATTRIB_BUFFLEN, BYTE ] ,
;3116     3091  2     NAME_DESC           : STATDESC,
;3117     3092  2     NAME_BUFF           : VECTOR [ KGB$S_NAME, BYTE ] ,
```

```
;  3118          3093  2     RESULT_ID           : $BBLOCK [4] ,
;  3119          3094  2     SD_ATTRIBUTES       : QUALSTR_DESC ('Attributes'),
;  3120          3095  2     SD_NAME             : QUALSTR_DESC ('Name'),
;  3121          3096  2     SD_RESOURCE         : QUALSTR_DESC ('RESOURCE'),
;  3122          3097  2     SD_NORESOURCE       : QUALSTR_DESC ('NORESOURCE') ,
;  3123          3098  2     SD_VALUE            : QUALSTR_DESC ('Value') ;
;  3124          3099  2
;  3125          3100  2 RESULT_ID = 0 ;
;  3126          3101  2 IDENT = 0 ;
;  3127          3102  2
;  3128          3103  2 NAME_DESC[POINTER] = NAME_BUFF ;
;  3129          3104  2 NAME_DESC[LENGTH]  = KGB$S_NAME ;
;  3130        P 3105  2 STATUS = $IDTOASC ( ID      = .START_IDENT,
;  3131        P 3106  2                      NAMLEN  = NAME_DESC[LENGTH] ,
;  3132        P 3107  2                      NAMBUF  = NAME_DESC ,
;  3133        P 3108  2                      RESID   = RESULT_ID ,
;  3134        P 3109  2                      ATTRIB  = ATTRIBUTES ,
;  3135          3110  2                      CONTXT  = CONTEXT ) ;
;  3136          3111  2 IF NOT .STATUS
;  3137          3112  2     THEN
;  3138          3113  3         BEGIN
;  3139          3114  3         $FINISH_RDB ( CONTXT = CONTEXT ) ;
;  3140          3115  3         RETURN .STATUS ;
;  314           3116  3         END
;  314           3117  2     ELSE IF .RDB_HEADER_FLAG
;  314           3118  2         THEN
;  3144          3119  3             BEGIN
;  3145          3120  3             !
;  3146          3121  3             ! Print a header
;  3147          3122  3             !
;  3148        P 3123  3             FAOMAC ( IDENTHEAD,
;  3149        P 3124  3                      .SD_NAME[LENGTH], .SD_NAME[POINTER],
;  3150        P 3125  3                      .SD_VALUE[LENGTH], .SD_VALUE[POINTER],
;  3151          3126  3                      SD_ATTRIBUTES ) ;
;  3152          3127  3             RDB_HEADER_FLAG = FALSE ;
;  3153          3128  2             END ;
;  3154          3129  2
;  3155          3130  2
;  3156          3131  2 !
;  3157          3132  2 ! Loop until we find all the identifiers
;  3158          3133  2 !
;  3159          3134  2 DO
;  3160          3135  3     BEGIN
;  3161          3136  3
;  3162          3137  3     !
;  3163          3138  3     ! Build the attribute string
;  3164          3139  3     !
;  3165          3140  3     CH$FILL ( %C' ', ATTRIB_BUFFLEN, ATTRIB_BUFF ) ;
;  3166          3141  3     ATTRIB_DESC[LENGTH]  = 0 ;
;  3167          3142  3     ATTRIB_DESC[POINTER] = ATTRIB_BUFF ;
;  3168          3143  4     IF ( .ATTRIBUTES AND KGB$M_RESOURCE )
;  3169          3144  3         THEN
;  3170          3145  4             BEGIN
;  3171          3146  4             CH$MOVE ( .SD_RESOURCE[LENGTH], .SD_RESOURCE[POINTER],
;  3172          3147  4                       ATTRIB_BUFF[.ATTRIB_DESC[LENGTH]] ) ;
;  3173          3148  4             ATTRIB_DESC[LENGTH] =
;  3174          3149  4               .ATTRIB_DESC[LENGTH] + .SD_RESOURCE[LENGTH] + 1 ;
```

```
;  3175          3150  4              END
;  3176          3151  3          ELSE
;  3177          3152  4              BEGIN
;  3178          3153  4              CH$MOVE ( .SD_NORESOURCE[LENGTH], .SD_NORESOURCE[POINTER],
;  3179          3154  4                        ATTRIB_BUFF[.ATTRIB_DESC[LENGTH]] ) ;
;  3180          3155  4              ATTRIB_DESC[LENGTH] =
;  3181          3156  4                .ATTRIB_DESC[LENGTH] + .SD_NORESOURCE[LENGTH] + 1 ;
;  3182          3157  3              END ;
;  3183          3158  3
;  3184          3159  3      !
;  3185          3160  3      ! Output the line
;  3186          3161  3      !
;  3187          3162  3      IF .RESULT_ID[UIC$V_FORMAT] EQL UIC$K_UIC_FORMAT
;  3188          3163  3          THEN
;  3189        P 3164  3              FAOMAC ( IDENTLINEU,
;  3190        P 3165  3                       .NAME_DESC[LENGTH],
;  3191        P 3166  3                       .NAME_DESC[POINTER],        ! Identifier name
;  3192        P 3167  3                       .RESULT_ID[UIC$V_GROUP],    ! Group number
;  3193        P 3168  3                       .RESULT_ID[UIC$V_MEMBER],   ! member number
;  3194          3169  4                       ATTRIB_DESC )               ! Attributes
;  3195          3170  3          ELSE
;  3196        P 3171  3              FAOMAC ( IDENTLINE,
;  3197        P 3172  3                       .NAME_DESC[LENGTH],
;  3198        P 3173  3                       .NAME_DESC[POINTER],        ! Identifier name
;  3199        P 3174  3                       .RESULT_ID,                 ! Identifier value
;  3200          3175  3                       ATTRIB_DESC ) ;             ! Attributes
;  3201          3176  3      !
;  3202          3177  3      ! If this is a show /FULL then list this guys holders
;  3203          3178  3      !
;  3204          3179  3      IF .FULLFLAG
;  3205          3180  3          THEN
;  3206          3181  4              BEGIN
;  3207          3182  4              STATUS = UAF$WRITE_HOLDERS ( .RESULT_ID ) ;
;  3208          3183  4              IF NOT .STATUS
;  3209          3184  4                  THEN
;  3210          3185  5                      BEGIN
;  3211          3186  5                      $FINISH_RDB ( CONTXT = CONTEXT ) ;
;  3212          3187  5                      RETURN .STATUS ;
;  3213          3188  4                      END ;
;  3214          3189  3              END ;
;  3215          3190  3
;  3216          3191  3
;  3217          3192  3      NAME_DESC[LENGTH]  = KGB$S_NAME ;
;  3218        P 3193  3      STATUS = $IDTOASC ( ID      = .START_IDENT,
;  3219        P 3194  3                          NAMLEN  = NAME_DESC[LENGTH] ,
;  3220        P 3195  3                          NAMBUF  = NAME_DESC ,
;  3221        P 3196  3                          RESID   = RESULT_ID ,
;  3222        P 3197  3                          ATTRIB  = ATTRIBUTES ,
;  3223          3198  3                          CONTXT  = CONTEXT ) ;
;  3224          3199  3      IF (NOT .STATUS) AND
;  3225          3200  4         (.STATUS NEQ SS$_NOSUCHID )
;  3226          3201  3          THEN
;  3227          3202  4              BEGIN
;  3228          3203  4              $FINISH_RDB ( CONTXT = CONTEXT ) ;
;  3229          3204  4              RETURN .STATUS ;
;  3230          3205  3              END ;
;  3231          3206  3      END
```

```
;  3232          3207  3 !
;  3233          3208  3 ! Only continue looping if we are dealing with a wild card
;  3234          3209  3 !
;  3235          3210  2 WHILE ( .STATUS ) AND ( .WILD_CARD ) ;
;  3236          3211  2
;  3237          3212  2 $FINISH_RDB ( CONTXT = CONTEXT ) ;
;  3238          3213  2
;  3239          3214  2 RETURN TRUE ;
;  3240          3215  2
;  3241          3216  1 END ;                                   ! End of UAF$WRITE_IDENT


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                    11  000BD P.AAW:  .BYTE   17
21 20 46 41 35 31 21 20 46 41 32 33 21 20           20  000BE        .ASCII  \  !32AF !15AF !AS\
                                                    53  000CD
                                                    16  000CF P.AAX:  .BYTE   22
20 20 4C 58 21 58 25 20 46 41 32 33 21 20           20  000D0        .ASCII  \  !32AF %X!XL      !AS\
                        53 41 21 20 20 20           20  000DF
                                                    15  000E6 P.AAY:  .BYTE   21
4F 21 2C 57 4F 21 5B 20 46 41 32 33 21 20           20  000E7        .ASCII  \  !32AF [!OW,!OW] !AS\
                        53 41 21 20 5D 57           F6  000F6
                  73 65 74 75 62 69 72 74 74 41     FC  000FC P.AAZ:  .ASCII  \Attributes\
                                    65 6D 61 4E     06  00106 P.ABA:  .ASCII  \Name\
                     45 43 52 55 4F 53 45 52     0A10  0010A P.ABB:  .ASCII  \RESOURCE\
                  45 43 52 55 4F 53 45 52 4F 4E     12  00112 P.ABC:  .ASCII  \NORESOURCE\
                                    65 75 6C 61 56  1C  0011C P.ABD:  .ASCII  \Value\

                                                    IDENTHEAD=          P.AAW
                                                    IDENTLINE=          P.AAX
                                                    IDENTLINEU=         P.AAY


                                                        .PSECT  $CODE$,NOWRT,2

                                                OFFC 00000 UAF$WRITE_IDENT:
                                                        .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11    ; 3038
                5B 00000000G   00  9E 00002             MOVAB   DISDSC, R11                             ;
                5A 00000000G   00  9E 00009             MOVAB   RABPTR, R10                             ;
                59 00000000'   00  9E 00010             MOVAB   ATTRIBUTES, R9                          ;
                58 00000000'   00  9E 00017             MOVAB   STATUS, R8                              ;
                57 00000000G   00  9E 0001E             MOVAB   FAODSC, R7                              ;
                56 00000000'   00  9E 00025             MOVAB   P.AAZ, R6                               ;
                5E         98  AE  9E 0002C             MOVAB   -104(SP), SP                            ;
                               7E  D4 00030             CLRL    CONTEXT                                 ; 3039
            64  AE 010E0000    8F  D0 00032             MOVL    #17694720, ATTRIB_DESC                  ; 3089
                          68   AE  D4 0003A             CLRL    ATTRIB_DESC+4                           ;
            4C  AE 010E0000    8F  D0 0003D             MOVL    #17694720, NAME_DESC                    ; 3091
                          50   AE  D4 00045             CLRL    NAME_DESC+4                             ;
            24  AE 010E000A    8F  D0 00048             MOVL    #17694730, SD_ATTRIBUTES                ; 3094
            28  AE                 66  9E 00050         MOVAB   P.AAZ, SD_ATTRIBUTES+4                  ;
            1C  AE 010E0004    8F  D0 00054             MOVL    #17694724, SD_NAME                      ; 3095
            20  AE         0A  A6  9E 0005C             MOVAB   P.ABA, SD_NAME+4                        ;
            14  AE 010E0008    8F  D0 00061             MOVL    #17694728, SD_RESOURCE                  ; 3096
            18  AE         0E  A6  9E 00069             MOVAB   P.ABB, SD_RESOURCE+4                    ;
```

```
                    0C   AE 010E000A    8F   D0 0006E                MOVL    #17694730, SD_NORESOURCE                ; 3097
                    10   AE       16    A6   9E 00076                MOVAB   P.ABC, SD_NORESOURCE+4                  ;
                    04   AE 010E0005    8F   D0 0007B                MOVL    #17694725, SD_VALUE                     ; 3098
                    08   AE       20    A6   9E 00083                MOVAB   P.ABD, SD_VALUE+4                       ;
                                        7E   D4 00088                CLRL    RESULT_ID                               ; 3100
                                    0C  A9   D4 0008A                CLRL    IDENT                                   ; 3101
                    54   AE       30    AE   9E 0008D                MOVAB   NAME_BUFF, NAME_DESC+4                  ; 3103
                    50   AE           20   B0 00092                MOVW    #32, NAME_DESC                          ; 3104
                                    04   AE   9F 00096                PUSHAB  CONTEXT                                 ; 3110
                                        59   DD 00099                PUSHL   R9                                      ;
                                    08   AE   9F 0009B                PUSHAB  RESULT_ID                               ;
                                    5C   AE   9F 0009E                PUSHAB  NAME_DESC                               ;
                                    60   AE   9F 000A1                PUSHAB  NAME_DESC                               ;
                                    04   AC   DD 000A4                PUSHL   START_IDENT                             ;
              00000000G             00        FB 000A7                CALLS   #6, SYS$IDTOASC                         ;
                                    68        D0 000AE                MOVL    R0, STATUS                              ;
                                    03        E8 000B1                BLBS    STATUS, 1$                              ; 3111
                                          011A 31 000B4                BRW     8$                                      ;
                                    35   FC   A9 E9 000B7 1$:         BLBC    RDB_HEADER_FLAG, 2$                     ; 3117
                                    67   C1   A6 9B 000BB            MOVZBW  IDENTHEAD, FAODSC                       ; 3126
                    04   A7       C2   A6   9E 000BF                MOVAB   IDENTHEAD+1, FAODSC+4                   ;
                                    28   AE   9F 000C4                PUSHAB  SD_ATTRIBUTES                           ;
                                    10   AE   DD 000C7                PUSHL   SD_VALUE+4                              ;
                                    7E   10   AE 3C 000CA            MOVZWL  SD_VALUE, -(SP)                         ;
                                    30   AE   DD 000CE                PUSHL   SD_NAME+4                               ;
                                    7E   30   AE 3C 000D1            MOVZWL  SD_NAME, -(SP)                          ;
                                        5B   DD 000D5                PUSHL   R11                                     ;
                7E                  6A        22 C1 000D7            ADDL3   #34, RABPTR, -(SP)                      ;
                                        57   DD 000DB                PUSHL   R7                                      ;
                   00000000G        00        08 FB 000DD            CALLS   #8, SYS$FAO                             ;
                                        6A   DD 000E4                PUSHL   RABPTR                                  ;
                   00000000G        00        01 FB 000E6            CALLS   #1, SYS$PUT                             ;
                                FC  A9   94 000ED                    CLRB    RDB_HEADER_FLAG                         ; 3127
                0F          20           6E   00 2C 000F0 2$:         MOVC5   #0, (SP), #32, #15, ATTRIB_BUFF         ; 3140
                                    58   AE       000F5
                                    68   AE   B4 000F7                CLRW    ATTRIB_DESC                             ; 3141
                                6C  AE   58   AE 9E 000FA            MOVAB   ATTRIB_BUFF, ATTRIB_DESC+4              ; 3142
                                    50   68   AE 3C 000FF            MOVZWL  ATTRIB_DESC, R0                         ; 3147
                                    51   58   AE 9E 00103            MOVAB   ATTRIB_BUFF, R1                         ;
                                    11        69 E9 00107            BLBC    ATTRIBUTES, 3$                          ; 3143
                        6140        1C   BE   18 AE 28 0010A        MOVC3   SD_RESOURCE, @SD_RESOURCE+4, (R1)[R0]   ; 3147
                                    50   68   AE 3C 00111            MOVZWL  ATTRIB_DESC, R0                         ; 3149
                                    51   18   AE 3C 00115            MOVZWL  SD_RESOURCE, R1                         ;
                                         0F   11 00119               BRB     4$                                      ;
                        6140        14   BE   10 AE 28 0011B 3$:     MOVC3   SD_NORESOURCE, @SD_NORESOURCE+4, (R1)[R0] ; 3154
                                    50   68   AE 3C 00122            MOVZWL  ATTRIB_DESC, R0                         ; 3156
                                    51   10   AE 3C 00126            MOVZWL  SD_NORESOURCE, R1                       ;
                                    50        51 C0 0012A 4$:         ADDL2   R1, R0                                  ;
                            68  AE   50        01 A1 0012D            ADDW3   #1, R0, ATTRIB_DESC                     ;
                                50   6A        22 C1 00132            ADDL3   #34, RABPTR, R0                         ; 3169
                                C0   8F   03  AE 93 00136            BITB    RESULT_ID+3, #192                       ; 3162
                                         2C   12 00139               BNEQ    5$                                      ;
                                    67   EA   A6 9B 0013D            MOVZBW  IDENTLINEU, FAODSC                      ; 3169
                                04  A7   EB   A6 9E 00141            MOVAB   IDENTLINEU+1, FAODSC+4                  ;
                                    68   AE   9F 00146               PUSHAB  ATTRIB_DESC                             ;
                                    7E   04   AE 3C 00149            MOVZWL  RESULT_ID, -(SP)                        ;
                7E      0A  AE          0E        00 EF 0014D        EXTZV   #0, #14, RESULT_ID+2, -(SP)             ;
```

```
                                60    AE  DD 00153          PUSHL   NAME_DESC+4
                             7E 60    AE  3C 00156          MOVZWL  NAME_DESC, -(SP)
                           0801 8F    BB  0015A          PUSHR   #^M<R0,R11>
                                57    DD  0015E          PUSHL   R7
                 00000000G   00       08  FB 00160          CALLS   #8, SYS$FAO
                                23    11  00167          BRB     6$
                             67 D3    A6  9B 00169 5$:       MOVZBW  IDENTLINE, FAODSC                       ;3175
                          04 A7 D4    A6  9E 0016D           MOVAB   IDENTLINE+1, FAODSC+4
                                68    AE  9F 00172           PUSHAB  ATTRIB_DESC
                                04    AE  DD 00175           PUSHL   RESULT_ID
                                5C    AE  DD 00178           PUSHL   NAME_DESC+4
                             7E 5C    AE  3C 0017B           MOVZWL  NAME_DESC, -(SP)
                           0801 8F    BB  0017F          PUSHR   #^M<R0,R11>
                                57    DD  00183          PUSHL   R7
                 00000000G   00       07  FB 00185          CALLS   #7, SYS$FAO
                                6A    DD  0018C 6$:      PUSHL   RABPTR
                 00000000G   00       01  FB 0018E          CALLS   #1, SYS$PUT
                             0D 08    AC  E9 00195           BLBC    FULLFLAG, 7$                            ;3179
                                6E    DD  00199          PUSHL   RESULT_ID                               ;3182
                      FCB8   CF       01  FB 0019B          CALLS   #1, UAF$WRITE_HOLDERS
                             68       50  D0 001A0          MOVL    R0, STATUS
                             2B       68  E9 001A3          BLBC    STATUS, 8$                              ;3183
                          50 AE       20  B0 001A6 7$:       MOVW    #32, NAME_DESC                          ;3192
                                04    AE  9F 001AA           PUSHAB  CONTEXT                                 ;3198
                                59    DD  001AD          PUSHL   R9
                                08    AE  9F 001AF           PUSHAB  RESULT_ID
                                5C    AE  9F 001B2           PUSHAB  NAME_DESC
                                60    AE  9F 001B5           PUSHAB  NAME_DESC
                                04    AC  DD 001B8          PUSHL   START_IDENT
                 00000000G   00       06  FB 001BB          CALLS   #6, SYS$IDTOASC
                             68       50  D0 001C2          MOVL    R0, STATUS
                             17       50  E8 001C5          BLBS    R0, 9$                                  ;3199
                 000021EC    8F       50  D1 001C8          CMPL    R0, #8684                               ;3200
                                0E    13  001CF          BEQL    9$
                                04    AE  9F 001D1 8$:       PUSHAB  CONTEXT                                 ;3203
                 00000000G   00       01  FB 001D4          CALLS   #1, SYS$FINISH_RDB
                             50       68  D0 001DB          MOVL    STATUS, R0                              ;3204
                                      04  001DE          RET
                             07       68  E9 001DF 9$:       BLBC    STATUS, 10$                             ;3210
                             03    0C AC  E9 001E2           BLBC    WILD_CARD, 10$
                                FF07  31  001E6          BRW     2$
                                04    AE  9F 001E9 10$:      PUSHAB  CONTEXT                                 ;3212
                 00000000G   00       01  FB 001EC          CALLS   #1, SYS$FINISH_RDB
                             50       01  D0 001F3          MOVL    #1, R0                                  ;3214
                                      04  001F6          RET                                             ;3216

; Routine Size:  503 bytes,    Routine Base:  $CODE$ + 1773

; 3242          3217  1
```

```
;  3244     3218  1 %SBTTL ' UAF$WRITE_RIGHTS - Display rights held by user'
;  3245     3219  1
;  3246     3220  1 GLOBAL ROUTINE UAF$WRITE_RIGHTS ( HOLDER_ADDR ) =
;  3247     3221  2 BEGIN
;  3248     3222  2 !++
;  3249     3223  2 !
;  3250     3224  2 ! FUNCTIONAL DESCRIPTION:
;  3251     3225  2 !
;  3252     3226  2 !       Display all identifiers held by the specified identifier
;  3253     3227  2 !
;  3254     3228  2 ! INPUTS:
;  3255     3229  2 !
;  3256     3230  2 !       HOLDER  address of a quadword holder constructed from the id
;  3257     3231  2 !               value of the identifier.
;  3258     3232  2 !
;  3259     3233  2 ! IMPLICIT INPUTS:
;  3260     3234  2 !
;  3261     3235  2 !       RDB_HEADER_FLAG - true if a header is to be output, else false.
;  3262     3236  2 !
;  3263     3237  2 ! OUTPUTS:
;  3264     3238  2 !
;  3265     3239  2 !       None
;  3266     3240  2 !
;  3267     3241  2 ! IMPLICIT OUTPUTS:
;  3268     3242  2 !
;  3269     3243  2 !       none
;  3270     3244  2 !
;  3271     3245  2 ! ROUTINE VALUE:
;  3272     3246  2 !
;  3273     3247  2 !       none
;  3274     3248  2 !
;  3275     3249  2 ! SIDE EFFECTS:
;  3276     3250  2 !
;  3277     3251  2 !       The output file is written to
;  3278     3252  2 !
;  3279     3253  2 !--
;  3280     3254  2
;  3281     3255  2
;  3282     3256  2 $ASSUME (KGB$S_NAME, EQL, 32 ) ; ! If length changes RIGHTSLINE must also be fixed
;  3283     3257  2
;  3284     3258  2 LITERAL
;  3285     3259  2     ATTRIB_BUFFLEN      = 15 ;
;  3286     3260  2
;  3287     3261  2 BIND
;  3288     3262  2     RIGHTSHEAD1         = CSTRING ('!34AF !15AF !AS') ,
;  3289     3263  2     RIGHTSHEAD2         = CSTRING ('Identifiers held by !AS : ') ,
;  3290     3264  2     RIGHTSLINE          = CSTRING ('  !32AF %X!XL      !AS') ,
;  3291     3265  2     RIGHTSLINEU         = CSTRING ('  !32AF [!OW,!OW] !AS') ;
;  3292     3266  2
;  3293     3267  2 LOCAL
;  3294     3268  2     CONTINUE            : BYTE ,
;  3295     3269  2     CONTEXT             : LONG INITIAL (0) ,
;  3296     3270  2     ATTRIB_DESC         : STATDESC,
;  3297     3271  2     ATTRIB_BUFF         : VECTOR [ ATTRIB_BUFFLEN, BYTE ] ,
;  3298     3272  2     NAME_DESC           : STATDESC,
;  3299     3273  2     NAME_BUFF           : VECTOR [ KGB$S_NAME, BYTE ] ,
;  3300     3274  2     SD_ATTRIBUTES       : QUALSTR_DESC ('Attributes'),
```

```
;  3301          3275  2     SD_NAME             : QUALSTR_DESC ('Identifier'),
;  3302          3276  2     SD_RESOURCE         : QUALSTR_DESC ('RESOURCE'),
;  3303          3277  2     SD_NORESOURCE       : QUALSTR_DESC ('NORESOURCE') ,
;  3304          3278  2     SD_VALUE            : QUALSTR_DESC ('Value') ;
;  3305          3279  2
;  3306        P 3280  2 STATUS = $FIND_HELD ( HOLDER  = .HOLDER_ADDR,
;  3307        P 3281  2                       ID      = IDENT,
;  3308        P 3282  2                       ATTRIB  = ATTRIBUTES,
;  3309          3283  2                       CONTXT  = CONTEXT ) ;
;  3310          3284  2 IF NOT .STATUS
;  3311          3285  2     THEN
;  3312          3286  3         BEGIN
;  3313          3287  3         $FINISH_RDB ( CONTXT = CONTEXT ) ;
;  3314          3288  3         IF .STATUS EQL SS$_NOSUCHID
;  3315          3289  3             THEN STATUS = SS$_NORMAL ;
;  3316          3290  3         RETURN .STATUS ;
;  3317          3291  3         END
;  3318          3292  3
;  3319          3293  2     ELSE
;  3320          3294  3         BEGIN
;  3321          3295  3         !
;  3322          3296  3         ! Init the name descriptor
;  3323          3297  3         ! and find the ASCII name of the holder
;  3324          3298  3         !
;  3325          3299  3         NAME_DESC[POINTER] = NAME_BUFF ;
;  3326          3300  3         NAME_DESC[LENGTH]  = KGB$S_NAME ;
;  3327          3301  3         CH$FILL ( %C' ', .NAME_DESC[LENGTH], .NAME_DESC[POINTER] ) ;
;  3328        P 3302  3         STATUS = $IDTOASC ( ID     = ..HOLDER_ADDR,
;  3329        P 3303  3                             NAMLEN = NAME_DESC[LENGTH],
;  3330          3304  3                             NAMBUF = NAME_DESC ) ;
;  3331          3305  3         !
;  3332          3306  3         ! Print a header
;  3333          3307  3         !
;  3334          3308  3         IF .RDB_HEADER_FLAG
;  3335          3309  3             THEN
;  3336          3310  4                 BEGIN
;  3337        P 3311  4                 FAOMAC ( RIGHTSHEAD1,
;  3338        P 3312  4                          .SD_NAME[LENGTH], .SD_NAME[POINTER],
;  3339        P 3313  4                          .SD_VALUE[LENGTH], .SD_VALUE[POINTER],
;  3340          3314  4                          SD_ATTRIBUTES ) ;
;  3341          3315  4                 RDB_HEADER_FLAG = FALSE ;
;  3342          3316  3                 END ;
;  3343          3317  3         IF .STR_WILD OR .MEM_WILD OR .GRP_WILD THEN
;  3344          3318  3             FAOMAC ( RIGHTSHEAD2, NAME_DESC ) ;
;  3345          3319  2         END ;
;  3346          3320  2
;  3347          3321  2
;  3348          3322  2 !
;  3349          3323  2 ! Loop until we find all the identifiers held by this guy
;  3350          3324  2 !
;  3351          3325  2 CONTINUE = TRUE ;
;  3352          3326  2 DO
;  3353          3327  3     BEGIN
;  3354          3328  3     !
;  3355          3329  3     ! Init the name descriptor
;  3356          3330  3     ! and find the ASCII name of the identifier
;  3357          3331  3     !
```

```
;  3358          3332  3      NAME_DESC[POINTER] = NAME_BUFF ;
;  3359          3333  3      NAME_DESC[LENGTH]  = KGB$S_NAME ;
;  3360          3334  3      CH$FILL ( %C' ', .NAME_DESC[LENGTH], .NAME_DESC[POINTER] ) ;
;  3361        P 3335  3      STATUS = $IDTOASC ( ID     = .IDENT,
;  3362        P 3336  3                          NAMLEN = NAME_DESC[LENGTH],
;  3363          3337  3                          NAMBUF = NAME_DESC ) ;
;  3364          3338  3
;  3365          3339  3      !
;  3366          3340  3      ! Build the attribute string
;  3367          3341  3      !
;  3368          3342  3      CH$FILL ( %C' ', ATTRIB_BUFFLEN, ATTRIB_BUFF ) ;
;  3369          3343  3      ATTRIB_DESC[LENGTH]  = 0 ;
;  3370          3344  3      ATTRIB_DESC[POINTER] = ATTRIB_BUFF ;
;  3371          3345  4      IF ( .ATTRIBUTES AND KGB$M_RESOURCE )
;  3372          3346  3          THEN
;  3373          3347  4              BEGIN
;  3374          3348  4              CH$MOVE ( .SD_RESOURCE[LENGTH], .SD_RESOURCE[POINTER],
;  3375          3349  4                        ATTRIB_BUFF[.ATTRIB_DESC[LENGTH]] ) ;
;  3376          3350  4              ATTRIB_DESC[LENGTH] =
;  3377          3351  4                .ATTRIB_DESC[LENGTH] + .SD_RESOURCE[LENGTH] + 1 ;
;  3378          3352  4              END
;  3379          3353  3          ELSE
;  3380          3354  4              BEGIN
;  3381          3355  4              CH$MOVE ( .SD_NORESOURCE[LENGTH], .SD_NORESOURCE[POINTER],
;  3382          3356  4                        ATTRIB_BUFF[.ATTRIB_DESC[LENGTH]] ) ;
;  3383          3357  4              ATTRIB_DESC[LENGTH] =
;  3384          3358  4                .ATTRIB_DESC[LENGTH] + .SD_NORESOURCE[LENGTH] + 1 ;
;  3385          3359  3              END ;
;  3386          3360  3
;  3387          3361  3      !
;  3388          3362  3      ! Output the line
;  3389          3363  3      !
;  3390          3364  3      IF .IDENT[UIC$V_FORMAT] EQL UIC$K_UIC_FORMAT
;  3391          3365  3          THEN
;  3392        P 3366  3              FAOMAC ( RIGHTSLINE,
;  3393        P 3367  3                       .NAME_DESC[LENGTH],
;  3394        P 3368  3                       .NAME_DESC[POINTER],       ! Identifier name
;  3395        P 3369  3                       .IDENT[UIC$V_GROUP],       ! Group number
;  3396        P 3370  3                       .IDENT[UIC$V_MEMBER],      ! Member number
;  3397          3371  4                       ATTRIB_DESC )              ! Attributes
;  3398          3372  3          ELSE
;  3399        P 3373  3              FAOMAC ( RIGHTSLINE,
;  3400        P 3374  3                       .NAME_DESC[LENGTH],
;  3401        P 3375  3                       .NAME_DESC[POINTER],       ! Identifier name
;  3402        P 3376  3                       .IDENT,                    ! Identifier value
;  3403          3377  3                       ATTRIB_DESC ) ;            ! Attributes
;  3404          3378  3
;  3405        P 3379  3      STATUS = $FIND_HELD ( HOLDER  = .HOLDER_ADDR,
;  3406        P 3380  3                            ID      = IDENT,
;  3407        P 3381  3                            ATTRIB  = ATTRIBUTES,
;  3408          3382  3                            CONTXT  = CONTEXT ) ;
;  3409          3383  3      IF NOT .STATUS
;  3410          3384  3          THEN
;  3411          3385  4              BEGIN
;  3412          3386  4              CONTINUE = FALSE ;
;  3413          3387  4              IF .STATUS EQL SS$_NOSUCHID
;  3414          3388  4                  THEN STATUS = SS$_NORMAL ;
```

```
;  3415          3389  3              END ;
;  3416          3390  3          END
;  3417          3391  2 WHILE .CONTINUE ;
;  3418          3392  2
;  3419          3393  2 $FINISH_RDB ( CONTXT = CONTEXT ) ;
;  3420          3394  2
;  3421          3395  2 RETURN .STATUS ;
;  3422          3396  2
;  3423          3397  1 END ;                                   ! End of UAF$WRITE_RIGHTS


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                        0F  00121 P.ABE:  .BYTE   15                                    ;
53 41 21 20 46 41 35 31 21 20 46 41 34 33               21  00122         .ASCII  \!34AF !15AF !AS\                     ;
                                                        1A  00131 P.ABF:  .BYTE   26                                    ;
6C 65 68 20 73 72 65 69 66 69 74 6E 65 64               49  00132         .ASCII  \Identifiers held by !AS : \          ;
                  20 3A 20 53 41 21 20 79 62 20         64  00141                                                       ;
                                                        16  0014C P.ABG:  .BYTE   22                                    ;
20 20 4C 58 21 58 25 20 46 41 32 33 21 20               20  0014D         .ASCII  \  !32AF %X!XL       !AS\             ;
                        53 41 21 20 20 20               20  0015C                                                       ;
                                                        15  00163 P.ABH:  .BYTE   21                                    ;
4F 21 2C 57 4F 21 5B 20 46 41 32 33 21 20               20  00164         .ASCII  \  !32AF [!OW,!OW] !AS\               ;
                        53 41 21 20 5D                  57  00173                                                       ;
                  73 65 74 75 62 69 72 74 74            41  00179 P.ABI:  .ASCII  \Attributes\                          ;
                  72 65 69 66 69 74 6E 65 64            49  00183 P.ABJ:  .ASCII  \Identifier\                          ;
                     45 43 52 55 4F 53 45               52  0018D P.ABK:  .ASCII  \RESOURCE\                            ;
                  45 43 52 55 4F 53 45 52 4F            4E  00195 P.ABL:  .ASCII  \NORESOURCE\                          ;
                                    65 75 6C 61         56  0019F P.ABM:  .ASCII  \Value\                               ;

                                                                RIGHTSHEAD1=        P.ABE
                                                                RIGHTSHEAD2=        P.ABF
                                                                RIGHTSLINE=         P.ABG
                                                                RIGHTSLINEU=        P.ABH
                                                                .EXTRN  SYS$FIND_HELD

                                                                .PSECT  $CODE$,NOWRT,2

                                                        0FFC 00000      .ENTRY  UAF$WRITE_RIGHTS, Save R2,R3,R4,R5,R6,R7,-  ; 3220
                                                                                R8,R9,R10,R11
      5B 00000000G   00  9E 00002      MOVAB   RABPTR, R11                                                             ;
      5A 00000000'   00  9E 00009      MOVAB   STATUS, R10                                                             ;
      59 00000000G   00  9E 00010      MOVAB   FAODSC, R9                                                              ;
      58 00000000'   00  9E 00017      MOVAB   RIGHTSLINE+1, R8                                                        ;
      57 00000000'   00  9E 0001E      MOVAB   IDENT, R7                                                               ;
      5E         98  AE  9E 00025      MOVAB   -104(SP), SP                                                            ;
                     7E  D4 00029      CLRL    CONTEXT                                                                 ; 3221
64    AE 010E0000  8F  D0 0002B      MOVL    #17694720, ATTRIB_DESC                                                  ; 3270
              68   AE  D4 00033      CLRL    ATTRIB_DESC+4                                                           ;
4C    AE 010E0000  8F  D0 00036      MOVL    #17694720, NAME_DESC                                                    ; 3272
              50   AE  D4 0003E      CLRL    NAME_DESC+4                                                             ;
24    AE 010E000A  8F  D0 00041      MOVL    #17694730, SD_ATTRIBUTES                                                ; 3274
28    AE     2C    A8  9E 00049      MOVAB   P.ABI, SD_ATTRIBUTES+4                                                  ;
1C    AE 010E000A  8F  D0 0004E      MOVL    #17694730, SD_NAME                                                      ; 3275
20    AE     36    A8  9E 00056      MOVAB   P.ABJ, SD_NAME+4                                                        ;
14    AE 010E0008  8F  D0 0005B      MOVL    #17694728, SD_RESOURCE                                                  ; 3276
```

```
                                18   AE        40   A8  9E 00063          MOVAB   P.ABK, SD_RESOURCE+4                          ;
                                0C   AE  010E000A   8F  D0 00068          MOVL    #17694730, SD_NORESOURCE                     ; 3277
                                10   AE        48   A8  9E 00070          MOVAB   P.ABL, SD_NORESOURCE+4                       ;
                                04   AE  010E0005   8F  D0 00075          MOVL    #17694725, SD_VALUE                          ; 3278
                                08   AE        52   A8  9E 0007D          MOVAB   P.ABM, SD_VALUE+4                            ;
                                                    5E  DD 00082          PUSHL   SP                                           ; 3283
                                              F4    A7  9F 00084          PUSHAB  ATTRIBUTES                                   ;
                                                    57  DD 00087          PUSHL   R7                                           ;
                                              04    AC  DD 00089          PUSHL   HOLDER_ADDR                                  ;
                          00000000G   00            04  FB 0008C          CALLS   #4, SYS$FIND_HELD                            ;
                                      6A            50  D0 00093          MOVL    R0, STATUS                                   ;
                                      18            6A  E8 00096          BLBS    STATUS, 2$                                   ; 3284
                                                    5E  DD 00099          PUSHL   SP                                           ; 3287
                          00000000G   00            01  FB 0009B          CALLS   #1, SYS$FINISH_RDB                           ;
                          000021EC    8F            6A  D1 000A2          CMPL    STATUS, #8684                                ; 3288
                                                    03  12 000A9          BNEQ    1$                                           ;
                                      6A            01  D0 000AB          MOVL    #1, STATUS                                   ; 3289
                                                01A9  31 000AE 1$:        BRW     13$                                          ; 3290
                                50   AE        2C   AE  9E 000B1 2$:     MOVAB   NAME_BUFF, NAME_DESC+4                       ; 3299
                                4C   AE              20  B0 000B6          MOVW    #32, NAME_DESC                               ; 3300
    4C   AE              20                    6E        00  2C 000BA          MOVC5   #0, (SP), #32, NAME_DESC, @NAME_DESC+4       ; 3301
                                              50    BE      000C0
                                                    7E  7C 000C2          CLRQ    -(SP)                                        ; 3304
                                                    7E  D4 000C4          CLRL    -(SP)                                        ;
                                              58    AE  9F 000C6          PUSHAB  NAME_DESC                                    ;
                                              5C    AE  9F 000C9          PUSHAB  NAME_DESC                                    ;
                                              04    BC  DD 000CC          PUSHL   @HOLDER_ADDR                                 ;
                          00000000G   00            06  FB 000CF          CALLS   #6, SYS$IDTOASC                              ;
                                      6A            50  D0 000D6          MOVL    R0, STATUS                                   ;
                                      39            F0  A7  E9 000D9          BLBC    RDB_HEADER_FLAG, 3$                      ; 3308
                                      69            D4  A8  9B 000DD          MOVZBW  RIGHTSHEAD1, FAODSC                      ; 3314
                                04    A9            D5  A8  9E 000E1          MOVAB   RIGHTSHEAD1+1, FAODSC+4                  ;
                                                    24  AE  9F 000E6          PUSHAB  SD_ATTRIBUTES                            ;
                                                    0C  AE  DD 000E9          PUSHL   SD_VALUE+4                               ;
                                      7E            0C  AE  3C 000EC          MOVZWL  SD_VALUE, -(SP)                          ;
                                                    2C  AE  DD 000F0          PUSHL   SD_NAME+4                                ;
                                      7E            2C  AE  3C 000F3          MOVZWL  SD_NAME, -(SP)                           ;
                                          00000000G   00  9F 000F7          PUSHAB  DISDSC                                   ;
    7E                    6B                        22  C1 000FD          ADDL3   #34, RABPTR, -(SP)                       ;
                                                    59  DD 00101          PUSHL   R9                                       ;
        00000000G   00                              08  FB 00103          CALLS   #8, SYS$FAO                              ;
                                                    6B  DD 0010A          PUSHL   RABPTR                                   ;
        00000000G   00                              01  FB 0010C          CALLS   #1, SYS$PUT                              ;
                                              F0    A7  94 00113          CLRB    RDB_HEADER_FLAG                          ; 3315
                                      0E  00000000G   00  E8 00116 3$:     BLBS    STR_WILD, 4$                             ; 3317
                                      07  00000000G   00  E8 0011D          BLBS    MEM_WILD, 4$                             ;
                                      28  00000000G   00  E9 00124          BLBC    GRP_WILD, 5$                             ;
                                      69            E4  A8  9B 0012B 4$:     MOVZBW  RIGHTSHEAD2, FAODSC                      ; 3318
                                04    A9            E5  A8  9E 0012F          MOVAB   RIGHTSHEAD2+1, FAODSC+4                  ;
                                                    4C  AE  9F 00134          PUSHAB  NAME_DESC                                ;
                                          00000000G   00  9F 00137          PUSHAB  DISDSC                                   ;
    7E                    6B                        22  C1 0013D          ADDL3   #34, RABPTR, -(SP)                       ;
                                                    59  DD 00141          PUSHL   R9                                       ;
        00000000G   00                              04  FB 00143          CALLS   #4, SYS$FAO                              ;
                                                    6B  DD 0014A          PUSHL   RABPTR                                   ;
        00000000G   00                              01  FB 0014C          CALLS   #1, SYS$PUT                              ;
                                      56            01  90 00153 5$:     MOVB    #1, CONTINUE                             ; 3325
```

```
                          50    AE       2C    AE   9E 00156 6$:    MOVAB   NAME_BUFF, NAME_DESC+4                     : 3332
                          4C    AE             20   B0 0015B        MOVW    #32, NAME_DESC                             : 3333
  4C   AE        20             6E             00   2C 0015F        MOVC5   #0, (SP), #32, NAME_DESC, @NAME_DESC+4     : 3334
                                         50    BE      00165                                                       :
                                               7E   7C 00167        CLRQ    -(SP)                                      : 3337
                                               7E   D4 00169        CLRL    -(SP)                                      :
                                         58    AE   9F 0016B        PUSHAB  NAME_DESC                                  :
                                         5C    AE   9F 0016E        PUSHAB  NAME_DESC                                  :
                                               67   DD 00171        PUSHL   IDENT                                      :
                  00000000G     00             06   FB 00173        CALLS   #6, SYS$IDTOASC                            :
                                6A             50   D0 0017A        MOVL    R0, STATUS                                 :
       0F        20             6E             00   2C 0017D        MOVC5   #0, (SP), #32, #15, ATTRIB_BUFF            : 3342
                                         54    AE      00182                                                       :
                                         64    AE   B4 00184        CLRW    ATTRIB_DESC                                : 3343
                          68    AE       54    AE   9E 00187        MOVAB   ATTRIB_BUFF, ATTRIB_DESC+4                 : 3344
                                50       64    AE   3C 0018C        MOVZWL  ATTRIB_DESC, R0                            : 3349
                                51       54    AE   9E 00190        MOVAB   ATTRIB_BUFF, R1                            :
                                11       F4    A7   E9 00194        BLBC    ATTRIBUTES, 7$                             : 3345
                 6140     18    BE       14    AE   28 00198        MOVC3   SD_RESOURCE, @SD_RESOURCE+4, (R1)[R0]      : 3349
                                50       64    AE   3C 0019F        MOVZWL  ATTRIB_DESC, R0                            : 3351
                                51       14    AE   3C 001A3        MOVZWL  SD_RESOURCE, R1                            :
                                               0F   11 001A7        BRB     8$                                         :
                 6140     10    BE       0C    AE   28 001A9 7$:    MOVC3   SD_NORESOURCE, @SD_NORESOURCE+4, (R1)[R0]  : 3356
                                50       64    AE   3C 001B0        MOVZWL  ATTRIB_DESC, R0                            : 3358
                                51       0C    AE   3C 001B4        MOVZWL  SD_NORESOURCE, R1                          :
                                50             51   C0 001B8 8$:    ADDL2   R1, R0                                     :
            64   AE             50             01   A1 001BB        ADDW3   #1, R0, ATTRIB_DESC                        :
                                51       FF    A8   9A 001C0        MOVZBL  RIGHTSLINE, R1                             : 3371
                                69             51   B0 001C4        MOVW    R1, FAODSC                                 :
                 50             6B             22   C1 001C7        ADDL3   #34, RABPTR, R0                            :
                          C0    8F       03    A7   93 001CB        BITB    IDENT+3, #192                              : 3364
                                2A             12      001D0        BNEQ    9$                                         :
                          04    A9             68   9E 001D2        MOVAB   RIGHTSLINE+1, FAODSC+4                     : 3371
                                         64    AE   9F 001D6        PUSHAB  ATTRIB_DESC                                :
                                7E             67   3C 001D9        MOVZWL  IDENT, -(SP)                               :
       7E        02    A7       0E             00   EF 001DC        EXTZV   #0, #14, IDENT+2, -(SP)                    :
                                         5C    AE   DD 001E2        PUSHL   NAME_DESC+4                                :
                                7E       5C    AE   3C 001E5        MOVZWL  NAME_DESC, -(SP)                           :
                                   00000000G   00   9F 001E9        PUSHAB  DISDSC                                     :
                                               50   DD 001EF        PUSHL   R0                                         :
                                               59   DD 001F1        PUSHL   R9                                         :
                  00000000G     00             08   FB 001F3        CALLS   #8, SYS$FAO                                :
                                               21   11 001FA        BRB     10$                                        :
                          04    A9             68   9E 001FC 9$:    MOVAB   RIGHTSLINE+1, FAODSC+4                     : 3377
                                         64    AE   9F 00200        PUSHAB  ATTRIB_DESC                                :
                                               67   DD 00203        PUSHL   IDENT                                      :
                                         58    AE   DD 00205        PUSHL   NAME_DESC+4                                :
                                7E       58    AE   3C 00208        MOVZWL  NAME_DESC, -(SP)                           :
                                   00000000G   00   9F 0020C        PUSHAB  DISDSC                                     :
                                               50   DD 00212        PUSHL   R0                                         :
                                               59   DD 00214        PUSHL   R9                                         :
                  00000000G     00             07   FB 00216        CALLS   #7, SYS$FAO                                :
                                               6B   DD 0021D 10$:   PUSHL   RABPTR                                     :
                  00000000G     00             01   FB 0021F        CALLS   #1, SYS$PUT                                :
                                               5E   DD 00226        PUSHL   SP                                         : 3382
                                         F4    A7   9F 00228        PUSHAB  ATTRIBUTES                                 :
                                               57   DD 0022B        PUSHL   R7                                         :
```

```
                                    04   AC  DD 0022D          PUSHL   HOLDER_ADDR                                  ;
                00000000G  00            04  FB 00230          CALLS   #4, SYS$FIND_HELD                           ;
                           6A            50  D0 00237          MOVL    R0, STATUS                                  ;
                           0E            50  E8 0023A          BLBS    R0, 11$                                     ; 3383
                                         56  94 0023D          CLRB    CONTINUE                                    ; 3386
                000021EC   8F            50  D1 0023F          CMPL    R0, #8684                                   ; 3387
                                          03 12 00246          BNEQ    11$                                         ;
                           6A             01 D0 00248          MOVL    #1, STATUS                                  ; 3388
                           03             56 E9 0024B 11$:     BLBC    CONTINUE, 12$                               ; 3391
                                       FF05  31 0024E          BRW     6$                                          ;
                                          5E DD 00251 12$:     PUSHL   SP                                          ; 3393
                00000000G  00             01 FB 00253          CALLS   #1, SYS$FINISH_RDB                          ;
                           50             6A D0 0025A 13$:     MOVL    STATUS, R0                                  ; 3395
                                             04 0025D          RET                                                 ; 3397

; Routine Size:  606 bytes,    Routine Base:  $CODE$ + 196A

; 3424          3398  1
; 3425          3399  1 END                                   ! End of module
; 3426          3400  0 ELUDOM
```

```
;                                  PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  $GLOBAL$                        20  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $OWN$                          436  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $PLIT$                         420  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                        7112  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
```

```
;                           Library Statistics
;
;                                    -------- Symbols --------      Pages       Processing
;       File                         Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1      18619        93         0        1000        00:02.0
;  _$255$DUA28:[SYSLIB]TPAMAC.L32;1      42         0         0          14        00:00.2
```

```
;                                COMMAND QUALIFIERS
;
;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:RIGHTSMAN/OBJ=OBJ$:RIGHTSMAN MSRC$:RIGHTSMAN/UPDATE=(ENH$:RIGHTSMAN)
```

```
; Size:          7112 code + 876 data bytes
; Run Time:        01:50.9
; Elapsed Time:    01:54.6
; Lines/CPU Min:    1839
; Lexemes/CPU-Min: 21878
; Memory Used:  324 pages
; Compilation Complete
```

UAFMAIN
LIS

RIGHTSMAN
LIS